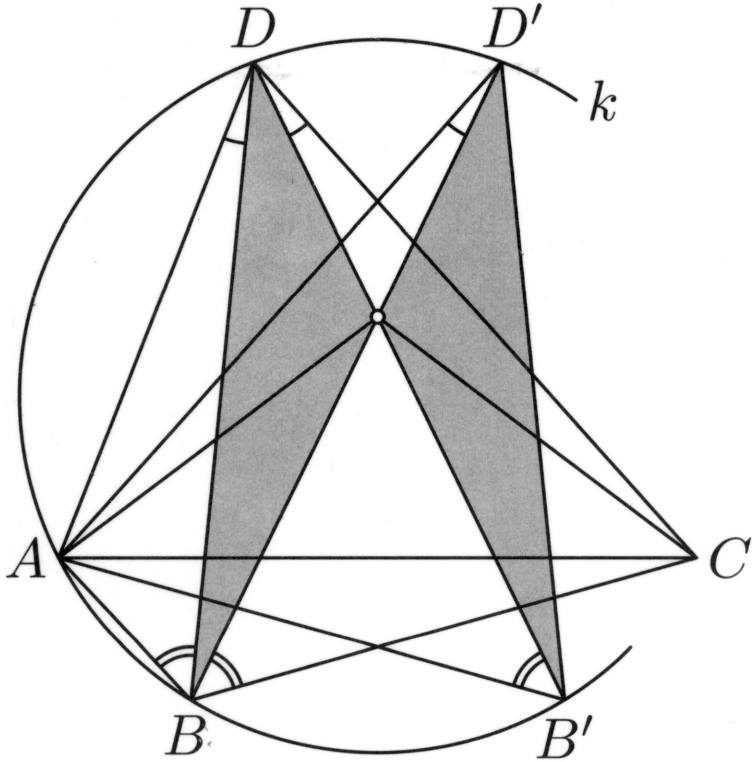
53. ročník matematické olympiády na středních školách

MO

53. ročník matematické olympiády na středních školách



Fč
m+f

PADESÁTÝ TŘETÍ ROČNÍK MATEMATICKÉ OLYMPIÁDY NA STŘEDNÍCH ŠKOLÁCH

Zpráva o řešení úloh ze soutěže konané ve školním roce 2003/2004

45. MEZINÁRODNÍ MATEMATICKÁ OLYMPIÁDA

ŠESTNÁCTÁ MEZINÁRODNÍ OLYMPIÁDA V INFORMATICE

JEDNOTA ČESKÝCH MATEMATIKŮ A FYZIKŮ PRAHA

Za přispění spolupracovníků zpracovali
RNDr. Karel Horák, CSc., Mgr. Jan Kára,
doc. RNDr. Jaromír Šimša, CSc., RNDr. Jaroslav Švrček, CSc.,
doc. RNDr. Pavel Töpfer, CSc., RNDr. Jaroslav Zhouf, PhD.

ISBN 80-7015-025-4

Obsah

O průběhu 53. ročníku matematické olympiády

Hlavními pořadateli 53. ročníku matematické olympiády, který se uskutečnil ve školním roce 2003/04, byly podobně jako v letech předešlých Ministerstvo školství, mládeže a tělovýchovy ČR, Jednota českých matematiků a fyziků a Matematický ústav akademie věd ČR. Organizací soutěže byl pověřen Ústřední výbor MO, jehož předsedou je doc. RNDr. *Jaromír Šimša*, CSc. Garanci za chod soutěže v kategoriích A, B, C, Z a P pak přebírají příslušní místopředsedové ÚV MO, a to RNDr. *Jaroslav Švrček*, CSc., doc. RNDr. *Pavel Tlustý*, CSc., a doc. RNDr. *Pavel Töpfer*, CSc. Funkci tajemníka ÚV MO vykonával i v tomto ročníku soutěže RNDr. *Karel Horák*, CSc.

Stejně jako v předešlých letech jsou výběrem úloh pro jednotlivá kola matematické olympiády v jednotlivých kategoriích pověřeny dvě úlohové komise — jedna pro kategorie A, B a C, druhá pro kategorie Z. Obě komise připravují úlohy s ročním předstihem a členové pověření garancí nad jednotlivými kategoriemi v jednotlivých ročnících soutěže připravují t0ž komentáře k úlohám I. kola. V průběhu 53. ročníku MO byli garanty jednotlivých kategorií RNDr. *Jaroslav Švrček*, CSc. (kategorie A), doc. RNDr. *Pavel Novotný*, CSc. (kategorie B) a RNDr. *Jaroslav Zhouf*, Ph.D. (kategorie C).

Obě komise se scházejí pravidelně dvakrát během školního roku, a to vždy v listopadu a v květnu. Úlohy všech kol kategorie P připravili pro tento ročník soutěže připravil tým pracovníků z MFF UK pod vedením doc. Töpfera.

Na rozdíl od předešlého, 52. ročníku MO bylo zajištění všech kol soutěže provedeno s velkým časovým předstihem, takže v průběhu 53. ročníku se nevyskytly téměř žádné komplikace. Oba letáky i komentáře k úlohám I. kola se dostaly do škol včas. Učitelé matematiky měli navíc možnost seznámit se se zadáním úloh I. kola 53. ročníku MO ještě před zahájením nového školního roku prostřednictvím časopisu *Matematika–fyzika–informatika*, na jehož stránkách vždy v posledních dvou číslech jednotlivých ročníků vycházejí texty úloh I. kola pro nadcházející ročník MO.

Ústřední výbor matematické olympiády se sešel během tohoto ročníku na dvou jednodenních zasedáních, a to 12. prosince 2003 v Praze a podruhé tradičně u příležitosti celostátního kola kategorií A a P — 29. března 2004 v Přerově. Na obou zasedáních byly kromě pravidelných bodů programu řešeny i některé otázky týkající se především průběhu příštích ročníků matematické olympiády.

Celostátní III. kolo 53. ročníku matematické olympiády v kategoriích A a P se uskutečnilo 28. března – 3. dubna 2004 v Přerově. Ústřední výbor MO pověřil organizací III. kola Gymnázium Jakuba Škody v Přerově. Vedení školy ve spolupráci s Přírodovědeckou fakultou Univerzity Palackého v Olomouci, olomouckou pobočkou JČMF a představiteli města Přerov vytvořilo pro soutěž mimořádně kvalitní podmínky. Všichni soutěžící (43 v kategorii A, 30 v kategorii P) a pozvaní hosté byli ubytováni v luxusním hotelu Jana, v jehož konferenčních sálech se uskutečnilo III. kolo kategorie A, teoretická část III. kola kategorie P a také zasedání ÚV MO. Praktická část III. kola kategorie P pak proběhla ve specializovaných učebnách nedalekého Gymnázia J. Škody. Zejména soutěžícím tak byly připraveny pro vlastní soutěž takřka ideální podmínky. O mimořádně zdařilý průběh celého III. kola soutěže se postaral také organizační tým učitelů matematiky Gymnázia Jakuba Škody v Přerově pod vedením ředitele školy Mgr. *Jana Rašky.*

Oficiální zahájení soutěže proběhlo v neděli 28. března 2004 v prostorách starobylé auly gymnázia Jakuba Škody za přítomnosti prof. RNDr. *Lubomíra Dvořáka*, CSc., děkana PřF UP v Olomouci, prof. RNDr. *Karla Segetha*, CSc., ředitele MÚ AV ČR, doc. RNDr. *Jiřího Vanžury*, CSc., *Jindřicha Valoucha*, starosty města Přerova, a dalších významných představitelů společenského života Olomouckého kraje a města Přerova.

Pořadatelé zajistili pro účastníky obou III. kol v kategorii A i P hodnotný doprovodný program. Soutěžící obou kategorií navštívili přerovské Muzeum J. A. Komenského, které se pyšní především ojedinělou sbírkou učebnic a učebních předmětů již z dob Rakouska-Uherska, a to až po nepříliš vzdálenou minulost. Vždy druhý soutěžní den odpoledne absolvovali soutěžící autobusový zájezd do blízkého okolí. Účastnící kategorie A měli možnost navštívit zámek a zámecký park v nedaleké Kroměříži, který sloužil jako letní sídlo olomouckých arcibiskupů, a rozsáhlou zříceninu hradu Helfštýn. Soutěžící kategorie P pak absolvovali polodenní zájezd za památkami města Olomouce spojený s návštěvou blízkého Svatého Kopečku. V rámci III. kola proběhla pro účastníky soutěže na hostitelském gymnáziu zajímavá prezentace matematického softwaru *MATHE-*

MATICA americké firmy Wolfram Research, kterou u nás zastupuje pražská firma Elkan.

Cenu profesora Zelinky za originální řešení ve výši 5 000 Kč, kterou již podruhé věnoval *Igor Puzanov*, absolvent bývalého matematického gymnázia W. Piecka v Praze, získal *Martin Selecký* z Gymnázia B. Němcové v Hradci Králové za hezké a originální řešení páté úlohy celostátního kola. Zvláštní cenu děkana PřF UP v Olomouci za jediné bezchybné řešení nejobtížnější praktické úlohy III. kola v kategorii P získal *Jiří Štěpánek* z Gymnázia v Brně na tř. Kpt. Jaroše.

Hned v následujícím týdnu 5.–9. dubna proběhlo v Kostelci nad Černými lesy výběrové soustředění členů českého reprezentačního družstva před 45. MMO. Na toto soustředění bylo pozváno všech devět vítězů III. kola kategorie A. Na základě jejich výsledků na tomto soustředění a ve II. a III. kole bylo šest nejlepších vybráno do českého reprezentačního družstva pro 45. MMO. Podrobnou informaci o 45. mezinárodní matematické olympiádě a o 16. mezinárodní olympiádě v informatice najdete v příslušných kapitolách této ročenky.

Pro nejlepší řešitele krajských kol v kategoriích B a C uspořádal ÚV MO od 1. do 8. června tradiční soustředění v Jevíčku. Zúčastnilo se ho 40 žáků z celé republiky. Podobně pro nejlepší řešitele kategorie A uspořádal ÚV MO v Janských Lázních 12.–18. září soustředění širšího výběru žáků před 46. MMO.

Ústřední výbor matematické olympiády v České republice si dovoluje na tomto místě poděkovat všem sponzorům za poskytnutí hodnotných cen pro nejlepší řešitele III. kola v kategoriích A a P, především pak přerovské firmě Emos a pražské firmě Elkan. ÚV MO dále děkuje přerovským akciovým společnostem Emos a Hanácká kyselka a pražské akciové společnosti Tezas za jejich podporu v souvislosti se zajištěním jednotného oblečení členů českého reprezentačního družstva na 45. MMO. Poděkování patří též pražské firmě Elkan, která se výraznou finanční podporou zasloužila o vytištění této ročenky.

Významným oceněním našich úspěšných olympioniků se už tradičně stává Præmium Bohemiæ, které vždy 4. prosince uděluje Nadace B. Jana Horáčka Českému ráji. V tomto roce byli mezi oceněnými ve studentské kategorii *František Konopecký* a *Jan Moláček* (25 000 Kč za stříbrnou medaili na 45. MMO), *Daniel Marek* a *Petr Škoda* (25 000 Kč za stříbrnou medaili na 16. IOI), *Vítězslav Kala* a *Jaromír Kuben* (15 000 Kč za bronzovou medaili na 45. MMO)

Projev předsedy Ústředního výboru MO při slavnostním zahájení ústředního kola 53. ročníku MO v Přerově

Dámy a pánové, vážení hosté, milí soutěžící,

úkolem organizátorů každého soutěžního zápolení je připravit pro jeho účastníky vhodné a důstojné podmínky, které v případě sportovních soutěží zahrnují regulérní hřiště s vhodným vybavením a tým spravedlivých rozhodčích. Pro naši matematickou olympiádu potřebujeme místo sportovního náčiní pěkné, dostatečně původní a neotřelé úlohy, na kterých si vy, mladí soutěžící, vyzkoušíte svoje znalosti, důvtip a tvůrčí invenci. Věřím, že vám řešení takových úloh přináší mnoho radosti a vzrušení, byť z malých, ale překvapivých a krásných objevů. Tak se už na střední škole přesvědčujete, že matematika je nevyčerpatelně rozmanitá a živoucí vědecká disciplína, ve které je stále mnoho prostoru pro další bádání.

Jsem velmi rád, že poslední slova mohu potvrdit oznámením nového významného výsledku, který byl při loňském ústředním kole pouhou hypotézou, třebaže starou více než 150 let. *Čísla 8 a 9 jsou jedinou dvojicí po sobě jdoucích přirozených čísel, která jsou obě mocninami přirozených čísel.* Číslo 8 je třetí mocninou čísla 2, číslo 9 je druhou mocninou čísla 3. V roce 2003 švýcarský matematik rumunského původu *Preda Mihăilescu* dokázal, že neexistuje žádná jiná dvojice po sobě jdoucích čísel, z nichž každé by bylo druhou, třetí, pátou nebo vyšší (asi víte, proč jsem vynechal *čtvrtou*) mocninou přirozeného čísla. Znamená to tedy, že rovnice $a^b - c^d = 1$ má jediné řešení v oboru přirozených čísel a, b, c, d, předpokládáme-li, že $b > 1$ a $d > 1$.

Milí soutěžící, zítra a pozítří vám předložíme úlohy, které nebudete řešit desítky let, měsíců či dnů, ale pouze několik hodin. Budete tedy bojovat nejen s obtížnými zadáními, ale i s časem. Přeji vám všem, abyste obstáli co nejlépe. Ústřední kolo 53. ročníku MO prohlašuji za zahájené.

Tabulka 1

Počty žáků středních škol soutěžících v I. kole 53. ročníku MO

Kraj	Kategorie								Celkem	
	A		B		C		P			
	S	U	S	U	S	U	S	U	S	U
Praha	151	54	128	43	177	115	19	16	475	228
Středočeský	105	36	99	41	118	85	18	17	340	179
Jihočeský	75	35	55	27	95	60	5	1	230	123
Plzeňský	47	12	43	14	193	61	3	3	286	90
Karlovarský	30	9	15	7	15	14	1	1	61	31
Ústecký	45	30	39	19	68	47	3	3	155	99
Liberecký	64	27	35	8	85	68	8	8	192	111
Královéhradecký	52	31	34	15	78	60	8	6	172	112
Pardubický	26	18	25	19	49	34	8	5	108	76
Vysočina	55	41	50	37	100	80	15	10	220	168
Jihomoravský	188	103	81	45	139	121	16	16	424	285
Zlínský	80	38	75	21	93	46	6	6	254	111
Olomoucký	46	24	43	13	54	46	2	2	145	85
Moravskoslezský	32	12	60	32	108	93	20	20	220	157
ČR	996	470	782	341	1 372	930	132	114	3 282	1 855

Tabulka 2

Počty žáků středních škol soutěžících v II. kole 53. ročníku MO

Kraj	Kategorie								Celkem	
	A		B		C		P			
	S	U	S	U	S	U	S	U	S	U
Praha	54	25	43	9	101	44	16	11	214	89
Středočeský	35	5	41	5	77	27	17	6	170	43
Jihočeský	35	3	27	5	60	25	1	1	123	34
Plzeňský	11	4	14	4	58	24	3	1	86	33
Karlovarský	9	2	7	2	14	5	1	0	31	9
Ústecký	30	2	16	0	39	8	3	0	88	10
Liberecký	27	0	8	2	29	13	8	4	72	19
Královéhradecký	31	10	14	2	59	21	6	2	110	35
Pardubický	16	4	18	3	33	12	5	3	72	22
Vysočina	35	7	22	2	70	17	7	3	134	29
Jihomoravský	97	31	39	11	107	36	16	9	259	87
Zlínský	37	9	21	3	46	27	6	3	110	42
Olomoucký	24	2	13	3	44	12	2	1	83	18
Moravskoslezský	12	7	32	8	93	17	20	5	157	37
ČR	453	111	315	59	830	288	111	49	1 709	507

S ... počet všech soutěžících U ... počet úspěšných řešitelů

Nejúspěšnější řešitelé II. kola MO v kategoriích A, B, C a P

Z každého kraje a z každé kategorie jsou dle dostupných výsledků uvedeni všichni úspěšní řešitelé, kteří skončili do desátého místa. Označení G znamená gymnázium.

Kraj Praha

Kategorie A

1. *Alexandr Kazda*, G Praha 6, Nad Alejí
2. *Pavel Kocourek*, SPŠST, Praha 1, Panská
3. *Jan Martínek*, G Ch. Dopplera, Praha 5

4.–11. *Václav Potoček*, SPŠST, Praha 1, Panská
Martin Doležal, G Praha 10, Omská
Jan Drašnar, G J. Keplera, Praha 6
Miroslav Hlaváč, G Ch. Dopplera, Praha 5
Martin Chudoba, G J. Heyrovského, Praha 5
Ondřej Kvapilík, G Ch. Dopplera, Praha 5
Mikuláš Peksa, G Ch. Dopplera, Praha 5
Jan Škampa, G J. Keplera, Praha 6

Kategorie B

1. *Daniel Marek*, G Ch. Dopplera, Praha 5

2.–3. *Tomáš Hejda*, G Ch. Dopplera, Praha 5
Radek Žlebčík, G Ch. Dopplera, Praha 5

4. *Ondřej Fremund*, G J. Nerudy, Praha 1

5.–6. *Vojtěch Horký*, G Praha 6, Nad Alejí
Adam Přenosil, G Praha 3, Sladkovského nám.

7. *Jan Papež*, G Praha 8, Ústavní

8.–9. *Josef Brechler*, G Ch. Dopplera, Praha 5
Jan Verner, G Praha 9, Špitálská

Kategorie C

1. *Šárka Gregorová*, G Praha 6, Nad Alejí
2. *Lukáš Malina*, G Ch. Dopplera, Praha 5
3.–4. *Michal Jex*, G J. Heyrovského, Praha 5
Tomáš Křen, G Ch. Dopplera, Praha 5
5.–6. *Michal Rolínek*, G J. Keplera, Praha 6
Jiří Vackář, G Praha 8, Ústavní
7. *Tomáš Bělka*, G Praha 10, Voděradská
8. *Tomáš Tintěra*, G Ch. Dopplera, Praha 5
9.–11. *Pavel Brožek*, G Praha 4, Písnická
Alena Kubíková, G Praha 8, Ústavní
Jakub Mrva, G J. Heyrovského, Praha 5

Kategorie P

1. *Daniel Marek*, G Ch. Dopplera, Praha 5
2. *Petr Škoda*, G Praha 8, Ústavní
3. *Daniel Balaš*, G Praha 6, Arabská
4. *Ján Zahornadský*, G Ch. Dopplera, Praha 5
5. *David Matoušek*, G Ch. Dopplera, Praha 5
6. *Zdeněk Sojka*, SPŠST, Praha 1, Panská
7. *Roman Smrž*, G Praha 4, Ohradní
8.–9. *Martin Bosák*, SPŠST, Praha 1, Panská
Tomáš Petříček, G Praha 6, Arabská
10.–11. *Martina Tomisová*, G Ch. Dopplera, Praha 5
Petr Soběslavský, G J. Heyrovského, Praha 5

Středočeský kraj

Kategorie A

1.–2. *Jindřich Soukup*, G Kladno, nám. E. Beneše
Benjamin Vejnar, G Nymburk
3.–5. *Martin Kozák*, G Český Brod
Michal Richter, GaSPgŠ Čáslav
Jan Váňa, GJO Kutná Hora

Kategorie B

1. *Lukáš Herman*, G dr. J. Pekaře, Mladá Boleslav
2. *Karolína Chmielová*, G Benešov

3.–4. *Václav Gergelits*, G Benešov
Marek Scholz, G Neratovice

Kategorie C

1. *Václav Šámal*, G Říčany
2. *Eva Myšáková*, G Kolín
3. *Petr Fojtů*, G Benešov
4. *Rudolf Rosa*, G Kladno, nám. E. Beneše
5. *Jiří Špale*, G Sedlčany

6.–7. *Tomáš Hubálek*, G Kralupy
Jindřich Libovický, G Kladno, nám. E. Beneše

8.–9. *Pavel Beščec*, G Beroun
Michal Holub, G Beroun

10.–11. *Miloš Broulík*, G Mladá Boleslav
Tomáš Tesař, G Příbram

Kategorie P

1. *Petr Švec*, G Beroun
2. *Jaroslav Havlín*, GOA Sedlčany
3. *Miroslav Frantes*, G Benešov
4. *Miroslav Kratochvíl*, G a SPedŠ Čáslav
5. *Petr Balek*, G Beroun
6. *Vojtěch Zehnálek*, G Benešov

Jihočeský kraj

Kategorie A

1.–2. *Lubomír Krčmář*, G České Budějovice, Jírovcova
Martin Pilát, G České Budějovice, Česká

3. *Jiří Václavík*, G Strakonice

Kategorie B

1. *Eliška Lehečková*, G České Budějovice, Česká
2. *Jiří Kníže*, G Strakonice
3.–4. *Zuzana Bainová*, G České Budějovice, Jírovcova
Tomáš Trnka, G České Budějovice, Jírovcova
5. *Vojtěch Dušek*, G České Budějovice, Jírovcova

Kategorie C

1. *Adam Kabela*, G České Budějovice, Jírovcova
2. *Radim Hošek*, G České Budějovice, Jírovcova
3. *Jiří Blažek*, ZŠ Nerudova, České Budějovice
4.–5. *Martin Houštecký*, G České Budějovice, Česká
Josef Špak, G České Budějovice, Jírovcova
6.–8. *Karel Chuchel*, G České Budějovice, Jírovcova
Jaroslav Icha, G České Budějovice, Jírovcova
Jakub Loucký, G Písek
9.–11. *Tomáš Kouba*, G Český Krumlov
Josef Pihera, G Strakonice
Martina Urbanová, G České Budějovice, Jírovcova

Kategorie P

1. *Vladimír Čunát*, G České Budějovice, Jírovcova

Plzeňský kraj

Kategorie A

1. *Pavel Paták*, G Sušice
2. *Martina Ullrichová*, G Plzeň, Mikulášské nám.
3. *Petr Cvachovec*, G Plzeň, Mikulášské nám.
4. *Jiří Kohout*, G Plzeň, Mikulášské nám.

Kategorie B

1. *Marie Dostálová*, G Stříbro

2.–3. *Vojtěch Kolomičenko*, G Plzeň, Mikulášské nám.
Vladimír Sirotek, G L. Pika, Plzeň
4. *Jan Šípek*, G L. Pika, Plzeň

Kategorie C

1.–2. *Tereza Hlaváčová*, G Plzeň, Mikulášské nám.
Dora Horáková, G Plzeň, Mikulášské nám.
3.–4. *Štěpán Kropík*, SPŠ elektro, Plzeň
Stanislav Šimek, SPŠ elektro, Plzeň
5. *Jana Horová*, G Plzeň, Mikulášské nám.
6.–9. *Marek Engelthaler*, SPŠ elektro, Plzeň
Lubomír Jurečka, SPŠ elektro, Plzeň
Jan Sedlák, SPŠ elektro, Plzeň
Radek Vozák, G Plzeň, Mikulášské nám.
10.–15. *Veronika Binderová*, G Plzeň, Mikulášské nám.
Daniela Krupičková, G Plzeň, Mikulášské nám.
Petra Nožičková, G Plzeň, Mikulášské nám.
Kryštof Touška, G Klatovy
Jan Václavík, G Plzeň, Mikulášské nám.
Michal Vahala, G Plasy

Kategorie P

1. *Jan Bulánek*, G J. Vrchlického, Klatovy

Karlovarský kraj

Kategorie A

1. *Eva Černohorská*, První české G Karlovy Vary
2. *Petr Dohnal*, G Sokolov

Kategorie B

1. *Petr Žáček*, SCHŠ Cheb
2. *Eva Černohorská*, První české G Karlovy Vary
3. *Vladimír Hanzlík*, První české G Karlovy Vary

Kategorie C

1. *Štěpán Masák*, První české G Karlovy Vary
2. *Radek Hájek*, G Cheb
3. *Kateřina Humplíková*, První české G Karlovy Vary
4. *Tomáš Beňák*, První české G Karlovy Vary
5. *Jakub Korselt*, G Aš

Ústecký kraj

Kategorie A

1. *Tomáš Franc*, G Teplice, Čs. Dobrovolců
2. *Vojtěch Kaiser*, G Ústí nad Labem, Stavbařů

Kategorie C

1.–2. *Petr Sýkora*, G Děčín, Komenského nám.
Jaroslav Vosáhlo, G Ústí nad Labem, Stavbařů
3. *Pavel Eger*, G TGM Litvínov
4. *Martin Obr*, G Chomutov
5.–6. *Jan Řezníček*, G Lovosice
Petr Tříletý, G Děčín, Komenského nám.
7. *Michal Merganič*, G Duchcov
8. *Daniel Šimsa*, G J. Jungmanna, Litoměřice

Liberecký kraj

Kategorie A

1. *Petr Píša*, G a SPeGŠ Liberec, Jeronýmova
2. *Michal Kollert*, GFXŠ, Liberec
3.–4. *Lukáš Ježek*, GFXŠ, Liberec
Miroslav Kloz, GFXŠ, Liberec

Kategorie B

1. *Jan Hrnčíř*, GFXŠ, Liberec
2. *Michal Vaner*, G Turnov

Kategorie C

1. *Václav Koucký*, G Tanvald
2. *Petr Pavlů*, SPŠSE, Liberec
3. *Veronika Počerová*, GFXŠ, Liberec

4.–5. *Tomáš Kobrle*, G Jilemnice
Lukáš Nykrýn, G Jablonec, U Balvanu

6. *Hana Schaabová*, G Dr. Randy, Jablonec n. N.

7.–10. *Michal Haniš*, G Frýdlant
Antonín Hoskovec, G Jablonec, U Balvanu
Jakub Chmelař, GFXŠ, Liberec
Veronika Mužíčková, G Dr. Randy, Jablonec n. N.

Kategorie P

1. *Michal Vaner*, G Turnov
2. *Jiří Schejbal*, G Turnov

3.–4. *Michal Tuláček*, G Liberec, Jeronýmova
Jan Kolomazník, G Turnov

Královéhradecký kraj

Kategorie A

1. *Jan Moláček*, GJKT, Hradec Králové
2. *Martin Selecký*, G B. Němcové, Hradec Králové
3. *Radek Moravec*, G B. Němcové, Hradec Králové
4. *Ivo Machek*, GJKT, Hradec Králové
Jan Marek, G B. Němcové, Hradec Králové
6. *Michal Černohorský*, G B. Němcové, Hradec Králové
Jan Ondruš, GFMP, Rychnov nad Kněžnou
Filip Studnička, G B. Němcové, Hradec Králové
Libor Šimůnek, GJKT, Hradec Králové

Pavel Špryňar, G B. Němcové, Hradec Králové

Kategorie B

1. *Jan Voltr*, GJKT, Hradec Králové
2. *Vendula Uchytilová*, GJKT, Hradec Králové

Kategorie C

1. *Pavel Kuchyňa*, G B. Němcové, Hradec Králové
   Martin Petr, GJKT, Hradec Králové
3. *Jakub Dundálek*, JG Náchod
4. *Miloslav Jára*, JG Náchod
   Pavlína Krátošková, G B. Němcové, Hradec Králové
6. *Marek Buday*, G B. Němcové, Hradec Králové
   Matěj Kouba, G Hostinné
8. *Kateřina Mudroňová*, GJKT, Hradec Králové
   Jaroslav Štěrba, GJKT, Hradec Králové
10. *Adam Martinek*, GJKT, Hradec Králové

Kategorie P

1. *Oto Petřík*, G Vrchlabí
2. *Petr Šťovík*, GJKT, Hradec Králové

Pardubický kraj

Kategorie A

1. *Tereza Klimošová*, G Lanškroun
2. *Bára Scholleová*, G Pardubice, Dašická
3.–4. *Vojtěch Novotný*, G Chrudim
   Marek Scholle, G Pardubice, Dašická

Kategorie B

1. *Tereza Klimošová*, G Lanškroun

2. *Helena Šedováková*, G Pardubice, Dašická
3. *Kristýna Stodolová*, G Polička

Kategorie C

1. *Marek Scholle*, G Pardubice, Dašická
2. *Jiří Hajný*, G Chrudim
3. *Jiří Havran*, G Polička
4. *Kristýna Hrdinová*, G Žamberk
5. *Eva Dobešová*, G Jevíčko

6.–7. *Lenka Balošková*, G Pardubice, Dašická
Michal Leňo, G Vysoké Mýto

8.–10. *Miroslav Klimoš*, G Lanškroun
Z. Mudruňková, G Pardubice, Dašická
Lubomír Štěpánek, G Pardubice, Dašická

Kategorie P

1. *Martin Dobroucký*, G Moravská Třebová
2. *Jindřich Flídr*, G Lanškroun
3. *Petr Paščenko*, G Pardubice, Dašická

Kraj Vysočina

Kategorie A

1. *Ondřej Křivánek*, G Třebíč
2. *Zdeněk Tichý*, G Pelhřimov

3.–4. *Luděk Gregor*, GVM Nové Město na Moravě
Dominik Macáš, G Bystřice nad Pern.

5.–7. *Josef Janák*, G Velké Meziříčí
Petr Pravlovský, G Jihlava
Klára Ševčíková, G Jihlava

Kategorie B

1. *Ondřej Hoferek*, G Žďár nad Sázavou
2. *Karel Lavička*, G Jihlava

Kategorie C

1. *Mirek Dočekal*, G Jihlava
2.–3. *Helena Dvořáková*, G Třebíč
Tomáš Herceg, G Třebíč
4. *Barbora Štohanzlová*, HG Havlíčkův Brod
5.–8. *Dana Dohnalová*, HG Havlíčkův Brod
Milan Dvořák, GVM Nové Město na Moravě
Jana Gajdošíková, G Třebíč
Jan Korbel, G Jihlava
9. *Miloslav Sobotka*, G Žďár nad Sázavou
10.–12. *František Horák*, G Velké Meziříčí
10.–12. *Rostislav Stríž*, GVM Nové Město na Moravě
10.–12. *Karel Trojan*, G Jihlava

Kategorie P

1.–2. *Petr Baudiš*, G Ad Fontes, Jihlava
Martin Tomec, G Třebíč
3. *Martin Jonáš*, SPŠ Jihlava

Jihomoravský kraj

Kategorie A

1. *Vítězslav Kala*, G Brno, tř. Kpt. Jaroše
2.–3. *Jaromír Kuben*, G Brno, tř. Kpt. Jaroše
Alexandr Pícha, G Brno, tř. Kpt. Jaroše
4. *Tomáš Hebelka*, G Brno, Vídeňská
5.–6. *Jan Křetínský*, GML Brno, Žižkova
Jan Novotný, G Brno, tř. Kpt. Jaroše
7.–8. *Jana Fabriková*, G Brno, tř. Kpt. Jaroše
Pavel Troubil, G Brno, tř. Kpt. Jaroše
9.–10. *Sven Dražan*, G Brno, tř. Kpt. Jaroše
Jiří Krčál, G Brno, tř. Kpt. Jaroše

Kategorie B

1.–2. *Jaromír Kuben*, G Brno, tř. Kpt. Jaroše
Vojtěch Říha, G Brno, tř. Kpt. Jaroše
3. *Alexandr Pícha*, G Brno, tř. Kpt. Jaroše
4. *Jakub Opršal*, G Brno, tř. Kpt. Jaroše
5. *Jiří Zelinka*, G Brno, tř. Kpt. Jaroše
6. *Jan Rygl*, G Brno, tř. Kpt. Jaroše
7. *Martin Křivánek*, G Brno, tř. Kpt. Jaroše
8.–9. *Martin Kočí*, G Brno, tř. Kpt. Jaroše
Jakub Pracný, G Brno, tř. Kpt. Jaroše
10. *Pavlína Novotná*, G Kyjov

Kategorie C

1.–2. *Petr Buršík*, G Brno, tř. Kpt. Jaroše
Zbyněk Konečný, G Brno, tř. Kpt. Jaroše
3. *Lucie Fabriková*, G Brno, tř. Kpt. Jaroše
4. *Jiří Řihák*, G Brno, tř. Kpt. Jaroše
5. *Petr Velan*, G Brno, tř. Kpt. Jaroše
6. *Petr Fiala*, G Brno, tř. Kpt. Jaroše
7.–9. *Petra Burešová*, G Brno, tř. Kpt. Jaroše
Tomáš Jelínek, G Brno, tř. Kpt. Jaroše
Lenka Košařová, G Brno, tř. Kpt. Jaroše
10.–11. *Jan Komínek*, G Brno, tř. Kpt. Jaroše
Jan Kovář, G Brno, tř. Kpt. Jaroše

Kategorie P

1. *Jiří Štěpánek*, G Brno, tř. Kpt. Jaroše
2. *Martin Vejnár*, G Brno, tř. Kpt. Jaroše
3.–4. *Martin Křivánek*, G Brno, tř. Kpt. Jaroše
Pavel Troubil, G Brno, tř. Kpt. Jaroše
5. *Kryštof Hoder*, G Brno, tř. Kpt. Jaroše
6. *Jana Fabriková*, G Brno, tř. Kpt. Jaroše
7. *Jiří Zelinka*, G Brno, tř. Kpt. Jaroše
8. *Martin Lopatář*, G Brno, tř. Kpt. Jaroše
9. *Jiří Appl*, G Brno, Vejrostova

Kategorie A

1. *Marek Pechal*, G Zlín, Lesní čtvrť
2. *Stanislav Basovník*, G Kroměříž
3. *František Konopecký*, G L. Jaroše, Holešov

4.–5. *Martin Dungl*, G Kroměříž
Milan Procházka, SPŠE, Rožnov pod Radhoštěm

6.–8. *Aleš Holub*, G Uherské Hradiště
Jan Olšina, G Kroměříž
Milan Pečeňa, G Zlín, Lesní čtvrť

9. *Pavel Šalom*, G Rožnov pod Radhoštem

Kategorie B

1. *Pavel Šalom*, G Rožnov pod Radhoštem
2. *Jan Váňa*, G Zlín, Lesní čtvrť
3. *Lukáš Strnad*, G Kroměříž

Kategorie C

1. *Petr Dlabaja*, G L. Jaroše, Holešov

2.–4. *Jan Blizňák*, GFP, Valašské Meziříčí
Martin Tesař, G Zlín, Lesní čtvrť
Jan Urban, G Zlín, Lesní čtvrť

5. *Iva Pobořilová*, GFP, Valašské Meziříčí
6. *Martina Stehnová*, G L. Jaroše, Holešov

7.–11. *Hana Bosáková*, GFP, Valašské Meziříčí
Tomáš Ehrlich, G L. Jaroše, Holešov
Tereza Pechová, GaOA Valašské Klobouky
Roman Stachoň, G Kroměříž
Daniela Veselá, G Zlín, Lesní čtvrť

Kategorie P

1. *Ondřej Bílka*, G Zlín, Lesní čtvrť
2. *Stanislav Basovník*, G Kroměříž
3. *Jan Zimmermann*, G Zlín, Lesní čtvrť

Olomoucký kraj

Kategorie A

1. *Jana Hrudíková*, GJŠ, Přerov
2. *Jana Sedláčková*, GJŠ, Přerov

Kategorie B

1. *Jan Šrámek*, GJŠ, Přerov
2.–3. *Marek Šnyrch*, SGO, Olomouc
Zdeněk Černohouz, SGO, Olomouc

Kategorie C

1. *Anežka Faltýnková*, GJŠ, Přerov
2. *Veronika Kuncová*, G Uničov
3.–4. *Jakub Zouhar*, SGO, Olomouc
Jakub Križan, GJW, Prostějov
5. *Simona Macháčová*, GJŠ, Přerov
6.–12. *Martin Faltičko*, SGO, Olomouc
Tomáš Javůrek, G Jeseník
Ondřej Klabal, GJW, Prostějov
Markéta Paloncýová, G Šumperk
Martin Poklop, SGO, Olomouc
Jan Sluše, SPŠS, Olomouc
Jakub Suchý, G Šumperk

Kategorie P

1. *Jan Matoušek*, GJW, Prostějov

Moravskoslezský kraj

Kategorie A

1. *Tomáš Gavenčiak*, GMK, Bílovec
2. *Jaroslav Hančl*, GMK, Bílovec

3. *Jan Uhlík*, GMK, Bílovec
4.–5. *Pavel Motloch*, G P. Bezruče, Frýdek-Místek
Vojtěch Skubanič, GMK, Bílovec
6.–7. *Zuzana Safernová*, GMK, Bílovec
Josef Toman, GMK, Bílovec

Kategorie B

1. *Hančl Jaroslav*, GMK, Bílovec
2. *Kubetta Adam*, GMK, Bílovec
3. *Uhlík Jan*, GMK, Bílovec
4. *Motloch Pavel*, G P. Bezruče, Frýdek-Místek
5.–6. *Kunčar Jiří*, GMK, Bílovec
Zajac Vít, G Bruntál
7. *Žabenský Josef*, GMK, Bílovec
8. *Štambachr Jakub*, G Ostrava-Poruba, Čs. exilu

Kategorie C

1.–2. *Šebetovský Jan*, SOŠ, Kopřivnice
Štufka Miroslav, GMK, Bílovec
3.–4. *Kuchař Martin*, G Karviná, Mírová
Zedník Lukáš, G Martinská čtvrť
5. *Slovák Petr*, G P. Bezruče, Frýdek-Místek
6.–7. *Hadamčík Lukáš*, Mendelovo G
Princ Tomáš, G Ostrava-Poruba, Čs. exilu
8.–11. *Jeziorský Tomáš*, GMK, Bílovec
Koval Jakub, G Karviná, Mírová
Tomala Jiří, GMK, Bílovec
Zezula Adam, G P. Bezruče, Frýdek-Místek

Kategorie P

1. *Tomáš Gavenčiak*, GMK, Bílovec
2. *Pavel Motloch*, G P. Bezruče, Frýdek-Místek
3. *Ondřej Májek*, GMK, Bílovec
4. *Michal Šenkýř*, GMK, Bílovec
5. *Vilém Šustr*, GMK, Bílovec

Výsledky celostátního kola 53. ročníku MO
kategorie A

Vítězové

1.–2.	*Alexandr Kazda*, 8/8, G Praha 6, Nad Alejí	42 b.
	Pavel Kocourek, 3/4, SPŠST, Praha 1, Panská	42 b.
3.–5.	*František Konopecký*, 7/8, GLJ Holešov	41 b.
	Jaromír Kuben, 2/4, G Brno, tř. Kpt. Jaroše	41 b.
	Jan Moláček, 4/4, GJKT, Hradec Králové	41 b.
6.	*Vítězslav Kala*, 4/4, G Brno, tř. Kpt. Jaroše	39 b.
7.–9.	*Sven Dražan*, 4/4, G Brno, tř. Kpt. Jaroše	38 b.
	Tomáš Gavenčiak, 4/4, GMK Bílovec	38 b.
	Marek Pechal, 6/8, G Zlín, Lesní čtvrť	38 b.

Další úspěšní řešitelé

10.–11.	*Jana Fabriková*, 4/4, G Brno, tř. Kpt. Jaroše	30 b.
	Jan Křetínský, 8/8, GML Brno, Žižkova	30 b.
12.–13.	*Jakub Opršal*, 2/4, G Brno, tř. Kpt. Jaroše	28 b.
	Michal Rychnovský, 3/4, G Brno, tř. Kpt. Jaroše	28 b.
14.–15.	*Stanislav Basovník*, 7/8, G Kroměříž, Masarykovo nám.	23 b.
	Jan Uhlík, 2/4, GMK Bílovec	23 b.
16.–17.	*Tomáš Hebelka*, 8/8, G Brno, Vídeňská	22 b.
	Alexandr Pícha, 2/4, G Brno, tř. Kpt. Jaroše	22 b.
18.	*Ondřej Křivánek*, 7/8, G Třebíč, Masarykovo nám.	19 b.
19.–21.	*Tereza Klimošová*, 8/8, G Lanškroun	18 b.
	Ivo Machek, 4/4, GJKT, Hradec Králové	18 b.
	Radek Moravec, 5/6, GBN, Hradec Králové	18 b.

Výsledky celostátního kola 53. ročníku MO
kategorie P

Vítězové

1.	*Daniel Marek*, 2/4, G Ch. Dopplera, Praha	36 b.
2.	*Martin Vejnár*, 3/4, G Brno, tř. Kpt. Jaroše	35 b.
3.–5.	*Jindřich Flídr*, 8/8, G Lanškroun	33 b.
	Kryštof Hoder, 4/4, G Brno, tř. Kpt. Jaroše	33 b.
	Petr Škoda, 8/8, G Praha, Ústavní	33 b.
6.–7.	*Tomáš Gavenčiak*, 4/4, G M. Koperníka, Bílovec	30 b.
	David Matoušek, 8/8, G Ch. Dopplera, Praha 5	30 b.

Další úspěšní řešitelé

8.–9.	*Ondřej Májek*, 4/4, G M. Koperníka, Bílovec	29 b.
	Oto Petřík, 6/8, G Vrchlabí	29 b.
10.	*Martin Dobroucký*, 7/8, G Moravská Třebová	28 b.
11.–12.	*Ondřej Bílka*, 2/4, G Zlín, Lesní čtvrť	26 b.
	Pavel Troubil, 4/4, G Brno, tř. Kpt. Jaroše	26 b.
13.–14.	*Zdeněk Sojka*, 4/4, SPŠST, Praha 1, Panská	24 b.
	Jiří Štěpánek, 4/4, G Brno, tř. Kpt. Jaroše	24 b.

Kategorie C

Texty úloh

C – I – 1

Dokažte, že pro každé přirozené číslo n, které je větší než 3 a není dělitelné třemi, platí: Šachovnici $n \times n$ lze rozřezat na jeden čtverec 1×1 a obdélníky 3×1. (*J. Zhouf*)

C – I – 2

Je dán obdélník $ABCD$. Nechť přímky p a q, které procházejí vrcholem A, protínají polokružnice vně připsané stranám BC a CD daného obdélníku po řadě v bodech K a L ($B \neq K \neq C \neq L \neq D$) a rovněž strany BC a CD po řadě v bodech P a Q tak, že trojúhelník ABP má stejný obsah jako trojúhelník KCP a zároveň trojúhelník AQD má stejný obsah jako trojúhelník CLQ. Dokažte, že body K, L, C leží na téže přímce.

(*J. Švrček*)

C – I – 3

Žák měl vypočítat příklad $X \cdot Y : Z$, kde X je dvojmístné číslo, Y trojmístné číslo a Z trojmístné číslo s číslicí 2 na místě jednotek. Výsledkem příkladu mělo být přirozené číslo. Žák však tečku přehlédl a součin $X \cdot Y$ chápal jako pětimístné číslo. Získal tak sedmkrát větší výsledek, než měl vyjít. Jaký příklad měl žák počítat? (*P. Černek*)

C – I – 4

Nechť P je libovolný vnitřní bod rovnostranného trojúhelníku ABC. Uvažujme obrazy K, L a M bodu P v osových souměrnostech s osami AB, BC a CA. Určete množinu všech bodů P takových, že trojúhelník KLM je rovnoramenný. (*J. Zhouf*)

C – I – 5

Přirozené číslo nazveme *magickým*, právě když je lze rozložit na součet dvou trojmístných čísel zapsaných stejnými číslicemi, ale v opačném pořadí. Například číslo 1 413 je magické, neboť platí $1\,413 = 756 + 657$; nejmenší magické číslo je 202.

a) Určete počet všech magických čísel.

b) Ukažte, že součet všech magických čísel je roven 187 000. (*J. Šimša*)

C – I – 6

Ze všech čtyřúhelníků, jež lze vepsat do kružnice o daném poloměru r a které mají dvě strany dané délky m, určete ten, který má největší obsah. (*P. Leischner*)

C – S – 1

Určete počet všech trojmístných čísel, která jsou devatenáctkrát větší než součet jejich číslic. (*J. Šimša*)

C – S – 2

Je dán čtverec o straně délky 5 cm. Mezi všemi čtyřúhelníky, které leží v tomto čtverci tak, že dvě jejich strany mají délku 2 cm a leží na hranici čtverce, určete všechny ty, které mají maximální obsah. (*P. Leischner*)

C – S – 3

Dlaždičky A složené ze tří jednotkových čtverců mají tvar ⌜, dlaždičky B složené ze čtyř jednotkových čtverců mají tvar ⊓. Kolik dlaždiček jednotlivých typů potřebujeme na vydlaždičkování čtverce o straně 6 jednotek? Pro každý možný počet dlaždiček uveďte příklad takového pokrytí. (*J. Földes*)

C – II – 1

V rovině je dán obdélník $ABCD$, kde $|AB| = a < b = |BC|$. Na jeho straně BC existuje bod K a na straně CD bod L tak, že daný obdélník je úsečkami AK, KL a LA rozdělen na čtyři navzájem podobné trojúhelníky. Určete hodnotu poměru $a : b$. (*J. Švrček*)

C – II – 2

Najděte všechny trojice prvočísel p, q a r, pro které platí

$$\frac{14}{p} + \frac{51}{q} = \frac{65}{r}.$$

(*P. Novotný*)

C – II – 3

Do kružnice o poloměru $r = 6$ vepište osmiúhelník $ABCDEFGH$, jehož strany AB, CD, EF a GH mají po řadě délky 3, 4, 5 a 6 a strany BC, DE, FG a HA jsou shodné. (*P. Novotný*)

C – II – 4

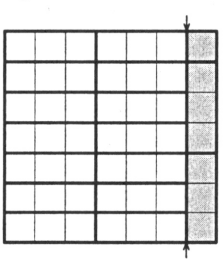
Žáci měli vypočítat příklad $x + y \cdot z$ pro trojmístné číslo x a dvojmístná čísla y a z. Martin umí násobit a sčítat čísla zapsaná v desítkové soustavě, zapomněl však na pravidlo o přednosti násobení před sčítáním. Proto mu sice vyšlo zajímavé číslo, které se čte stejně zleva doprava jako zprava doleva, správný výsledek byl ale o 2004 menší. Určete čísla x, y a z.

(*J. Šimša*)

C – I – 1

Budeme-li přemýšlet, jak navrhovat postupy řezání šachovnic velkých rozměrů, jistě nás napadne myšlenka, že na obdélníky 3×1 lze rozřezat každý „pás“ šachovnice tvořený třemi sousedními řádky nebo sloupci. Takové pásy se proto vyplatí od šachovnice opakovaně odřezávat (dokud je to možné), a tak zmenšovat její rozměry o násobky tří. Proto bude pro naši úlohu o šachovnici $n \times n$ výhodné rozlišit, zda dané číslo $n > 3$ dává při dělení třemi zbytek 1, anebo zbytek 2 (zbytek 0 je zadáním vyloučen). Každý z těchto případů prozkoumáme odděleně.

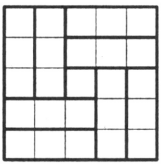
Případ $n = 3k + 1$. Nejprve z šachovnice $(3k + 1) \times (3k + 1)$ odřežeme pás prvních $3k$ sloupců, tedy obdélník $(3k + 1) \times 3k$, který pak rozřežeme (po trojicích sloupců) na k pásů $(3k + 1) \times 3$ a každý z nich konečně rozřežeme na $3k + 1$ obdélníků 1×3. Z původní šachovnice nám pak zůstane nerozřezán poslední sloupec; protože má $3k + 1$ polí, snadno ho rozřežeme na jeden čtverec 1×1 a k obdélníků 3×1. Na obr. 1 je znázorněno výsledné rozřezání šachovnice 7×7 (počáteční odřezání pásu 7×6 je vyznačeno šipkami, zbylý sloupec je šedý). Ze stejného obrázku nahlédneme i způsob řešení pro $n = 4$.



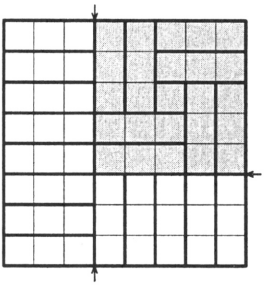
Obr. 1

Případ $n = 3k + 2$. Kdybychom šachovnici $(3k + 2) \times (3k + 2)$ důsledně „ořezávali“ postupem z úvodu řešení, dostali bychom (po oddělení dvou pásů $(3k + 2) \times 3k$ a $3k \times 2$) jako zbytek šachovnici 2×2, kterou však není možné rozřezat požadovaným způsobem (na díly 1×1 a 3×1). To je možné provést až s „následující“ šachovnicí 5×5, jak vidíme na obr. 2.

Zbývá popsat, jak každou větší šachovnici $(3k + 2) \times (3k + 2)$ řezáním zredukovat na právě posouzený čtverec 5×5. Nejprve oddělíme pás $(3k + 2) \times (3k - 3)$ tvořený prvními $(k - 1)$ trojicemi sloupců šachovnice;



Obr. 2



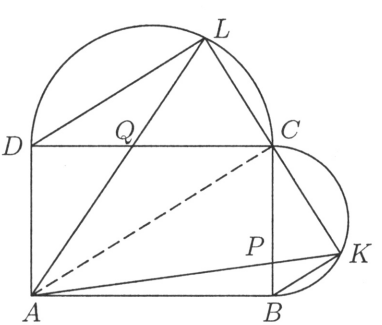
Obr. 3

ze zbylé šachovnice $(3k+2) \times 5$ pak oddělíme pás $(3k-3) \times 5$ tvořený jejími posledními $(k-1)$ trojicemi řádků, z původní šachovnice pak zbude kýžený čtverec 5×5 v pravém horním rohu (šedý na obr. 3 pro šachovnici 8×8).

Dodejme, že při řešení dané úlohy jsme nebrali v úvahu obarvení polí šachovnice. Barvy polí se uplatňují v jiných situacích, zejména tehdy, když potřebujeme dokázat, že rozřezání šachovnice na díly předepsaného tvaru není možné (doplňující úlohy 4 a 5).

C – I – 2

Trojúhelníky ABP a KCP mají podle zadání stejné obsahy; připojíme-li ke každému z nich trojúhelník ACP (obr. 4), usoudíme, že stejné obsahy mají i trojúhelníky ABC a AKC. Protože strana AC je oběma těmto trojúhelníkům společná, obě k ní příslušné výšky musí být shodné. Body B a K tudíž mají stejnou vzdálenost od přímky AC (a leží ve stejné polorovině touto přímkou určené). To znamená, že $BK \parallel AC$. Podle Thaletovy věty ovšem platí $BK \perp CK$, takže platí rovněž $AC \perp CK$.



Obr. 4

Podobně z rovnosti obsahů trojúhelníků AQD, CLQ a kolmosti přímek CL a DL odvodíme, že $AC \perp CL$. Dohromady to znamená, že úhel KCL je složen ze dvou pravých úhlů ACK a ACL. Body K a L tudíž leží na přímce, která prochází bodem C kolmo k úhlopříčce AC.

C – I – 3

Protože Y je trojmístné číslo, pětimístné číslo se zápisem XY je číslo $1\,000X+Y$. Žák tedy počítal příklad $(1\,000X+Y) : Z$ a podle textu úlohy mu v porovnání s původním příkladem vyšel sedmkrát větší výsledek, tedy

$$\frac{1\,000X+Y}{Z} = 7 \cdot \frac{X \cdot Y}{Z}.$$

Odtud po násobení číslem Z dostaneme rovnici $1\,000X + Y = 7XY$, kterou vyřešíme vzhledem k neznámé Y:

$$Y = \frac{1\,000X}{7X-1}.$$

Pro která X je poslední zlomek celočíselný? Jinak vyjádřeno: kdy je číslo $1\,000X$ dělitelné číslem $7X-1$? Protože čísla X a $7X-1$ jsou nesoudělná (nesoudělná jsou totiž dvě po sobě jdoucí čísla $7X-1$ a $7X$), hledáme ta X, pro která číslo $7X-1$ dělí číslo $1\,000$. Abychom nemuseli vypisovat všechny dělitele čísla $1\,000$, uvědomíme si, že X je dvojmístné, tudíž $69 \leqq 7X-1 \leqq 692$. Rozložme proto číslo $1\,000$ všemi způsoby na součin dvou činitelů tak, aby jeden (řekněme první) z činitelů byl z intervalu $\langle 69, 692\rangle$:

$$1\,000 = 500 \cdot 2 = 250 \cdot 4 = 200 \cdot 5 = 125 \cdot 8 = 100 \cdot 10.$$

Z rovnic

$$7X-1 = 500,\ 7X-1 = 250,\ 7X-1 = 200,\ 7X-1 = 125,\ 7X-1 = 100$$

má jedině rovnice $7X-1 = 125$ celočíselné řešení $X = 18$, pro něž vychází $Y = 1\,000X/(7X-1) = 1\,000 \cdot 18/125 = 144$.

Nyní určíme neznámé číslo Z. Využijeme k tomu podmínku úlohy, že hodnota výrazu $X \cdot Y : Z$ je přirozené číslo. Protože $X = 18$ a $Y = 144$, jedná se o číslo $18 \cdot 144 : Z$, tedy číslo $2^5 \cdot 3^4 : Z$. Takové číslo je celé, právě když má číslo Z rozklad na prvočinitele tvaru $2^a 3^b$, kde $0 \leqq a \leqq 5$ a $0 \leqq b \leqq 4$. Exponenty a, b najdeme z podmínky, že číslo $Z = 2^a 3^b$ je

podle zadání trojmístné a na místě jednotek má číslici 2. Protože $3^4 = 81$ a $2^5 \cdot 3 = 96$, musí být $a \geqq 1$ a $b \geqq 2$. Všechna čísla $2^a 3^b$, kde $a \in \{1, 2, 3, 4, 5\}$ a $b \in \{2, 3, 4\}$, teď vypíšeme do tabulky:

b \ a	1	2	3	4	5
2	18	36	72	144	288
3	54	108	216	432	864
4	162	324	648	1 296	2 592

Z vypočtených čísel mají požadovanou vlastnost pouze čísla $Z = 432 = 2^4 3^3$ a $Z = 162 = 2^1 3^4$.

Odpověď: Úloha má dvě řešení. Žák měl počítat buď příklad $18 \cdot 144 : 432$, nebo příklad $18 \cdot 144 : 162$.

Jiné řešení. Jako v prvním řešení odvodíme vyjádření

$$Y = \frac{1\,000X}{7X - 1},$$

tentokrát však získaný zlomek upravíme částečným vydělením čísla 1 000 číslem 7. Na základě rovnosti $1\,000 = 7 \cdot 143 - 1$ dostáváme

$$Y = \frac{1\,000X}{7X - 1} = \frac{143(7X - 1) + 143 - X}{7X - 1} = 143 + \frac{143 - X}{7X - 1}.$$

Aby bylo Y celé, musí být poslední zlomek $(143 - X)/(7X - 1)$ celočíselný. Protože číslo X je dvojmístné, náš zlomek splňuje odhady

$$\frac{143 - 99}{7 \cdot 99 - 1} < \frac{143 - X}{7X - 1} < \frac{143 - 10}{7 \cdot 10 - 1}.$$

Levý zlomek je roven 44/692, pravý je roven 133/69, takže jediná možná celočíselná hodnota prostředního zlomku je rovna 1. Musí tedy být $Y = 144$. Rovnice

$$\frac{143 - X}{7X - 1} = 1$$

pak má jediné řešení $X = 18$. Dále už postupujeme jako v prvním řešení.

Další řešení. Dříve získanou rovnici $1\,000X + Y = 7XY$ upravíme do součinového tvaru $Y = X \cdot (7Y - 1\,000)$. Musí proto platit $7Y - 1\,000 > 0$, odkud

$$Y > \frac{1\,000}{7} > 142, \quad \text{neboli} \quad Y \geqq 143.$$

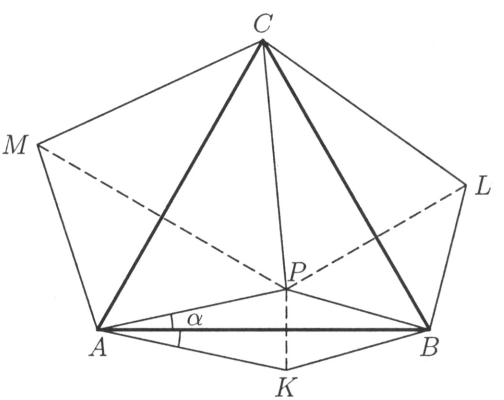
Číslo X je dvojmístné, proto z rovnosti $Y = X \cdot (7Y - 1\,000)$ vychází odhad

$$Y \geqq 10 \cdot (7Y - 1\,000), \quad \text{neboli} \quad Y \leqq \frac{10\,000}{69} < 145.$$

Dohromady dostáváme, že číslo Y je rovno jednomu z čísel 143 nebo 144. Rovnice $143 = X \cdot (7 \cdot 143 - 1\,000)$ má řešení $X = 143$, což ovšem není dvojmístné číslo; rovnice $144 = X \cdot (7 \cdot 144 - 1\,000)$ má řešení $X = 18$. Tak jsme znovu ukázali, že $X = 18$ a $Y = 144$; číslo Z určíme jako v prvním řešení.

C – I – 4

Označme $\alpha = |\sphericalangle BAP|$, $0° < \alpha < 60°$ (obr. 5). Protože úhly BAP a BAK



Obr. 5

jsou souměrně sdružené podle osy AB, platí rovněž $|\sphericalangle BAK| = \alpha$. Protože $|\sphericalangle CAP| = |\sphericalangle CAB| - |\sphericalangle BAP| = 60° - \alpha$, ze souměrnosti podle osy CA plyne rovnost $|\sphericalangle CAM| = 60° - \alpha$. Pro velikost úhlu KAM tudíž platí

$$|\sphericalangle KAM| = |\sphericalangle BAK| + |\sphericalangle BAC| + |\sphericalangle CAM| = \alpha + 60° + (60° - \alpha) = 120°.$$

Ze souměrností podle os AB a CA rovněž plynou rovnosti $|AK| = |AP| = |AM|$. Proto je trojúhelník KAM rovnoramenný a jeho úhel při hlavním vrcholu A má velikost 120°. Podobně se zdůvodní, proč i trojúhelníky LBK a MCL jsou rovnoramenné a jejich vnitřní úhly při hlavních vrcholech B a C mají velikost 120°.

Při posuzování podmínky, že trojúhelník KLM je rovnoramenný, musíme rozlišit, které z jeho stran KL, LM, MK jsou shodné. S ohledem na symetrii rozebereme podrobně pouze případ, kdy $|KL| = |MK|$. Z podobných rovnoramenných trojúhelníků KAM a LBK vyplývá, že jejich základny MK a KL jsou shodné, právě když jsou shodná jejich ramena AK a BK. Zapišme to pomocí délek úseček: rovnost $|KL| = |MK|$ platí, právě když platí rovnost $|AK| = |BK|$, neboli rovnost $|AP| = |BP|$. Poslední rovnost ovšem nastane, právě když bod P leží na ose strany AB. Obdobně se zjistí podmínky ekvivalentní rovnostem $|MK| = |LM|$ a $|KL| = |LM|$.

Odpověď: Trojúhelník KLM je rovnoramenný, právě když bod P leží na aspoň jedné z os stran daného rovnostranného trojúhelníku ABC. Hledaná množina je proto sjednocením tří úseček – výšek trojúhelníku ABC (bez jejich krajních bodů).

C – I – 5

Na příkladu čísla 1 413 vidíme, že někdy není snadné poznat, zda dané trojmístné či čtyřmístné číslo je magické či nikoliv. Podíváme se proto nejdříve, jak se magické číslo x vyjádří pomocí číslic těch trojmístných čísel $\overline{abc}$ a $\overline{cba}$, jejichž je součtem:

$$x = \overline{abc} + \overline{cba} = (100a + 10b + c) + (100c + 10b + a) = 101(a + c) + 20b.$$

Vidíme, že číslo x je určeno číslicemi a, b, c tak, že závisí jen na b a na součtu $a + c$. Znamená to, že různé trojice číslic a, b, c mohou určovat totéž magické číslo x (nemyslíme tím pouze trojice lišící se vzájemnou výměnou číslic a a c). Je-li např. $a + c = 14$ a $b = 9$, najdeme tři různá vyjádření magického čísla 1 594:

$$1\,594 = 599 + 995 = 698 + 896 = 797 + 797.$$

Existují ještě jiná „magická“ vyjádření čísla 1 594? Vše závisí na tom, zda jsou rovnicí $1\,594 = 101s + 20b$ hodnoty součtu číslic $s = a + c$ a číslice b jednoznačně určeny. Z rovnice ihned vidíme, že číslo s končí číslicí 4, takže $s = 4$ nebo $s = 14$ (jiné hodnoty součtu $s = a + c$ nejsou číslicemi a, c dosažitelné). Zatímco hodnotě $s = 14$ odpovídá (jak dobře víme) hodnota $b = 9$, pro $s = 4$ dostaneme rovnici $1\,594 = 404 + 20b$, která nemá celočíselné řešení.

Poučeni uvedeným příkladem, pokusíme se stanovit počet magických čísel jako počet čísel tvaru $x = 101s + 20b$, kde číslo s (rovné součtu číslic a a c, jež jsou *nenulové*) probíhá množinu $\{2, 3, 4, \ldots, 18\}$, zatímco číslice b probíhá (nezávisle na součtu s) množinu $\{0, 1, 2, \ldots, 9\}$. Protože číslo s nabývá celkem 17 různých hodnot a číslo b celkem 10 různých hodnot, je počet všech dvojic (s, b), které můžeme do vzorce $x = 101s + 20b$ dosadit, roven číslu $17 \cdot 10 = 170$. Ukážeme-li nyní, že po dosazení libovolných dvou různých dvojic (s_1, b_1) a (s_2, b_2) dostaneme dvě různá magická čísla

$$x_1 = 101s_1 + 20b_1 \quad \text{a} \quad x_2 = 101s_2 + 20b_2,$$

bude to znamenat, že počet všech hodnot x (tedy *počet všech magických čísel*) je rovněž roven číslu 170.

Připusťme, že pro některé dvojice (s_1, b_1) a (s_2, b_2) platí $x_1 = x_2$. Rovnost $101s_1 + 20b_1 = 101s_2 + 20b_2$ upravíme do tvaru $101(s_1 - s_2) = 20(b_2 - b_1)$, z něhož vzhledem k nesoudělnosti čísel 20 a 101 vyplývá, že číslo $b_2 - b_1$ je násobkem čísla 101. Musí jít přitom o nulový násobek, neboť $|b_2 - b_1| \leqq 9$ (b_1 a b_2 jsou číslice!). Platí tedy $b_2 - b_1 = 0$, takže rovněž $s_1 - s_2 = 0$, což dohromady znamená, že dvojice (s_1, b_1) a (s_2, b_2) jsou stejné. Jen v tomto případě je tedy rovnost $x_1 = x_2$ možná.

Součet všech magických čísel (tedy čísel tvaru $x = 101s + 20b$) určíme výhodně, když čísla nejprve uspořádáme do obdélníkového schématu (podle stejných hodnot s do řádků a podle stejných hodnot b do sloupců)

$$\begin{array}{ccccc}
101 \cdot 2 + 20 \cdot 0 & 101 \cdot 2 + 20 \cdot 1 & 101 \cdot 2 + 20 \cdot 2 & \ldots & 101 \cdot 2 + 20 \cdot 9 \\
101 \cdot 3 + 20 \cdot 0 & 101 \cdot 3 + 20 \cdot 1 & 101 \cdot 3 + 20 \cdot 2 & \ldots & 101 \cdot 3 + 20 \cdot 9 \\
101 \cdot 4 + 20 \cdot 0 & 101 \cdot 4 + 20 \cdot 1 & 101 \cdot 4 + 20 \cdot 2 & \ldots & 101 \cdot 4 + 20 \cdot 9 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
101 \cdot 17 + 20 \cdot 0 & 101 \cdot 17 + 20 \cdot 1 & 101 \cdot 17 + 20 \cdot 2 & \ldots & 101 \cdot 17 + 20 \cdot 9 \\
101 \cdot 18 + 20 \cdot 0 & 101 \cdot 18 + 20 \cdot 1 & 101 \cdot 18 + 20 \cdot 2 & \ldots & 101 \cdot 18 + 20 \cdot 9
\end{array}$$

a pak čísla sečteme buď po sloupcích, nebo po řádcích. Rozhodneme se pro sčítání po sloupcích, přitom budeme brát v úvahu, o kolik se čísla uvažovaného sloupce liší od příslušných čísel prvního sloupce. Součet čísel v prvním sloupci je

$$101 \cdot (2 + 3 + \ldots + 18) = 101 \cdot 170,$$

ve druhém sloupci je součet $101 \cdot 170 + 17 \cdot 20 \cdot 1$, ve třetím $101 \cdot 170 + 17 \cdot 20 \cdot 2$, atd. až v posledním (desátém) sloupci je součet čísel roven $101 \cdot 170 + 17 \cdot 20 \cdot 9$. Součet všech magických čísel je tedy roven

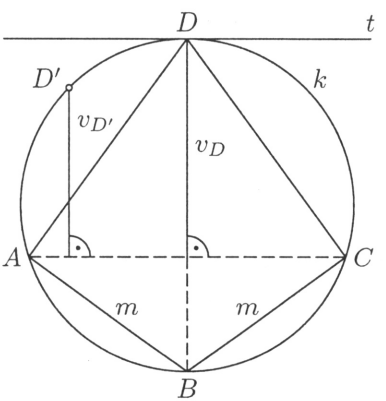
$$10 \cdot 101 \cdot 170 + 17 \cdot 20 \cdot (1 + 2 + \ldots + 9) = 187\,000.$$

C – I – 6

V celém řešení budeme předpokládat, že dané délky m a r splňují nerovnost $m < 2r$, jinak žádný čtyřúhelník požadovaných vlastností neexistuje. Strany délky m každého takového čtyřúhelníku jsou totiž tětivami kružnice o poloměru r a nejvýše jedna z nich může být jejím průměrem.

Zkoumané čtyřúhelníky rozdělíme do dvou skupin podle toho, zda jsou jejich strany dané délky m sousední, nebo protilehlé.

Libovolný čtyřúhelník z první skupiny označíme $ABCD$ tak, aby platilo $|AB| = |BC| = m$. Úhlopříčka rozdělí tento tětivový čtyřúhelník na dva trojúhelníky ABC a ACD (obr. 6), přitom je jasné, že první z nich,

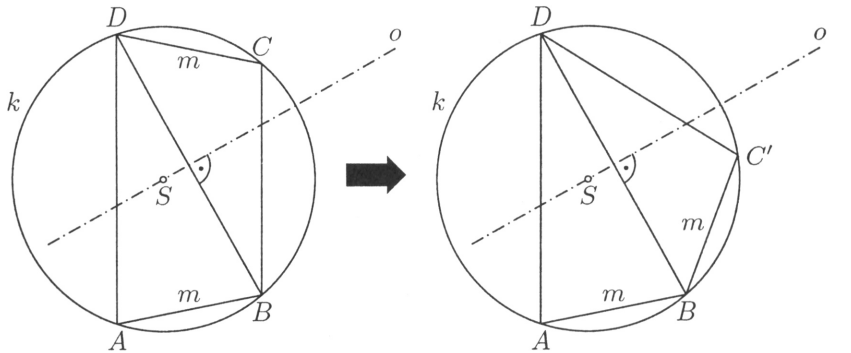


$v_{D'} < v_D$

Obr. 6

trojúhelník ABC, je poloměrem r opsané kružnice k a délkou m dvou jeho stran určen (až na shodnost) jednoznačně, takže má pevně určený obsah. Proto bude obsah takového čtyřúhelníku $ABCD$ maximální, právě když bude maximální obsah trojúhelníku ACD. Tento trojúhelník má určenou délku strany AC, takže jeho obsah bude maximální, právě když bude maximální jeho výška v_D z vrcholu D. Při pevné poloze trojúhelníku ABC bod D probíhá ten oblouk AC kružnice k, jenž neobsahuje bod B, takže výška v_D je zřejmě největší, právě když bod D je středem tohoto oblouku, leží tedy (stejně jako bod B) na ose úsečky AC. (Tvrzení zdůvodníme pomocí tečny t ke kružnici k, jež prochází nalezeným bodem D rovnoběžně s přímkou AC, obr. 6). Tak docházíme k závěru, že v první skupině má maximální obsah ten čtyřúhelník, který je deltoid (je-li $m \neq r\sqrt{2}$), respektive čtverec (je-li $m = r\sqrt{2}$).

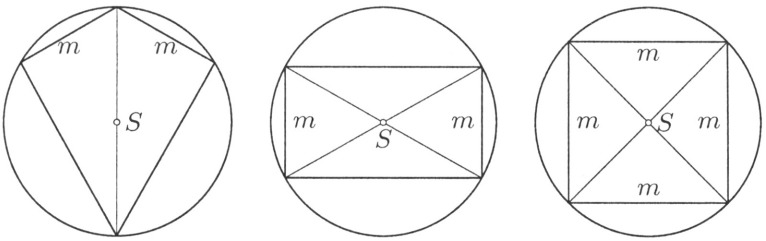
Přejděme nyní ke čtyřúhelníkům druhé skupiny. Libovolný z nich označme $ABCD$ tak, aby platilo $|AB| = |CD| = m$ (obr. 7).



Obr. 7

Obrázek ukazuje, jak k takovému čtyřúhelníku $ABCD$ sestrojit pomocný čtyřúhelník $ABC'D$, který má stejný obsah jako $ABCD$, je vepsán do téže kružnice k a má sousední strany AB a BC' dané délky m. Konstrukci teď popíšeme a zmíněné vlastnosti čtyřúhelníku $ABC'D$ podrobně zdůvodníme. Bod C' sestrojíme jako obraz bodu C v souměrnosti podle osy o úsečky BD; protože je kružnice k souměrná podle osy každé své tětivy, platí $C' \in k$. Trojúhelníky BCD a $DC'B$ jsou souměrně sdružené podle osy o, takže mají stejný obsah, tudíž stejný obsah mají i čtyřúhelníky $ABCD$ a $ABC'D$. Ze zmíněné souměrnosti rovněž plynou rovnosti $|CD| = |BC'|$ a $|BC| = |DC'|$, takže čtyřúhelníky $ABCD$ a $ABC'D$ se liší pouze „prohozením" dvou sousedních stran. Tím jsou potřebné vlastnosti čtyřúhelníku $ABC'D$ zdůvodněny. Jak už víme z předchozího odstavce, čtyřúhelník $ABC'D$ má největší možný obsah, právě když platí rovnost $|C'D| = |AD|$, kterou můžeme přepsat jako rovnost $|BC| = |AD|$. Ta nastane, právě když je čtyřúhelník $ABCD$ rovnoběžník (neboť od počátku předpokládáme, že $|AB| = |CD|$). Každý rovnoběžník vepsaný do kružnice je ale pravoúhelník (součet protilehlých vnitřních úhlů tětivového čtyřúhelníku je 180°, takové úhly jsou ale v případě rovnoběžníku shodné, a tedy pravé). Shrňme výsledek tohoto odstavce: ve druhé skupině čtyřúhelníků má maximální obsah ten čtyřúhelník, který je obdélník (je-li $m \neq r\sqrt{2}$), respektive čtverec (je-li $m = r\sqrt{2}$).

Celkový závěr: Hledané čtyřúhelníky s maximálním obsahem tvoří v případě $m < 2r$, $m \neq r\sqrt{2}$, dvě skupiny: skupinu shodných deltoidů



Obr. 8

a skupinu shodných obdélníků; v případě $m = r\sqrt{2}$ jsou všechny hledané čtyřúhelníky shodné čtverce (obr. 8). (V případě $m \geqq 2r$ je množina uvažovaných čtyřúhelníků prázdná.)

C – S – 1

Trojmístné číslo se zápisem $\overline{abc}$ má požadovanou vlastnost, právě když jeho číslice a, b, c splňují rovnost

$$100a + 10b + c = 19(a + b + c), \quad \text{neboli} \quad 9a = b + 2c.$$

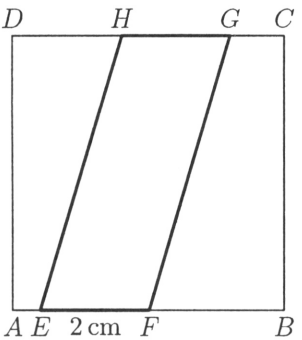
Protože $b \leqq 9$ a $c \leqq 9$, platí nerovnost $b + 2c \leqq 27$. Z rovnosti $9a = b + 2c$ proto plyne odhad $a \leqq 3$, takže platí $a \in \{1, 2, 3\}$ (číslice $a = 0$ není na začátku zápisu povolena). Pro $a = 1$ dostáváme rovnici $9 = b + 2c$, ze které plyne $c \leqq 4$; pro každé takové $c \in \{0, 1, 2, 3, 4\}$ je číslice b určena rovností $b = 9 - 2c$. Proto s číslicí $a = 1$ existuje právě 5 vyhovujících čísel. Právě tolik je i vyhovujících čísel s číslicí $a = 2$: z rovnice $18 = b + 2c$ totiž plyne $c \in \{5, 6, 7, 8, 9\}$ a $b = 18 - 2c$. Konečně pro $a = 3$ z rovnice $27 = b + 2c$ plyne $b = c = 9$. Hledaný počet čísel je tedy $5 + 5 + 1 = 11$.

Jiné řešení. Součet číslic libovolného trojmístného čísla nepřevyšuje číslo 27, jehož devatenáctinásobek je 513. Proto každé vyhovující číslo nepřevyšuje 513, takže součet jeho číslic je nejvýše $4 + 9 + 9 = 22$. Protože nejmenší trojmístný násobek čísla 19 je číslo $114 = 19 \cdot 6$, bude úloha vyřešena, když zjistíme, kolik čísel tvaru $19s$, kde $s \in \{6, 7, 8, \ldots, 22\}$, má součet číslic rovný právě číslu s. Rutinní prověrkou zjistíme, že ze zmíněných 17 čísel vyhovují právě čísla 114, 133, 152, 171, 190, 209, 228, 247, 266, 285 a 399. Těchto čísel je 11.

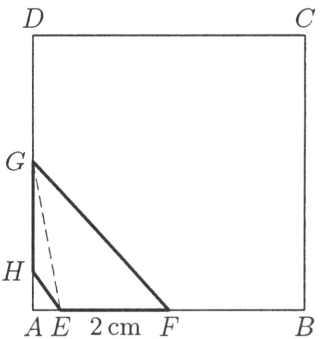
C – S – 2

Čtyřúhelník $EFGH$ můžeme do daného čtverce $ABCD$ umístit třemi způsoby:

1. Dvě strany délky 2 cm leží na protilehlých stranách daného čtverce (obr. 9). Obsah každého takového čtyřúhelníku (rovnoběžníku) je $S = 5 \cdot 2\,\mathrm{cm}^2 = 10\,\mathrm{cm}^2$.



Obr. 9

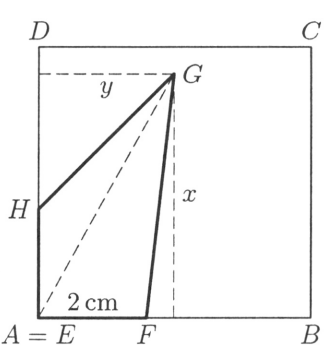


Obr. 10

2. Obě strany délky 2 cm leží na sousedních stranách daného čtverce a přitom jsou protilehlými stranami čtyřúhelníku $EFGH$ (obr. 10). Obsah takového čtyřúhelníku je

$$S = \frac{1}{2}|EF| \cdot |AG| + \frac{1}{2}|GH| \cdot |AE| = \frac{1}{2} \cdot 2\,\mathrm{cm} \cdot |AG| + \frac{1}{2} \cdot 2\,\mathrm{cm} \cdot |AE| \leqq$$
$$\leqq \big(5 + (5-2)\big)\,\mathrm{cm}^2 = 8\,\mathrm{cm}^2 < 10\,\mathrm{cm}^2.$$

3. Obě strany délky 2 cm leží na sousedních stranách daného čtverce a přitom jsou sousedními stranami čtyřúhelníku $EFGH$ (obr. 11). Označíme-li po řadě x a y vzdálenosti bodu G od stran AB a AD (tedy



Obr. 11

výšku trojúhelníku EFG na stranu EF a výšku trojúhelníku EHG na stranu EH), je obsah takového čtyřúhelníku

$$S = \frac{1}{2}|EF| \cdot x + \frac{1}{2}|AH| \cdot y \leqq 2 \cdot \frac{1}{2} \cdot 2 \cdot 5 = 10\,\mathrm{cm}^2.$$

Přitom rovnost nastane, právě když $x = y = 5\,\mathrm{cm}$, tj. právě když $G = C$.

Závěr: Největší možný obsah ($10\,\mathrm{cm}^2$) mají všechny rovnoběžníky, jejichž dvě strany délky 2 cm leží na protějších stranách daného čtverce, a čtyři deltoidy, jejichž jedna úhlopříčka je zároveň úhlopříčkou daného čtverce.

C – S – 3

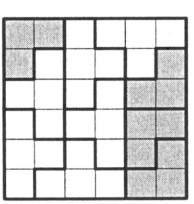
Předpokládejme, že čtverec o straně 6 jednotek je vydlaždičkován a dlaždičkami A a b dlaždičkami B (nevylučujeme případ, že $a = 0$ nebo $b = 0$). Pro obsah vydlaždičkované plochy pak platí rovnost $36 = 3a+4b$, ze které plyne, že číslo a je násobkem čtyř (a číslo b násobkem tří). Proto má rovnice $36 = 3a + 4b$ v oboru celých nezáporných čísel za řešení pouze tyto dvojice (a, b): $(0, 9)$, $(4, 6)$, $(8, 3)$ a $(12, 0)$. Posoudíme dále, zda pro jednotlivé dvojice (a, b) je příslušné vydlaždičkování daného čtverce možné.

(i) 9 dlaždiček B. Vysvětlíme, proč takové vydlaždičkování neexistuje. Obarvěme jednotkové čtverečky celého čtverce jako obvyklou šachovnici; získáme 18 černých a 18 bílých „polí". Každá dlaždička B pokrývá tři pole jedné barvy a jedno pole druhé barvy. Připusťme, že celý čtverec pokrývá 9 dlaždiček B, přitom právě x z nich má tu vlastnost, že pokrývají po 3 černých polích, takže $9 - x$ z nich má tu vlastnost, že pokrývají po 1 černém poli. Pro celkový počet černých polí pak platí rovnost $18 = 3x + (9 - x)$, odkud $x = 9/2$, což je spor.

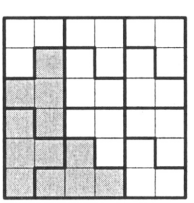
(ii) 4 dlaždičky A a 6 dlaždiček B. Možné řešení vidíte na obr. 12.

(iii) 8 dlaždiček A a 3 dlaždičky B. Možné řešení vidíte na obr. 13.

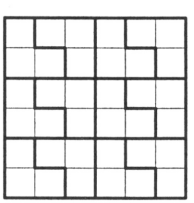
(iv) 12 dlaždiček A. Možné řešení vidíte na obr. 14.



Obr. 12



Obr. 13

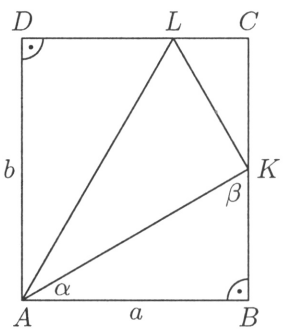


Obr. 14

Poznámka. Uveďme ještě jiný argument, proč nelze devíti dlaždičkami B vyplnit uvažovaný čtverec. Dlaždička, která pokrývá rohové pole, může být umístěna (až na souměrnost podle úhlopříčky čtverce) jediným způsobem, např. tak jako dlaždička B v levém dolním rohu čtverce na obr. 13, pak ale dlaždička B, která v takovém případě pokrývá druhé pole zleva v dolní řadě, musí být v poloze jako na obrázku. Poslední dvě pole dolní řady pak už jednou ani dvěma dlaždičkami B pokrýt nelze.

C – II – 1

V pravoúhlém trojúhelníku ABK označme $\alpha = |\sphericalangle BAK|$, $\beta = |\sphericalangle AKB| = 90° - \alpha$ (obr. 15). Stejné vnitřní úhly 90°, α, β mají



Obr. 15

i trojúhelníky AKL a ADL, neboť jsou dle zadání trojúhelníku ABK podobné. Všimněme si jejich (ostrých) úhlů u společného vrcholu A. Protože $|\sphericalangle KAD| = 90° - \alpha = \beta$, jsou oba úhly KAL a LAD menší než β, takže se rovnají úhlu α. Pravý úhel BAD je tedy polopřímkami AK, AL rozdělen na tři shodné úhly velikosti α, odkud $\alpha = 30°$ (a $\beta = 60°$). Z pravoúhlých trojúhelníků ADL a ABK pak vyplývá, že $|AK| = |AB|/\cos 30° = 2a/\sqrt{3}$ a $|AL| = |AD|/\cos 30° = 2b/\sqrt{3}$. Odtud s ohledem na podmínku $a < b$ plyne nerovnost $|AK| < |AL|$, tudíž přepona v trojúhelníku AKL je AL (delší z obou stran AK, AL). Pro poměr délek odvěsny AK a přepony AL pak platí $\cos 30° = |AK| : |AL| = a : b$, takže $a : b = \sqrt{3} : 2$.

Úlohu lze řešit mnoha obměněnými postupy, například rozlišit dva případy, kdy trojúhelník KAL má pravý úhel při vrcholu K respektive L, a v každém z nich vyjádřit vnitřní úhly všech čtyř podobných trojúhel-

níků (ve druhém případě pak ale vyjde $a : b = 2 : \sqrt{3} > 1$, což odporuje zadání úlohy).

C – II – 2

Všimněme si nejdříve, že pro čitatele zlomků z dané rovnice platí vztah $14 + 51 = 65$. Proto je řešením každá trojice stejných prvočísel $p = q = r$ a navíc pro libovolné řešení platí: jsou-li některá dvě z čísel p, q, r stejná, je stejné i třetí číslo. Budeme tedy dále předpokládat, že prvočísla p, q, r splňující danou rovnici jsou navzájem různá (a tedy navzájem nesoudělná).

Po vynásobení rovnice součinem pqr dostaneme

$$14qr + 51pr = 65pq,$$

odkud vzhledem ke zmíněné nesoudělnosti plyne

$$p \mid 14 = 2 \cdot 7, \quad q \mid 51 = 3 \cdot 17 \quad \text{a} \quad r \mid 65 = 5 \cdot 13.$$

To znamená, že $p \in \{2, 7\}$, $q \in \{3, 17\}$ a $r \in \{5, 13\}$. Nyní můžeme sestavit a do rovnice dosadit všech osm možných trojic (p, q, r); zjistíme tak, že vyhovuje jedině trojice $(7, 17, 13)$.

Prověrku dosazováním můžeme zkrátit tak, že vyloučíme kteroukoliv z hodnot $p = 2, q = 3$, resp. $r = 5$. Například po dosazení $r = 5$ dostaneme po vydělení pěti rovnici $14q + 51p = 13pq$, která nemá celočíselné řešení p ani pro $q = 3$ $(14 + 17p = 13p)$, ani pro $q = 17$ $(14 + 3p = 13p)$. Jiná možnost: z rovnice $14qr + 51pr = 65pq$ plyne $2p(q - r) = 7(2qr + 7pr - 9pq)$, takže součin $p(q - r)$ je dělitelný sedmi. Protože však $q \in \{3, 17\}$ a $r \in \{5, 13\}$ (viz výše), není rozdíl $q - r$ dělitelný sedmi, proto je sedmi dělitelné číslo p. Podobně lze zdůvodnit, proč $17 \mid q$ a $13 \mid r$.

Jiné řešení. Z dané rovnice vyjádříme r pomocí p a q:

$$r = \frac{65pq}{51p + 14q} = \frac{5 \cdot 13 \cdot p \cdot q}{51p + 14q}.$$

V posledním zlomku jsme zvýraznili rozklad čitatele na (čtyři) prvočinitele. Takový zlomek bude roven některému prvočíslu r, právě když jeho jmenovatel bude součinem tří prvočinitelů z čitatele (jiné krácení zlomku není možné). Hledáme tedy situace, kdy platí některý z případů:

$$\begin{aligned} 51p + 14q &= 5 \cdot 13 \cdot p \quad \text{a} \quad r = q, \\ 51p + 14q &= 5 \cdot 13 \cdot q \quad \text{a} \quad r = p, \\ 51p + 14q &= 5 \cdot p \cdot q \quad \text{a} \quad r = 13, \\ 51p + 14q &= 13 \cdot p \cdot q \quad \text{a} \quad r = 5. \end{aligned}$$

Snadnou úpravou rovnic zjistíme, že první dva případy nastanou pouze v situaci, kdy $p = q$ (tehdy ovšem rovněž $p = r$). Poslední dva případy vedou k vyjádřením

$$q = \frac{3 \cdot 17 \cdot p}{5p - 14}, \quad \text{resp.} \quad q = \frac{3 \cdot 17 \cdot p}{13p - 14},$$

ze kterých analogickou úvahou o krácení zlomků (případ $p = q$ již můžeme vynechat) s přihlédnutím k zřejmým nerovnostem $5p - 14 < 17p$ a $13p - 14 < 17p$ dostaneme rovnice

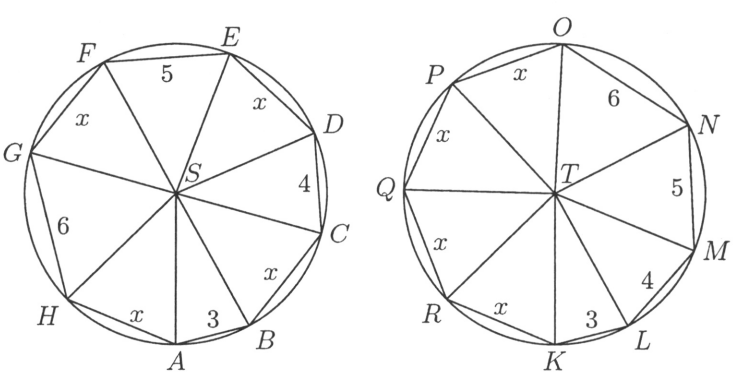
$$5p - 14 = 3p, \quad \text{resp.} \quad 13p - 14 = 3p.$$

První rovnice má řešení $p = 7$ (kterému odpovídá $q = 17$ a $r = 13$), druhá rovnice celočíselné řešení nemá.

Odpověď: Všechna řešení (p, q, r) jsou trojice (p, p, p), kde p je libovolné prvočíslo, a trojice $(7, 17, 13)$.

C – II – 3

Rozbor: Kromě hledaného osmiúhelníku $ABCDEFGH$ uvážíme ještě pomocný osmiúhelník $KLMNOPQR$, který je rovněž vepsán do kružnice o poloměru $r = 6$ a jehož strany splňují podmínky: $|KL| = 3$, $|LM| = 4$, $|MN| = 5$, $|NO| = 6$, $|OP| = |PQ| = |QR| = |RK|$ (obr. 16). Označ-



Obr. 16

me S, resp. T střed kružnice s vepsaným osmiúhelníkem $ABCDEFGH$, resp. $KLMNOPQR$. Podle věty sss platí shodnosti $\Delta ABS \simeq \Delta KLT$,

$\Delta CDS \simeq \Delta LMT$, $\Delta EFS \simeq \Delta MNT$, $\Delta GHS \simeq \Delta NOT$, a proto jsou shodné středové úhly ASB a KTL, CSD a LTM, ESF a MTN, GSH a NTO. Dále podle věty *sss* jsou shodné trojúhelníky BCS, DES, FGS a HAS, stejně jako trojúhelníky OPT, PQT, QRT a RKT. Ze shodnosti jejich úhlů při hlavním vrcholu S, resp. T proto plyne

$$\begin{aligned}|\sphericalangle BSC| &= \frac{1}{4}(360^\circ - |\sphericalangle ASB| - |\sphericalangle CSD| - |\sphericalangle ESF| - |\sphericalangle GSH|) = \\ &= \frac{1}{4}(360^\circ - |\sphericalangle KTL| - |\sphericalangle LTM| - |\sphericalangle MTN| - |\sphericalangle NTO|) = \\ &= |\sphericalangle OTP|.\end{aligned}$$

Využili jsme toho, že středy S a T jsou *vnitřními* body obou osmiúhelníků (tudíž součet všech osmi středových úhlů je v obou případech 360°), neboť v opačném případě by jeden z osmi středových úhlů byl roven součtu sedmi ostatních; musel by to být úhel příslušný tětivě délky 6, ten je však zřejmě menší než součet úhlů příslušných tětivám délek 3, 4 a 5. Trojúhelníky BCS a OPT jsou proto shodné podle věty *sus*, tudíž čtveřice shodných stran obou osmiúhelníků mají jednu společnou délku. Dokážeme-li proto sestrojit pomocný osmiúhelník $KLMNOPQR$, je konstrukce osmiúhelníku $ABCDEFGH$ nasnadě.

Konstrukce: Na libovolné kružnici $t(T; 6)$ sestrojíme v jednom směru body K, L, M, N a O tak, aby $|KL| = 3$, $|LM| = 4$, $|MN| = 5$ a $|NO| = 6$. Úhel KTO (ten, který neobsahuje body L, M, N) pak rozdělíme na čtyři shodné díly: nejprve sestrojíme průsečík Q kružnice t s osou úhlu KTO, pak průsečíky P, R kružnice t s osami úhlů OTQ resp. QTK. Poté přistoupíme ke konstrukci hledaného osmiúhelníku $ABCDEFGH$: na kružnici $k(S, 6)$ zvolíme bod A a pak na ní v jednom směru sestrojíme postupně body B, C, ..., H tak, aby $|AB| = 3$, $|BC| = |OP|$, $|CD| = 4$, $|DE| = |OP|$, $|EF| = 5$, $|FG| = |OP|$, $|GH| = 6$.

Důkaz správnosti: Ze shodnosti sedmi dvojic trojúhelníků $\Delta ABS \simeq \simeq \Delta KLT$, $\Delta BCS \simeq \Delta OPT$, ..., $\Delta GHS \simeq \Delta NOT$ plyne shodnost úhlů HSA a RTK, a tedy i shodnost osmé dvojice trojúhelníků $\Delta HAS \simeq \Delta RKT$. Proto mají délky stran sestrojeného osmiúhelníku $ABCDEFGH$ (shodné se stranami $KLMNOPQR$) všechny potřebné vlastnosti.

Poznámka. O osmiúhelníku $KLMNOPQR$ jsme nemuseli v celém řešení vůbec mluvit a vést úvahy takto: úhly shodné se středovými úhly ASB, CSD, ESF, GSH dokážeme sestrojit, pro společnou velikost ω

shodných středových úhlů BSC, DSE, FSG a HSA pak platí rovnice

$$4\omega + |\sphericalangle ASB| + |\sphericalangle CSD| + |\sphericalangle ESF| + |\sphericalangle GSH| = 360°, \qquad (1)$$

kterou lze snadno konstrukčně vyřešit; osmiúhelník $KLMNOPQR$ je ovšem k tomuto účelu ideální pomůckou.

C – II – 4

Martin vypočítal hodnotu $(x+y)z$ místo $x+yz$, takže podle zadání platí

$$(x+y)z - (x+yz) = 2\,004, \quad \text{neboli} \quad x \cdot (z-1) = 2\,004 = 12 \cdot 167,$$

přičemž 167 je prvočíslo. Činitele x a $z-1$ určíme, když si uvědomíme, že z je dvojmístné číslo, takže $9 \leqq z-1 \leqq 98$. Vidíme, že nutně $z-1 = 12$ a $x = 167$, odkud $z = 13$. Martin tedy vypočítal číslo $V = (167+y) \cdot 13$. Číslo V je tedy čtyřmístné, a poněvadž se čte odpředu stejně jako odzadu, má tvar $\overline{abba} = 1\,001a + 110b$. Protože $1\,001 = 13 \cdot 77$, musí platit rovnost $(167+y) \cdot 13 = 13 \cdot 77a + 110b$, z níž plyne, že číslice b je dělitelná třinácti, takže $b = 0$. Po dosazení dostaneme (po dělení třinácti) rovnost $167 + y = 77a$, která s ohledem na nerovnosti $10 \leqq y \leqq 99$ znamená, že číslice a se rovná 3, tudíž $y = 64$.

V druhé části řešení jsme mohli postupovat rovněž následovně. Pro číslo $V = (167+y) \cdot 13$ vycházejí z nerovností $10 \leqq y \leqq 99$ odhady $2\,301 \leqq V \leqq 3\,458$. Zjistíme proto, která z čísel $\overline{2bb2}$, kde $b \in \{3,4,5,6,7,8,9\}$, a čísel $\overline{3bb3}$, kde $b \in \{0,1,2,3,4\}$, jsou dělitelná třinácti. I když lze těchto dvanáct čísel rychle otestovat na kalkulačce, udělejme to obecně jejich částečným vydělením třinácti:

$$\overline{2bb2} = 2\,002 + 110b = 13 \cdot (154 + 8b) + 6b,$$
$$\overline{3bb3} = 3\,003 + 110b = 13 \cdot (231 + 8b) + 6b.$$

Vidíme, že vyhovuje jedině číslo $\overline{3bb3}$ pro $b = 0$, kdy $167 + y = 231$, takže $y = 64$.

Odpověď: Žáci měli počítat příklad $167 + 64 \cdot 13$, tedy $x = 167$, $y = 64$ a $z = 13$.

Kategorie B

Texty úloh

B – I – 1

Každou z hvězdiček na místě jednotek čísel ve výrazu

$$\left| \frac{777\,777\,777\,77*}{777\,777\,777\,77*} - \frac{555\,555\,555\,554}{555\,555\,555\,559} \right|$$

nahraďte nějakou číslicí tak, aby výraz nabyl co nejmenší hodnoty.

(*J. Šimša*)

B – I – 2

V rovnoramenném lichoběžníku $ABCD$ platí $|BC| = |CD| = |DA|$ a $|\sphericalangle DAB| = |\sphericalangle ABC| = 36°$. Na základně AB je dán bod K tak, že $|AK| = |AD|$. Dokažte, že kružnice opsané trojúhelníkům AKD a KBC mají vnější dotyk. (*J. Zhouf*)

B – I – 3

V oboru reálných čísel řešte rovnici

$$x\lfloor x \rfloor - 5x + 7 = 0,$$

kde $\lfloor x \rfloor$ znamená dolní celou část čísla x, tedy největší celé číslo k, pro něž platí $k \leqq x$. (Například $\lfloor \sqrt{2} \rfloor = 1$ a $\lfloor -3{,}1 \rfloor = -4$.) (*E. Kováč*)

B – I – 4

Číslo a_n vznikne tak, že za sebe napíšeme prvních n po sobě jdoucích přirozených čísel, například $a_{13} = 12\,345\,678\,910\,111\,213$. Určete, kolik čísel dělitelných 24 se nachází mezi čísly $a_1, a_2, \ldots, a_{10\,000}$. (*P. Černek*)

B – I – 5

Je dána přímka p a mimo ni bod A. Sestrojte lichoběžník $ABCD$ s minimálním obsahem a ramenem BC na přímce p tak, aby $|BC| = |AC|$ a průsečík E jeho úhlopříček splňoval vztah $|BE| = 3|DE|$.

(*P. Leischner*)

B – I – 6

Určete všechna přirozená čísla M dělitelná 240, pro která má rovnice $M = \mathrm{NSN}(x, y)$ s neznámými x a y právě 1 001 řešení v oboru přirozených čísel. (Symbol $\mathrm{NSN}(x, y)$ značí nejmenší společný násobek čísel x a y.)

(*P. Černek*)

B – S – 1

Zjistěte, kolik řešení má v oboru reálných čísel rovnice

$$x = \lfloor x \rfloor + \frac{x}{2\,004},$$

kde $\lfloor x \rfloor$ označuje největší celé číslo, které nepřevyšuje číslo x.

(*J. Šimša*)

B – S – 2

Uveďte příklad množiny M dvojmístných čísel, jež má maximální počet prvků a přitom splňuje obě následující podmínky:

(i) Každá dvě čísla z M jsou nesoudělná.

(ii) Změníme-li pořadí číslic libovolného čísla z M, dostaneme opět číslo z množiny M.

(*J. Földes*)

B – S – 3

Je dán lichoběžník $ABCD$ s ostrými úhly při základně AB. Na ní existuje bod E takový, že kružnice opsané trojúhelníkům AED a EBC mají vnější dotyk. Dokažte, že bod E leží na kružnici opsané trojúhelníku CDV, kde V je průsečík přímek AD a BC.

(*R. Horenský*)

B – II – 1

Číslo a_n vznikne tak, že za sebe zapíšeme prvních n druhých mocnin po sobě jdoucích přirozených čísel. Např. $a_{11} = 149\,162\,536\,496\,481\,100\,121$.

Zjistěte, kolik čísel dělitelných dvanácti je mezi čísly $a_1, a_2, \ldots, a_{100\,000}$.
(*P. Černek*)

B – II – 2

Najděte všechny kvadratické trojčleny $ax^2 + bx + c$ takové, že pokud libovolný z koeficientů a, b, c zvětšíme o 1, dostaneme nový kvadratický trojčlen, který bude mít dvojnásobný kořen. (*E. Kováč*)

B – II – 3

Pro dané přirozené číslo n řešte v oboru kladných reálných čísel rovnici

$$\lfloor x\sqrt{n^2 - 1} \rfloor = nx - 1.$$

(Symbol $\lfloor r \rfloor$ označuje největší celé číslo, jež nepřevyšuje číslo r.)
(*J. Šimša*)

B – II – 4

Je dán ostroúhlý trojúhelník VBA. Sestrojte tečnový čtyřúhelník $ABCD$ s minimálním obsahem tak, aby jeho vrcholy C a D ležely po řadě na polopřímkách opačných k polopřímkám BV a AV. (*P. Leischner*)

B – I – 1

Označme x a y číslice, které doplníme do čitatele, resp. jmenovatele prvního zlomku. Protože celý výraz v absolutní hodnotě budeme algebraicky upravovat, kvůli přehlednějším zápisům zavedeme označení $N = 111\,111\,111\,110$. Jednotlivá čísla z daného výrazu pak mají vyjádření:

$$\begin{aligned} 777\,777\,777\,77x &= 7N + x, \\ 777\,777\,777\,77y &= 7N + y, \\ 555\,555\,555\,554 &= 5N + 4, \\ 555\,555\,555\,559 &= 5N + 9. \end{aligned}$$

Zkoumaný výraz pak lze zapsat a upravit následujícím způsobem:

$$\begin{aligned} \left|\frac{7N+x}{7N+y} - \frac{5N+4}{5N+9}\right| &= \frac{|(7N+x)(5N+9) - (5N+4)(7N+y)|}{(7N+y)(5N+9)} = \\ &= \frac{|(35N^2 + 5xN + 63N + 9x) - (35N^2 + 5yN + 28N + 4y)|}{(7N+y)(5N+9)} = \\ &= \frac{|5 \cdot (7 - y + x) \cdot N + 9x - 4y|}{(7N+y)(5N+9)}. \end{aligned}$$

Označme ještě čitatele a jmenovatele získaného zlomku:

$$C = |5 \cdot (7 - y + x) \cdot N + 9x - 4y| \quad \text{a} \quad J = (7N + y)(5N + 9).$$

Budeme-li za x, y dosazovat různé dvojice číslic, jmenovatel J bude nabývat pouze deseti různých hodnot v rozmezí

$$(7N + 0)(5N + 9) \leqq J \leqq (7N + 9)(5N + 9).$$

Podívejme se nyní, jak velkých či malých hodnot bude nabývat čitatel C. Protože číslo $9x - 4y$ je nejvýše dvojmístné, zatímco číslo N dvanáctimístné, řád čitatele C bude záviset na tom, zda bude činitel $(7 - y + x)$ roven nule či nikoli. Proto tyto dvě možnosti posoudíme odděleně.

A. Případ $7 - y + x = 0$. Tehdy platí $y = x + 7$ a zkoumaný čitatel C je tvaru

$$C = |5 \cdot 0 \cdot N + (9x - 4y)| = |9x - 4(x + 7)| = |5x - 28|.$$

Jelikož číslice y (rovná $x+7$) je nejvýše 9, je číslice x rovna 0, nebo 1, nebo 2, takže výraz $|5x-28|$ se rovná 28, nebo 23, nebo 18. *Nejmenší* hodnota čitatele C je tudíž rovna 18 a dosáhneme ji jedině pro $x=2$ a $y=9$. Šťastnou „shodou okolností" má zrovna pro $y=9$ jmenovatel J *největší* hodnotu, takže

$$\min\left\{\frac{C}{J}\right\}=\frac{18}{(7N+9)(5N+9)}.$$

B. Případ $7-y+x\neq 0$. Ukažme, že hodnoty čitatele C (tudíž i hodnoty zlomku C/J) jsou v tomto případě „obrovské" ve srovnání s případem A. Z nerovnosti $7-y+x\neq 0$ plyne odhad $|7-y+x|\geqq 1$ (číslo $7-y+x$ je celé), tudíž máme

$$C=|5\cdot(7-y+x)\cdot N+9x-4y|\geqq 5\cdot|7-y+x|\cdot N-|9x-4y|\geqq 5N-|9x-4y|.$$

Protože x a y jsou číslice, platí zřejmě $|9x-4y|\leqq 81$. Z posledního odhadu C a maximální hodnoty J proto plyne nerovnost

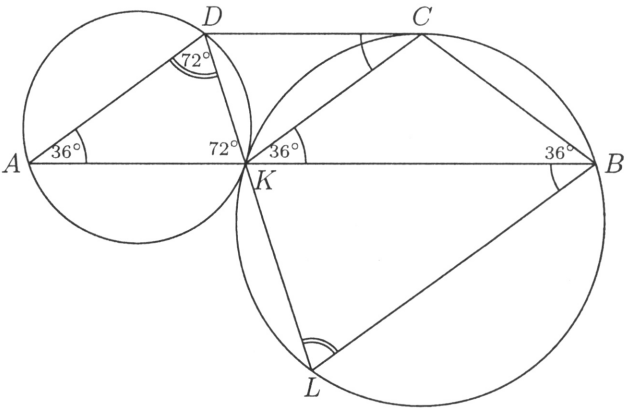
$$\frac{C}{J}\geqq\frac{5N-81}{(7N+9)(5N+9)}.$$

Poslední zlomek je „mnohokrát" větší než zlomek v závěru případu A, neboť oba zlomky mají stejný jmenovatel, zatímco srovnání čitatelů zřejmě dopadá takto: $5N-81\gg 18$ (nerovnost $5N-81>18$ platí již od hodnoty $N=20$).

Závěr: Do čitatele doplníme číslici $x=2$, do jmenovatele číslici $y=9$.

B – I – 2

V rovnoramenném trojúhelníku AKD známe úhel DAK proti základně KD. Můžeme dopočítat zbylé dva úhly při základně (obr. 17): $|\sphericalangle ADK|=|\sphericalangle AKD|=\frac{1}{2}(180^\circ-|\sphericalangle DAK|)=72^\circ$. Čtyřúhelník $AKCD$ má protější strany AK a CD shodné a rovnoběžné, takže se jedná o rovnoběžník, tudíž přímky KC a AD jsou rovnoběžné. Úhly DAK a CKB jsou tedy souhlasné a úhly CKB a KCD střídavé, proto $|\sphericalangle CKB|=|\sphericalangle KCD|=36^\circ$. Úhel DKC doplňuje úhly AKD a CKB do přímého úhlu, jeho velikost je tedy $|\sphericalangle DKC|=180^\circ-36^\circ-72^\circ=72^\circ$.

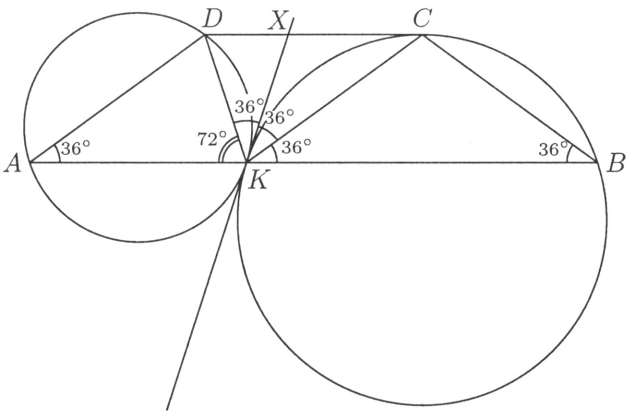


Obr. 17

Na polopřímce opačné k polopřímce KD zvolme bod L tak, že $|KL| = |AD|$. Potom $|\sphericalangle LKB| = |\sphericalangle AKD| = 72°$ a $|\sphericalangle CKL| = |\sphericalangle LKB| + |\sphericalangle CKB| = 108°$. Dopočítaním úhlů v lichoběžníku $ABCD$ dostáváme $|\sphericalangle BCD| = \frac{1}{2}(360° - 2 \cdot 36°) = 144°$ a můžeme vyjádřit velikost úhlu BCK: $|\sphericalangle BCK| = |\sphericalangle BCD| - |\sphericalangle KCD| = 144° - 36° = 108°$. Nyní již víme, že $|KL| = |CB|$ a $|\sphericalangle LKC| = |\sphericalangle KCB|$, což znamená, že $LBCK$ je rovnoramenný lichoběžník, a lze mu tedy opsat kružnici (shodnou s kružnicí opsanou trojúhelníku KBC). Dále můžeme z lichoběžníku $LBCK$ dopočítat $|\sphericalangle KLB| = \frac{1}{2}(360° - 2 \cdot 108°) = 72° = |\sphericalangle KDA|$. Z této rovnosti plyne, že $AD \parallel BL$, takže trojúhelníky ADK a BLK jsou vzájemně stejnolehlé podle středu K. Stejnolehlé jsou potom i kružnice jim opsané. Protože obě procházejí středem K zmíněné stejnolehlosti, mají v tomto bodě vnější dotyk.

Jiné řešení. Stejně jako v prvním řešení zjistíme, že $|AKD| = 72°$. Čtyřúhelník $AKCD$ je rovnoběžník (obr. 18), takže $|CK| = |AD|$. Z rovnosti $|CK| = |BC|$ v trojúhelníku KBC usoudíme, že $|\sphericalangle CKB| = |\sphericalangle KBC| = 36°$. Proto na základně CD existuje bod X tak, že $|AKX| = 108°$ (a $|\sphericalangle BKX| = 72°$). Pak $|\sphericalangle DKX| = |\sphericalangle AKX| - |\sphericalangle AKD| = 108° - 72° = 36°$, a tedy $|\sphericalangle DKX| = |\sphericalangle DAK|$, takže úhel DKX je úsekovým úhlem příslušným oblouku DAK v kružnici opsané trojúhelníku AKD, to znamená, že přímka KX je její tečnou. Podobně $|\sphericalangle CKX| = |\sphericalangle BKX| - |\sphericalangle BKC| = 72° - 36° = 36° = |\sphericalangle KBC|$, takže KX je i tečnou ke kružnici opsané trojúhelníku KBC. Kružnice opsané trojúhelníkům AKD a KBC mají tedy společnou tečnu KX

procházející společným bodem K. Obě kružnice se tudíž v tomto bodě dotýkají.



Obr. 18

B – I – 3

Označme $k = \lfloor x \rfloor$, tedy $x = k + \alpha$, $0 \leqq \alpha < 1$. Daná rovnice má potom tvar $(k+\alpha)k - 5(k+\alpha) + 7 = 0$. Odtud $\alpha = \dfrac{k^2 - 5k + 7}{5 - k}$. Hledáme tedy celá čísla k, pro která platí

$$0 \leqq \frac{k^2 - 5k + 7}{5 - k} < 1. \qquad (*)$$

Každou z těchto nerovností vyšetříme odděleně. Protože kvadratický trojčlen $k^2 - 5k + 7$ má záporný diskriminant, platí $k^2 - 5k + 7 \geqq 0$ pro každé $k \in \mathbb{R}$, takže levá nerovnost v $(*)$ platí, právě když $5 - k > 0$, neboli $k < 5$. Vyřešme pravou nerovnici:

$$\begin{aligned}
\frac{k^2 - 5k + 7}{5 - k} &< 1, \\
\frac{k^2 - 5k + 7 - (5 - k)}{5 - k} &< 0, \\
\frac{k^2 - 4k + 2}{5 - k} &< 0, \\
\frac{(k - 2 - \sqrt{2})(k - 2 + \sqrt{2})}{5 - k} &< 0.
\end{aligned}$$

Podle polohy čísel $2-\sqrt{2}$, $2+\sqrt{2}$ a 5 na číselné ose zjistíme, že poslední nerovnost platí, právě když $k \in (2-\sqrt{2}, 2+\sqrt{2}) \cup (5, \infty)$. Nerovnosti $(*)$ tedy platí současně, právě když $k \in (2-\sqrt{2}, 2+\sqrt{2})$. Této podmínce vyhovují pouze tři celá čísla $k \in \{1, 2, 3\}$. Pro $k = 1$ dopočteme $\alpha = \frac{3}{4}$, pro $k = 2$ vyjde $\alpha = \frac{1}{3}$ a pro $k = 3$ je $\alpha = \frac{1}{2}$. Celkem dostáváme tři řešení $x_1 = \frac{7}{4}$, $x_2 = \frac{7}{3}$, $x_3 = \frac{7}{2}$.

Jiné řešení. Jako v prvním řešení označíme $k = \lfloor x \rfloor$ a z rovnice $kx - 5x + 7 = 0$ vyjádříme x ve tvaru $x = \dfrac{7}{5-k}$. Nyní hledáme celá čísla k, pro která platí $k \leqq \dfrac{7}{5-k} < k+1$. Obě nerovnice jsou splněny jedině pro celá $k \in \{1, 2, 3\}$, kterým odpovídají kořeny $x \in \{\frac{7}{4}, \frac{7}{3}, \frac{7}{2}\}$.

B – I – 4

Přirozené číslo je dělitelné číslem 24, právě když je dělitelné současně (navzájem nesoudělnými) čísly 3 a 8. Pro ciferný součet přirozeného čísla k zaveďme označení $S(k)$. Číslo a_n je dělitelné třemi, právě když je třemi dělitelný jeho ciferný součet, tedy číslo $S(1) + S(2) + \ldots + S(n)$. Zbytek po dělení třemi tohoto součtu závisí pouze na zbytcích (po dělení třemi) jednotlivých sčítanců $S(k)$. Protože při dělení třemi dává číslo $S(k)$ stejný zbytek jako číslo k (viz návodnou úlohu 1), dávají čísla $S(1), S(4), S(7), \ldots$ zbytek 1, čísla $S(2), S(5), S(8), \ldots$ zbytek 2 a čísla $S(3), S(6), S(9), \ldots$ zbytek 0. Proto například číslo $S(a_{14})$, tedy součet $S(1) + S(2) + \ldots + S(14)$, dává při dělení třemi stejný zbytek jako součet

$$(1+2+0) + (1+2+0) + (1+2+0) + (1+2+0) + 1 + 2.$$

Podle uzávorkovaných trojic snadno vidíme, že tento součet je dělitelný třemi. Protože obecně součet $S(3k-2) + S(3k-1) + S(3k)$ je dělitelný třemi pro každé přirozené k, můžeme obdobným způsobem uzávorkovat každý součet

$$S(1) + S(2) + \ldots + S(n)$$

a zjistit, že jeho zbytek při dělení třemi

- je roven 1, je-li $n = 3k - 2$;
- je roven 0, je-li $n = 3k - 1$ nebo $n = 3k$.

Čísla a_n tedy budou dělitelná třemi, právě když n bude tvaru $3k$ nebo $3k - 1$ $(k = 1, 2, \ldots)$.

Nyní rozeberme, kdy budou čísla a_n navíc dělitelná osmi. Přirozené číslo je dělitelné osmi, právě když je dělitelné osmi poslední trojčíslí jeho

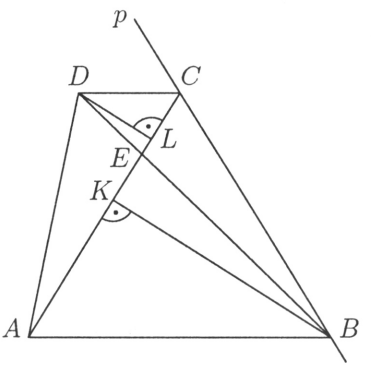
zápisu v desítkové soustavě. Naše úvahy budou tedy záviset na počtu číslic čísla n:

- ▷ Alespoň trojmístná n. Pro taková n je tedy a_n dělitelné osmi, právě když je dělitelné osmi číslo n.
  Protože se zbytky čísel a_n po dělení třemi opakují po třech, zbytky po dělení osmi po osmi číslech a_n, budou se zbytky po dělení číslem 24 opakovat po nejmenším společném násobku těchto period, tedy po dvaceti čtyřech. Pro trojmístná n snadno zjistíme, že podmínce v úloze vyhovují čísla tvaru $104 + 24k$ a $120 + 24k$ (n musí být dělitelné osmi a dávat zbytek dva nebo nula po dělení třemi). Do 10 000 máme 413 čísel tvaru $104 + 24k$ ($413 = \lfloor \frac{1}{24}(10\,000 - 104) \rfloor + 1$) a 412 čísel tvaru $120 + 24k$.
- ▷ Dvojmístná n. Aby bylo číslo a_n dělitelné osmi, musí být dělitelné čtyřmi. O dělitelnosti čtyřmi rozhoduje poslední dvojčíslí, takže čtyřmi budou dělitelná právě všechna ta a_n, pro která je n dělitelné čtyřmi. Číslo $n-1$ je pak liché, tedy i a_{n-1} je číslo liché a číslo $100a_{n-1}$ dává zbytek čtyři po dělení osmi. Potom číslo $a_n = 100a_{n-1} + n$ bude dělitelné osmi, právě když n bude také dávat zbytek čtyři po dělení osmi, bude tedy tvaru $8k + 4$. Spolu s podmínkou na dělitelnost třemi dostáváme, že vyhovující dvojmístná čísla n mají (stejně jako výše) periodu 24 a jsou tvaru $n = 12 + 24k$ a $n = 20 + 24k$, $k \in \{0, 1, 2, 3\}$. Do sta to máme $4 + 4 = 8$ čísel.
- ▷ Jednomístná n. Snadno zjistíme, že ze všech sudých čísel a_n pro $n \leqq 8$ vyhovuje pouze $a_6 = 123\,456$.
  Celkem vyhovuje 834 čísel.

B – I – 5

Předpokládejme, že $ABCD$ je hledaný lichoběžník a K, L jsou paty kolmic z vrcholů B, D na přímku AC (obr. 19). Z podobnosti pravoúhlých trojúhelníků BKE a DLE plyne, že délky stran BK a DL, tedy odvěsen ve zmíněných trojúhelnících, jsou ve stejném poměru jako délky jejich přepon BE a DE, tedy $3 : 1$. BK a DL jsou však i výšky v trojúhelnících ABC a ACD, a to na společnou stranu AC. Obsahy těchto trojúhelníků jsou tedy také v poměru $3 : 1$, takže obsah lichoběžníku $ABCD$ je roven $\frac{4}{3}P$, kde P je obsah rovnoramenného trojúhelníku ABC. Výška tohoto trojúhelníku z bodu A na stranu BC je dána (vzdálenost bodu A od přímky p). Obsah trojúhelníku ABC bude tedy minimální,

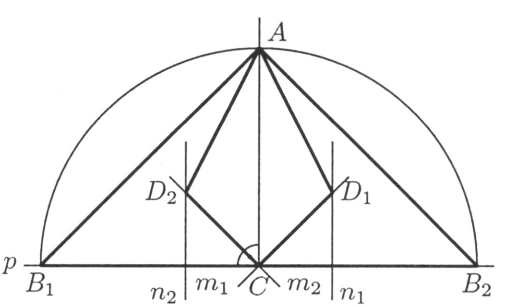
bude-li minimální délka strany BC, tedy i AC, tedy když úsečka AC bude kolmá na p.



Obr. 19

Konstrukce. Nejprve sestrojíme bod C (pata kolmice z A na p). Vrchol B nalezneme jako průsečík přímky p s kružnicí $k(C, |AC|)$ (dvě možnosti). Vrchol D je průsečíkem přímky m, vedené bodem C rovnoběžně s AB, a přímky n rovnoběžné s AC ve vzdálenosti $\frac{4}{3}|BC|$ od vrcholu B uvnitř poloroviny opačné k ACB.

Úloha má celkem dvě řešení souměrně sdružená podle přímky $AC \perp p$ (obr. 20).



Obr. 20

B – I – 6

Nejprve ukažme, že pro číslo M s prvočíselným rozkladem $M = \prod_{i=1}^{n} p_i^{c_i}$ (p_i jsou různá prvočísla) je počet řešení rovnice $\mathrm{NSN}(x, y) = M$ roven $\prod_{i=1}^{n} (2c_i + 1)$. Vskutku, každé řešení (x, y) dané rovnice má tu vlastnost, že libovolné prvočíslo p_i ($i = 1, \ldots, n$) dělí alespoň jedno z čísel x a y (a to nejvýše v takové mocnině, v jaké dělí M) a žádná jiná prvočísla už ani x, ani y nedělí; x a y jsou tedy tvaru $x = \prod_{i=1}^{n} p_i^{a_i}$, $y = \prod_{i=1}^{n} p_i^{b_i}$, $a_i, b_i \in \mathbb{N}_0$, a navíc $\max(a_i, b_i) = c_i$, $i = 1, \ldots, n$. Čísla x a y tak jednoznačně určují n-tice čísel a_i a b_i a obráceně jsou jimi jednoznačně určena. Všechna řešení dané rovnice jsou tedy popsána dvojicemi n-tic přirozených čísel takových, že na i-té pozici je v obou n-ticích číslo z množiny $\{0, \ldots, c_i\}$ a alespoň v jedné z nich se přímo rovná c_i. Takových n-tic je $\prod_{i=1}^{n} (2c_i + 1)$: Dvě n-tice čísel $(a_1, a_2, \ldots, a_n)$ a $(b_1, b_2, \ldots, b_n)$ můžeme uvážit jako n dvojic čísel (a_1, b_1), (a_2, b_2), …, (a_n, b_n). Libovolná dvojice (a_i, b_i) může nezávisle nabývat $(2c_i+1)$ různých hodnot $(0, c_i)$, $(1, c_i)$, …, $(c_i - 1, c_i)$, (c_i, c_i), $(c_i, c_i - 1)$, …, $(c_i, 1)$, $(c_i, 0)$. Podle kombinatorického pravidla součinu dostáváme výše uvedený počet.

Prvočíselný rozklad čísla 1 001 je $7 \cdot 11 \cdot 13$. Aby měla daná rovnice právě 1 001 řešení, musí exponenty c_i z prvočíselného rozkladu čísla M (obsahujícího dle zadání nejméně tři prvočísla, a to 2, 3 a 5) vyhovovat rovnici $\prod_{i=1}^{n} (2c_i + 1) = 7 \cdot 11 \cdot 13$. V prvočíselném rozkladu čísla M tedy musí být zastoupena právě tři prvočísla, a to v mocninách $\frac{1}{2}(7 - 1) = 3$, $\frac{1}{2}(11 - 1) = 5$ a $\frac{1}{2}(13 - 1) = 6$. Protože M má být dělitelné číslem $240 = 2^4 \cdot 3 \cdot 5$, tedy prvočísly 2, 3 a 5 v odpovídajících mocninách, jsou jediné možné volby pro M čísla $2^5 \cdot 3^3 \cdot 5^6$, $2^5 \cdot 3^6 \cdot 5^3$, $2^6 \cdot 3^5 \cdot 5^3$, $2^6 \cdot 3^3 \cdot 5^5$.

B – S – 1

Předpokládejme na okamžik, že celé číslo $k = \lfloor x \rfloor$ známe, dosaďme je do rovnice jako „parametr“ a získanou rovnici vyřešme:

$$\begin{aligned} x &= k + \frac{x}{2\,004}, \\ 2\,004x &= 2\,004k + x, \\ x &= \frac{2\,004k}{2\,003}. \end{aligned}$$

Budeme-li do posledního vzorce dosazovat jednotlivá celá čísla k, bude příslušné x skutečně řešením zkoumané rovnice, bude-li se jeho celá část rovnat právě číslu k, budou-li tedy platit nerovnosti

$$k \leqq \frac{2\,004k}{2\,003} < k + 1.$$

Zjistíme, která celá k vyhovují oběma nerovnostem. Levá nerovnost je ekvivalentní s nerovností $k \geqq 0$, pravá nerovnost s nerovností $k < 2\,003$. Hledaná k jsou tedy právě hodnoty $k \in \{0, 1, \ldots, 2\,002\}$, každá z nich určuje jediné řešení x, takže všech řešení x zadané rovnice je právě 2 003. Dodejme, že vyhovující k lze určit rovněž úpravou odvozeného vzorce do tvaru

$$x = \frac{2\,004k}{2\,003} = k + \frac{k}{2\,003},$$

z něhož je vidět, že číslo k je celou částí čísla x, právě když platí nerovnosti

$$0 \leqq \frac{k}{2\,003} < 1, \quad \text{neboli} \quad 0 \leqq k < 2\,003.$$

Jiné řešení. Protože pro každé reálné x platí $\lfloor x \rfloor \leqq x \leqq \lfloor x \rfloor + 1$, porovnáním se zadanou rovnicí dodejme k zjištění, že každé řešení x musí splňovat nerovnosti

$$0 \leqq \frac{x}{2\,004} < 1, \quad \text{neboli} \quad 0 \leqq x < 2\,004.$$

Číslo x splňující poslední nerovnosti bude řešením zkoumané rovnice, právě když hodnota $x - \dfrac{x}{2\,004}$ bude celočíselná. Protože platí

$$x - \frac{x}{2\,004} = \frac{2\,003x}{2\,004},$$

lze poslední podmínku vyslovit takto: číslo $2\,003x$ je celočíselným násobkem čísla 2 004. To s ohledem na nerovnosti $0 \leqq 2\,003x < 2\,003 \cdot 2\,004$ znamená, že číslo $2\,003x$ je rovno některému z čísel

$$0 \cdot 2\,004,\ 1 \cdot 2\,004,\ 2 \cdot 2\,004,\ \ldots,\ 2\,002 \cdot 2\,004,$$

takže zkoumaná rovnice má právě 2 003 řešení

$$\frac{0 \cdot 2\,004}{2\,003},\ \frac{1 \cdot 2\,004}{2\,003},\ \frac{2 \cdot 2\,004}{2\,003},\ \ldots,\ \frac{2\,002 \cdot 2\,004}{2\,003}.$$

B – S – 2

Kvůli podmínce (i) může být v množině M nejvýše jedno z čísel $11, 22, 33, \ldots, 99$ zapsaných dvěma stejnými číslicemi, která jsou vesměs dělitelná jedenácti. Kvůli podmínce (ii) a dělitelnosti dvěma tam zas nesmí být žádné číslo zapsané dvěma různými sudými číslicemi; s jednou sudou číslicí může být v M nejvýše jedna dvojice čísel $\overline{ab}$, $\overline{ba}$.

Zbývá posoudit, kolik může množina M obsahovat dvojic čísel $\overline{ab}$, $\overline{ba}$ zapsaných dvěma různými lichými číslicemi a a b. Žádné z těchto čísel nesmí být dělitelné třemi (je-li číslo $\overline{ab}$ dělitelné třemi, je takové i číslo $\overline{ba}$), proto v úvahu připadá pouze sedm dvojic takových čísel: $(13, 31)$, $(17, 71)$, $(19, 91)$, $(35, 53)$, $(37, 73)$, $(59, 95)$ a $(79, 97)$. Kvůli dělitelnosti pěti, sedmi a devatenácti však může být v M pouze jedna z dvojic $(19, 91)$, $(35, 53)$ a $(59, 95)$, tedy nejvýše pět ze všech sedmi vypsaných dvojic.

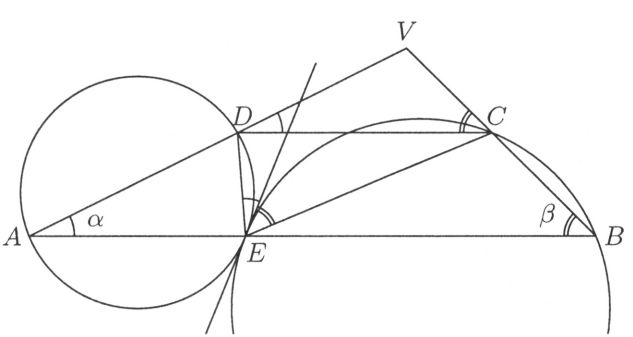
Celkově zjišťujeme, že množina M obsahuje nejvýše $1 + 2 + 2 \cdot 5 = 13$ čísel. Příkladem třináctiprvkové množiny je

$$\mathsf{M} = \{11, 23, 32, 13, 31, 17, 71, 35, 53, 37, 73, 79, 97\}.$$

(Existují i jiné příklady, naše úvahy však ukazují, že každá třináctiprvková množina M musí obsahovat čísla 13, 31, 17, 71, 37, 73, 79, 97 a jednu z dvojic $(35, 53)$ nebo $(59, 95)$; dvojice $(19, 91)$ je vyloučena, neboť číslo 91 je násobkem čísla 13.)

B – S – 3

Označme α a β po řadě vnitřní úhly při vrcholech A a B (obr. 21). Bodem E prochází společná tečna obou uvažovaných kružnic, úhel DEC



Obr. 21

je tedy součtem úsekových úhlů příslušných tětivě DE v jedné kružnici (s obvodovým úhlem α) a tětivě EC v druhé kružnici (s obvodovým úhlem β). Jeho velikost je tudíž $\alpha + \beta$. A protože velikost úhlu CVD je $180° - (\alpha + \beta)$, zjišťujeme, že ve čtyřúhelníku $CVDE$ se úhly u protějších vrcholů E a V doplňují do 180°. To, jak víme, znamená, že $CVDE$ je tětivový čtyřúhelník, tj. bod E leží na kružnici opsané trojúhelníku CDV.

B – II – 1

Jak víme, každé přirozené číslo k dává při dělení třemi stejný zbytek jako číslo $S(k)$ rovné součtu číslic původního čísla k. Číslo a_n proto dává při dělení třemi stejný zbytek jako součet $S(1^2) + S(2^2) + \ldots + S(n^2)$, tedy rovněž jako součet $1^2 + 2^2 + \ldots + n^2$. *Dvěma způsoby* ukážeme, že poslední součet je dělitelný třemi, právě když číslo n je tvaru $9k - 5$, $9k - 1$ nebo $9k$, kde k je přirozené číslo.

Při *prvním způsobu* využijeme známý vzorec

$$1^2 + 2^2 + \ldots + n^2 = \frac{n(n+1)(2n+1)}{6}, \tag{1}$$

z něhož plyne, že zkoumaný součet je dělitelný třemi, právě když je součin $n(n+1)(2n+1)$ dělitelný devíti. Protože čísla n, $n+1$ a $2n+1$ jsou navzájem nesoudělná, hledáme právě ta n, pro která je dělitelné devíti jedno z čísel n, $n+1$ nebo $2n+1$, a to jsou po řadě čísla tvaru $9k$, $9k-1$, $9k-5$.

Druhý způsob je založen na pozorování, že zbytky čísel 1^2, 2^2, 3^2, 4^2, 5^2, ... při dělení třemi jsou $1, 1, 0, 1, 1, 0, \ldots$, tedy opakují se s periodou 3. Skutečně, čísla $(k+3)^2$ a k^2 dávají stejný zbytek při dělení třemi, neboť jejich rozdíl je číslo $3(2k+3)$, což je násobek tří. Sčítáním uvedených zbytků dostaneme postupně zbytky prvních devíti součtů (1): 1, 2, 2, 0, 1, 1, 2, 0, 0; poté se zbytky dalších součtů (1) začnou periodicky opakovat. (Plyne to z toho, že předchozí součet devíti čísel dává nulový zbytek a zároveň je počet sčítanců násobkem periody 3 sčítaných zbytků.)

Víme již, která čísla a_n jsou dělitelná třemi; posoudíme nyní snazší otázku, která a_n jsou dělitelná čtyřmi. Ukažme, že to jsou všechna a_n se sudým $n > 2$ (a žádná jiná). Číslo a_n s lichým n je totiž liché, číslo a_2 se rovná 14 a číslo a_n se sudým $n > 2$ končí stejným dvojčíslím jako číslo n^2, takže je takové a_n (stejně jako zmíněné dvojčíslí) dělitelné čtyřmi.

Spojíme-li výsledky o dělitelnosti třemi a čtyřmi dohromady, dojdeme k zjištění, že číslo a_n je dělitelné dvanácti, právě když je číslo n jednoho

z tvarů $18k-14$, $18k-10$ nebo $18k$, kde k je libovolné přirozené číslo. Protože $100\,000 = 5\,556 \times 18 - 8$, je mezi přirozenými čísly od 1 do 100 000 právě 5 556 čísel $18k-14$, 5 556 čísel $18k-10$ a 5 555 čísel $18k$, dohromady je to 16 667 čísel.

B – II – 2

Pro koeficient a musí platit $a \neq 0$ a $a \neq -1$, aby všechny uvažované trojčleny byly skutečně kvadratické trojčleny. Jak víme, kvadratický trojčlen má dvojnásobný kořen, právě když je jeho diskriminant nulový. Sestavme proto diskriminanty všech tří trojčlenů se zvětšenými koeficienty:

$$\begin{aligned}
&(a+1)x^2+bc+c \quad \text{má diskriminant} \quad D_1 = b^2 - 4(a+1)c,\\
&ax^2+(b+1)x+c \quad \text{má diskriminant} \quad D_2 = (b+1)^2 - 4ac,\\
&ax^2+bx+(c+1) \quad \text{má diskriminant} \quad D_3 = b^2 - 4a(c+1).
\end{aligned}$$

Hledáme tedy reálná čísla a, b, c, pro která platí $a \neq 0$, $a \neq -1$ a $D_1 = D_2 = D_3 = 0$.

Z rovnosti $D_1 = D_3$ plyne $c = a$, takže $D_2 = (b+1)^2 - 4a^2 = (b+1-2a)(b+1+2a)$; rovnost $D_2 = 0$ pak znamená, že platí $b = \pm 2a - 1$, a proto $D_1 = (\pm 2a-1)^2 - 4(a+1)a = 4a^2 \mp 4a + 1 - 4a^2 - 4a = 1 \mp 4a - 4a$, tudíž $D_1 = -8a+1$ nebo $D_1 = 1$. Proto z rovnosti $D_1 = 0$ plyne $a = 1/8$, $b = 2a - 1 = -3/4$ a $c = a = 1/8$ (zkouška je snadná, není však nutná, naším postupem totiž máme zaručeny rovnosti $D_1 = D_3$, $D_2 = 0$ a $D_1 = 0$).

Odpověď: Úloze vyhovuje jediný trojčlen $\frac{1}{8}x^2 - \frac{3}{4}x + \frac{1}{8}$.

B – II – 3

Kladné číslo x je řešením rovnice s daným n, právě když je číslo nx přirozené a jsou splněny nerovnosti

$$nx - 1 \leqq x\sqrt{n^2-1} < nx.$$

Pravá nerovnost je splněna pro každé $x > 0$, neboť zřejmě platí $\sqrt{n^2-1} < \sqrt{n^2} = n$. Zbývá tedy vyřešit levou nerovnici (vzhledem k neznámé x). Po jednoduché úpravě dostáváme

$$\begin{aligned}
x\left(n - \sqrt{n^2-1}\right) &\leqq 1,\\
x \leqq \frac{1}{n-\sqrt{n^2-1}} &= n + \sqrt{n^2-1}.
\end{aligned}$$

Využili jsme toho, že výraz $n - \sqrt{n^2-1}$ je kladný a v součinu se sdruženým výrazem $n + \sqrt{n^2-1}$ dává číslo 1. Po vynásobení obou stran odvozené nerovnosti číslem n dostaneme pro přirozené číslo $k = nx$ ekvivalentní podmínku

$$k \leqq n^2 + n\sqrt{n^2-1},$$

která je splněna právě pro $k \in \{1, 2, \ldots, 2n^2-1\}$, neboť pro druhý sčítanec z pravé strany poslední nerovnosti zřejmě platí celočíselné odhady

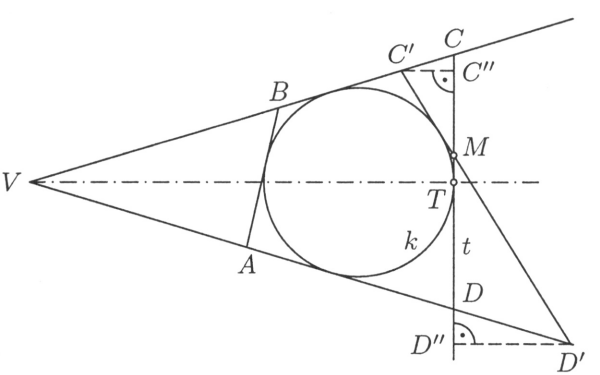
$$n^2 - 1 \leqq n\sqrt{n^2-1} < n^2$$

(znovu využíváme pouze nerovnost $\sqrt{n^2-1} < n$). Všechna řešení dané rovnice jsou tvaru $x = k/n$ a tvoří tak množinu zlomků

$$\left\{\frac{1}{n}, \frac{2}{n}, \ldots, \frac{2n^2-1}{n}\right\}.$$

B – II – 4

Kružnice vepsaná hledanému čtyřúhelníku je kružnicí k připsanou straně AB trojúhelníku BAV. Ten ze dvou průsečíků osy úhlu AVB s kružnicí k, který je dál od vrcholu V, označme T (obr. 22). Hledané body C a D nalezneme jako průsečíky tečny t v bodě T ke kružnici k po řadě s přímkami VB, VA. Dokažme, že takto sestrojený čtyřúhelník má ze všech čtyřúhelníků vyhovujících podmínkám úlohy nejmenší obsah.



Obr. 22

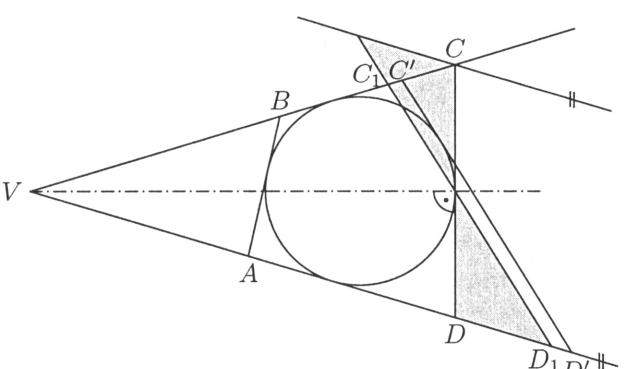
Označme C', D' vrcholy jiného tečnového čtyřúhelníku s vepsanou kružnicí k (přímka $C'D'$ je tečnou kružnice k). Bez újmy na obecnosti můžeme předpokládat, že průsečík M tečen t a $C'D'$ leží uvnitř úsečky TC.

To znamená, že platí $|MD| > |MC|$ (obr. 22). Označme C'' a D'' odpovídající paty kolmic spuštěných z bodů C' a D' na přímku t; bod C'' leží uvnitř úsečky MC a D'' na polopřímce MD vně úsečky MD, takže $|MC''| < |MC| < |MD| < |MD''|$ a z podobnosti pravoúhlých trojúhelníků $MC'C''$ a $MD'D''$ plyne $|C'C''| < |D'D''|$. Trojúhelník DMD' má tudíž větší obsah než trojúhelník CMC'. Rozdíl jejich obsahů je však roven rozdílu obsahů čtyřúhelníků $ABC'D'$ a $ABCD$, tedy obsah čtyřúhelníku $ABC'D'$ je větší než obsah čtyřúhelníku $ABCD$.

Jiné řešení. Stejně jako v prvním řešení označme C, D průsečíky tečny t připsané kružnice k s rameny úhlu VB, VA. Jsou-li C', D' vrcholy jiného tečnového čtyřúhelníku s vepsanou kružnicí k, platí pro obsahy tečnových čtyřúhelníků $ABCD$ a $ABC'D'$

$$S(ABCD) = S(VCD) - S(VAB),$$
$$S(ABC'D') = S(VC'D') - S(VAB).$$

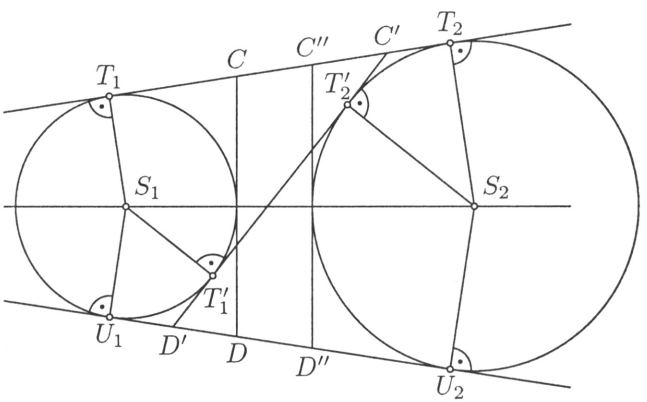
Stačí tedy ukázat, že pro libovolnou takovou tečnu $C'D'$, která není kolmá na osu úhlu AVB, platí $S(VC'D') > S(VCD)$. To je však zřejmé z obr. 23 (oba šedé trojúhelníky mají díky středové souměrnosti stejný obsah a přitom $S(VC'D') > S(VC_1D_1) > S(VCD)$).



Obr. 23

Jiné řešení. Obsah tečnového čtyřúhelníku $ABCD$, jehož vepsaná kružnice má poloměr r, je $S = \frac{1}{2}r(|AB| + |BC| + |CD| + |DA|) = \frac{1}{2}r(2|AB| + 2|CD|) = r(|AB| + |CD|)$. Obsah tečnového čtyřúhelníku $ABCD$ splňujícího podmínky úlohy bude tedy nejmenší, právě když bude nejkratší úsečka CD.

Uvažujme kružnici připsanou straně $C'D'$ trojúhelníku $VC'D'$ (obr. 24). Z vlastností tečen postupně nahlédneme, že je $|T_1C| = \frac{1}{2}|CD|$,



Obr. 24

$|T_2C''| = \frac{1}{2}|C''D''|$ a také $|C'D'| = |T_1T_2|$. Poslední rovnost plyne ze známých vlastností vepsané a připsané kružnice, totiž že jejich body dotyku na společnou stranu jsou souměrně sdružené podle středu strany; důkaz ovšem vyžaduje trochu počítání:

$$|T_1T_2| = |T_1C'| + |C'T_2| = |T_1'C'| + |C'T_2'| = |T_1'T_2'| + 2|T_2'C'|,$$
$$|U_1U_2| = |U_1D'| + |D'U_2| = |T_1'D'| + |D'T_2'| = |T_1'T_2'| + 2|T_1'D'|.$$

Ze souměrnosti podle osy úhlu AVB plyne $|T_1T_2| = |U_1U_2|$, takže $|T_2'C'| = |T_1'D'|$. Je tedy $|C'D'| = |T_1'T_2'| + 2|T_2'C'| = |T_1T_2|$.

Protože obě kružnice jsou odděleny společnou tečnou $C'D'$, nemohou se dotýkat, takže $|CD| < |C''D''|$, neboli $|CT_1| < |C''T_2|$. To znamená, že je

$$|CD| = 2|T_1C| < |T_1C| + |C''T_2| < |T_1T_2| = |C'D'|,$$

což jsme chtěli dokázat.

Kategorie A

Texty úloh

A – I – 1

Určete všechny dvojice (p, q) reálných čísel takové, že rovnice $x^2 + px + q = 0$ má řešení v oboru reálných čísel, přičemž platí: Je-li t kořenem této rovnice, je $|2t - 15|$ rovněž jejím kořenem. *(P. Černek)*

A – I – 2

V rovině daného čtverce $KLMN$ určete množinu všech bodů P, pro něž jsou úhly NPK, KPL a LPM shodné. *(J. Švrček)*

A – I – 3

Pro libovolné přirozené číslo n sestavme z písmen A, B všechna možná „slova" délky n. Rozdělme je do dvou skupin S_n a L_n podle toho, zda je v daném slově sudý, resp. lichý počet „slabik" BA (za sudý považujeme i počet 0). Například slova $\underline{BA}BBB\underline{BA}$ a $AAAAAAB$ patří obě do skupiny S_7, slova $AAB\underline{BA}BB$ a $\underline{BA}\,\underline{BA}A\underline{BA}$ patří obě do skupiny L_7. Určete, pro která n mají skupiny S_n a L_n stejný počet prvků.

(J. Šimša)

A – I – 4

Určete nejmenší reálné číslo p takové, že nerovnost

$$\sqrt{1^2+1} + \sqrt{2^2+1} + \sqrt{3^2+1} + \ldots + \sqrt{n^2+1} \leqq \frac{1}{2}n(n+p)$$

platí pro každé přirozené číslo n. *(S. Trávníček)*

A – I – 5

Nechť $ABCD$ je tětivový čtyřúhelník, jehož vnitřní úhel při vrcholu B má velikost $60°$.

a) Jestliže $|BC| = |CD|$, pak platí $|CD| + |DA| = |AB|$. Dokažte.
b) Rozhodněte, zda platí opačná implikace. (*E. Kováč*)

A – I – 6

V oboru reálných čísel řešte soustavu rovnic

$$x^2 = \frac{1}{y} + \frac{1}{z}, \quad y^2 = \frac{1}{z} + \frac{1}{x}, \quad z^2 = \frac{1}{x} + \frac{1}{y}.$$

(*J. Šimša*)

A – S – 1

Nechť $P(x) = ax^2 + bx + c$ je kvadratický trojčlen s nezápornými reálnými koeficienty. Dokažte, že pro libovolné kladné číslo x platí

$$P(x) \cdot P\Big(\frac{1}{x}\Big) \geqq \big(P(1)\big)^2.$$

(*E. Kováč*)

A – S – 2

Určete, jakou největší délku může mít úhlopříčka CE konvexního pětiúhelníku $ABCDE$, jehož strana AB má délku 6 cm, vnitřní úhly při vrcholech C a E jsou pravé a úhel ADB má velikost 120°. (*P. Černek*)

A – S – 3

V oboru reálných čísel řešte soustavu rovnic

$$\begin{aligned} x^2 + 2yz &= 6(y + z - 2), \\ y^2 + 2zx &= 6(z + x - 2), \\ z^2 + 2xy &= 6(x + y - 2). \end{aligned}$$

(*J. Šimša*)

A – II – 1

Určete počet všech pětimístných palindromů, které jsou dělitelné číslem 37. (Palindromem nazýváme číslo, jehož zápis v desítkové soustavě se čte zepředu stejně jako zezadu.) (*J. Šimša*)

A – II – 2

Pro libovolné přirozené číslo n sestavme z písmen A a B všechna možná „slova“ délky n a označme p_n počet těch z nich, která neobsahují ani trojici AAA po sobě jdoucích písmen A, ani dvojici BB po sobě jdoucích písmen B. Určete, pro která přirozená čísla n platí, že obě čísla p_n a p_{n+1} jsou sudá. *(R. Kučera)*

A – II – 3

Nechť K je libovolný vnitřní bod strany AB daného trojúhelníku ABC. Přímka CK protíná kružnici opsanou trojúhelníku ABC v bodě L ($L \neq C$). Označme k_1 kružnici opsanou trojúhelníku AKL a k_2 kružnici opsanou trojúhelníku BKL.

a) Dokažte, že přímka AC je tečna kružnice k_1, právě když přímka BC je tečna kružnice k_2.

b) Předpokládejme, že přímka AC je sečna kružnice k_1. Nechť P ($P \neq A$) je průsečík přímky AC s kružnicí k_1 a Q ($Q \neq B$) průsečík přímky BC s kružnicí k_2. Dokažte, že bod K leží na úsečce PQ.

(J. Šimša, J. Zhouf)

A – II – 4

Nechť K, L, M jsou po řadě průsečíky os vnitřních úhlů α, β, γ při vrcholech A, B, C daného trojúhelníku ABC s protějšími stranami BC, CA, AB. Dokažte, že platí nerovnost

$$\frac{|BC|}{|AK|}\cos\frac{\alpha}{2} + \frac{|CA|}{|BL|}\cos\frac{\beta}{2} + \frac{|AB|}{|CM|}\cos\frac{\gamma}{2} \geqq 3.$$

(J. Švrček)

A – III – 1

Určete všechny trojice (x, y, z) reálných čísel, pro něž platí

$$x^2 + y^2 + z^2 \leqq 6 + \min\left\{x^2 - \frac{8}{x^4},\ y^2 - \frac{8}{y^4},\ z^2 - \frac{8}{z^4}\right\}.$$

(J. Švrček)

A – III – 2

Pro libovolné přirozené číslo n sestavme z písmen A a B všechna možná „slova“ délky n a označme p_n počet těch z nich, která neobsahují ani čtveřici $AAAA$ po sobě jdoucích písmen A, ani trojici BBB po sobě jdoucích písmen B. Určete hodnotu výrazu

$$\frac{p_{2004} - p_{2002} - p_{1999}}{p_{2001} + p_{2000}}.$$

(R. Kučera)

A – III – 3

V rovině je dána kružnice k a 121 jejích sečen $p_1, p_2, \ldots, p_{121}$. Uvnitř této kružnice je na každé přímce p_i dán bod A_i. Dokažte, že na kružnici k existuje bod X takový, že úsečka A_iX svírá s přímkou p_i úhel menší než 21° pro nejméně 29 různých indexů i. *(J. Šimša)*

A – III – 4

Zjistěte, pro která přirozená čísla n je součet

$$\frac{n}{1!} + \frac{n}{2!} + \ldots + \frac{n}{n!}$$

číslo celé. *(E. Kováč)*

A – III – 5

Nechť L je libovolný vnitřní bod kratšího oblouku CD kružnice opsané čtverci $ABCD$. Označme K průsečík přímek AL a CD, M průsečík přímek AD a CL a N průsečík přímek MK a BC. Dokažte, že body B, L, M, N leží na téže kružnici. *(J. Švrček)*

A – III – 6

Nechť $\mathbb{R}_+$ značí množinu všech kladných reálných čísel. Určete všechny funkce $f\colon \mathbb{R}_+ \to \mathbb{R}_+$, které pro libovolná kladná čísla x, y splňují rovnost

$$x^2\big(f(x) + f(y)\big) = (x + y)f\big(f(x)y\big).$$

(P. Kaňovský)

A – I – 1

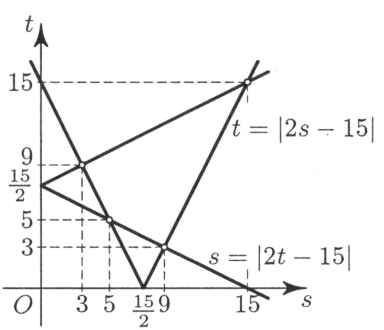
Nechť t, s jsou reálné kořeny dané kvadratické rovnice. Uvažujme nejprve případ, kdy uvažovaná kvadratická rovnice má dvojnásobný (reálný) kořen. Platí tedy $t = s$ a přitom podle podmínek úlohy $t = |2t - 15|$. Pro $t \geqq \frac{15}{2}$ dostáváme rovnici $t = 2t - 15$ s řešením $t = 15$ a pro $t < \frac{15}{2}$ rovnici $t = -(2t - 15)$ s řešením $t = 5$. Jim odpovídající kvadratické rovnice mají tvar $(x - 5)^2 = x^2 - 10x + 25 = 0$ a $(x - 15)^2 = x^2 - 30x + 225 = 0$.

Uvažujme nyní případ, kdy uvažovaná kvadratická rovnice má dva různé reálné kořeny t, s. Rozlišíme tři případy.

- ▷ $t = |2t - 15|$ a současně $s = |2s - 15|$. Řešení obou rovnic (viz výše) tvoří dvojici $\{t, s\} = \{5, 15\}$. Odpovídající kvadratická rovnice má tvar $(x - 5)(x - 15) = x^2 - 20x + 75 = 0$.
- ▷ $t = |2s - 15|$ a současně $s = |2t - 15|$. Řešením čtyř soustav rovnic

$$t = \pm(2s - 15), \quad s = \pm(2t - 15)$$

(jež odpovídají různým volbám znamének) dostaneme dvojice (s, t) rovné $(15, 15)$, $(5, 5)$, $(3, 9)$ a $(9, 3)$, z nichž pouze poslední dvě vyhovují původní soustavě a podmínce $s \neq t$. Dodejme, že soustavu rovnic $t = |2s - 15|$ a $s = |2t - 15|$ lze rovněž řešit graficky v rovině Ost, do které zakreslíme obě lomené čáry $t = |2s - 15|$ a $s = |2t - 15|$ (obr. 25). Dvojicím $(3, 9)$ a $(9, 3)$ odpovídá kvadratická rovnice $(x - 3)(x - 9) = x^2 - 12x + 27 = 0$.



Obr. 25

- ▷ $t = |2t - 15| = |2s - 15|$. Jak už víme, rovnice $t = |2t - 15|$ má řešení $t = 5$ a $t = 15$. Pro $t = 5$ z rovnice $5 = |2s - 15|$ plyne $s = 5$

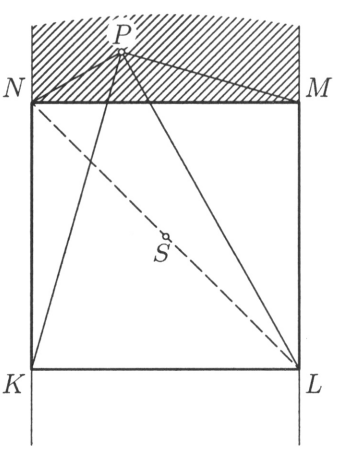
nebo $s = 10$, pro $t = 15$ z rovnice $15 = |2s - 15|$ plyne $s = 0$ nebo $s = 15$. S ohledem na podmínku $s \neq t$ tak dostáváme dvě řešení $(t, s) = (5, 10)$ a $(t, s) = (15, 0)$. Těmto řešením pak odpovídají po řadě dvě kvadratické rovnice $(x - 5)(x - 10) = x^2 - 15x + 50 = 0$ a $(x - 15)x = x^2 - 15x = 0$.

Závěr: Dané úloze vyhovuje šest dvojic (p, q) reálných čísel, a to dvojice $(-10, 25)$, $(-30, 225)$, $(-20, 75)$, $(-12, 27)$, $(-15, 50)$ a $(-15, 0)$.

A – I – 2

Označme P hledanou množinu bodů a S střed čtverce $KLMN$. Zřejmě $S \in \mathsf{P}$ (obr. 26).

Dále určíme všechny hledané body P ($P \neq S$), které leží uvnitř pásu omezeného rovnoběžkami KN a LM. Ukážeme, že každý takový bod P leží v polorovině opačné k polorovině MNK. Pro každý bod P uvažovaného pásu, který leží v polorovině opačné k polorovině KLM, platí totiž $|\sphericalangle KPL| > |\sphericalangle KPN|$, neboť polopřímka PN leží v úhlu KPL. Podobně zjistíme, že žádný bod čtverce $KLMN$ kromě jeho středu S nemá danou vlastnost.



Obr. 26

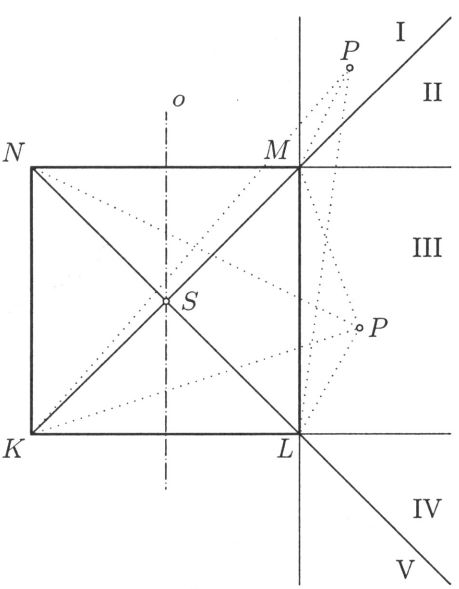
Leží-li tedy hledaný bod P ve vyšrafované oblasti na obr. 26, jsou přímky PK a PL podle zadání osami úhlů NPL a KPM. Proto v trojúhelníku LPN osa PK úhlu NPL protíná kružnici opsanou tomuto trojúhelníku (kromě bodu P) v bodě ležícím na ose strany NL. Tímto bodem

je ovšem vrchol K čtverce $KLMN$. Body P, N, K, L tedy leží na téže kružnici, kterou je kružnice opsaná čtverci $KLMN$. (Analogický výsledek obdržíme, uvažujeme-li osu PL úhlu KPM.) Bod P proto leží na kratším oblouku $l = \overset{\frown}{MN}$ kružnice opsané čtverci $KLMN$. Naopak pro každý bod $P \in l$ platí podle věty o obvodových úhlech (pro shodné tětivy NK, KL, LM)

$$|\sphericalangle NPK| = |\sphericalangle KPL| = |\sphericalangle LPM| = 45°.$$

Tím je hledání bodů P v pásu mezi rovnoběžkami KM a LM ukončeno.

Dále snadno nahlédneme, že libovolný vnitřní bod P každé z polopřímek opačných k polopřímkám KM, LN, MK, NL má danou vlastnost. Ukážeme, že žádný další bod roviny čtverce $KLMN$ uvedenou vlastnost nemá. Stačí se přitom díky symetrii omezit na jednu z polorovin vyťatých osou o strany KL daného čtverce. Protože jsme již vyšetřili celý pás omezený rovnoběžkami KN a LM, lze (bez újmy na obecnosti) zkoumat jen body poloroviny opačné k polorovině LMN. Přímky KL, MN, LM, KM a LN dělí tuto polorovinu na pět částí (obr. 27), přitom žádný bod přímek KL, LM a MN danou vlastnost očividně nemá.

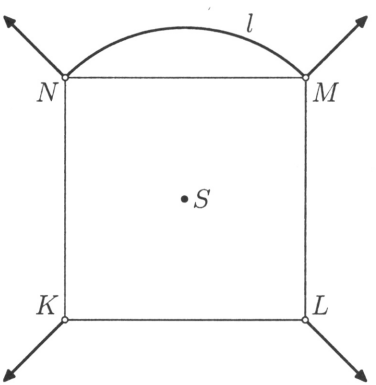


Obr. 27

Ukážeme, že žádný vnitřní bod každé z oblastí I–V̄ roviny čtverce $KLMN$ není prvkem množiny P. Jestliže P je vnitřním bodem oblasti I, evidentně platí $|\sphericalangle KPL| > |\sphericalangle LPM|$ (obr. 27). Je-li P vnitřním bodem libovolné z oblastí II nebo III, platí naopak $|\sphericalangle KPL| < |\sphericalangle LPN|$. Pro libovolný vnitřní bod oblasti IV zase platí $|\sphericalangle NPK| > |\sphericalangle KPL|$ a pro libovolný vnitřní bod P oblasti V platí naopak $|\sphericalangle NPK| < |\sphericalangle KPL|$. Ve všech pěti uvažovaných případech jsme se však vždy dostali do rozporu s podmínkami úlohy.

Tím jsme prozkoumali všechny body roviny čtverce $KLMN$.

Závěr: Hledaná množina bodů P se skládá ze všech vnitřních bodů kratšího oblouku MN kružnice opsané danému čtverci $KLMN$, ze všech vnitřních bodů polopřímek opačných k polopřímkám KM, LN, MK a NL a ze středu S daného čtverce (obr. 28).



Obr. 28

A – I – 3

Skupinu S_n rozdělme na dvě části $(SA)_n$ a $(SB)_n$ podle toho, zda slovo skupiny S_n končí písmenem A, resp. B. Skupinu L_n rozdělme analogicky na dvě části $(LA)_n$ a $(LB)_n$ podle toho, zda slovo skupiny L_n končí písmenem A, resp. B. Označme dále s_n, l_n, $(sA)_n$, $(sB)_n$, $(lA)_n$, $(lB)_n$ po řadě počty prvků skupin S_n, L_n, $(SA)_n$, $(SB)_n$, $(LA)_n$, $(LB)_n$. Pro každé přirozené číslo n pak podle našeho rozdělení platí

$$\begin{aligned} s_n &= (sA)_n + (sB)_n, \\ l_n &= (lA)_n + (lB)_n. \end{aligned} \tag{1}$$

Každé slovo ze skupiny $(SA)_{n+1}$ vznikne tak, že připíšeme písmeno A buď na konec slova ze skupiny $(SA)_n$, nebo na konec slova ze skupiny $(LB)_n$. Platí proto

$$(sA)_{n+1} = (sA)_n + (lB)_n.$$

Analogicky platí rovněž vztahy

$$\begin{aligned} (sB)_{n+1} &= (sA)_n + (sB)_n, \\ (lA)_{n+1} &= (sB)_n + (lA)_n, \\ (lB)_{n+1} &= (lA)_n + (lB)_n. \end{aligned} \tag{2}$$

Pro $n = 1$ mají skupiny následující tvar

$$(SA)_1 = \{A\},\ (SB)_1 = \{B\},\ (LA)_1 = \emptyset,\ (LB)_1 = \emptyset,$$

a tedy $(sA)_1 = (sB)_1 = 1$ a $(lA)_1 = (lB)_1 = 0$.

Předpokládejme, že pro určité přirozené číslo k obsahují skupiny $(SA)_k$ a $(SB)_k$ stejný počet prvků, který označíme p, a zároveň skupiny $(LA)_k$ a $(LB)_k$ mají stejný počet prvků, který označíme q. Navíc předpokládejme, že platí $p \neq q$, jak je tomu v případě $k = 1$, kdy $p = 1$ a $q = 0$. Do následující tabulky zapišme počty prvků ve skupinách pro čísla $n = k, k+1, k+2, k+3, k+4$. Přitom pro výpočty hodnot užijeme vztahy (1) a (2).

n	k	$k+1$	$k+2$	$k+3$	$k+4$
$(sA)_n$	p	$p+q$	$p+3q$	$2p+6q$	$6p+10q$
$(sB)_n$	p	$2p$	$3p+q$	$4p+4q$	$6p+10q$
$(lA)_n$	q	$p+q$	$3p+q$	$6p+2q$	$10p+6q$
$(lB)_n$	q	$2q$	$p+3q$	$4p+4q$	$10p+6q$
s_n	$2p$	$3p+q$	$4p+4q$	$6p+10q$	$12p+20q$
l_n	$2q$	$p+3q$	$4p+4q$	$10p+6q$	$20p+12q$

Z tabulky lze vyčíst několik poznatků. Protože $p \neq q$, platí rovněž $2p \neq$ $\neq 2q$, $3p+q \neq p+3q$ a $6p+10q \neq 10p+6q$. Vidíme, že $s_k \neq l_k$, $s_{k+1} \neq l_{k+1}$, $s_{k+2} = l_{k+2}$, $s_{k+3} \neq l_{k+3}$ a že skupiny $(SA)_{k+4}$ a $(SB)_{k+4}$ obsahují opět stejný počet prvků a skupiny $(LA)_{k+4}$ a $(LB)_{k+4}$ opět stejný počet prvků, přitom tyto počty jsou navzájem různé.

Užitím matematické indukce usoudíme, že uvedená tabulka má všechny zmíněné vlastnosti pro každé $k = 4m+1$, kde m je celé nezáporné číslo, takže rovnost $s_n = l_n$ platí, právě když $n = k+2 = 4m+3$.

Závěr: Skupiny S_n a L_n mají stejný počet prvků, právě když $n = 4m + 3$, kde m je celé nezáporné číslo.

Jiné řešení. Ze vztahů (1) a (2) plynou pro každé $n \geqq 2$ rovnosti

$$\begin{aligned} s_{n+1} &= (sA)_{n+1} + (sB)_{n+1} = (sA)_n + (lB)_n + s_n = \\ &= 2s_n + (lB)_n - (sB)_n = 2s_n + l_{n-1} - s_{n-1} = \\ &= 2s_n - 2s_{n-1} + \underbrace{l_{n-1} + s_{n-1}}. \end{aligned}$$

Svorkou označený součet je roven počtu všech slov délky $n - 1$, tedy číslu 2^{n-1}. Znamená to, že posloupnost zkoumaných čísel $\{s_n\}$ splňuje rekurentní rovnici

$$s_{n+1} = 2s_n - 2s_{n-1} + 2^{n-1} \qquad (n = 2, 3, \ldots), \tag{3}$$

jež (spolu s počátečními hodnotami $s_1 = 0$, $s_2 = 1$) umožňuje postupný výpočet všech hodnot s_n. Podle teorie rekurentních rovnic se dá najít řešení takové úlohy v explicitním tvaru

$$s_n = 2^{n-1} - \left(\sqrt{2}\right)^{n-1} \cos \frac{\pi(n-1)}{4},$$

z něhož plyne, že zkoumaná rovnost $s_n = 2^{n-1}$ nastane právě pro ta čísla n, jež jsou tvaru $4m + 3$. Bez znalosti této teorie se obejdeme takto: vypočteme pomocí (3) několik prvních hodnot s_n a zapíšeme je do tabulky, kam pro porovnání uvedeme i příslušné hodnoty 2^{n-1} a rozdíl $s_n - 2^{n-1}$:

n	1	2	3	4	5	6	7	8	9	10	11	...
s_n	0	1	4	10	20	36	64	120	240	496	1 024	...
2^{n-1}	1	2	4	8	16	32	64	128	256	512	1 024	...
$s_n - 2^{n-1}$	−1	−1	0	2	4	4	0	−8	−16	−16	0	...

Tak přijdeme k hypotéze, že hledaná n jsou tvaru $4m + 3$, a objevíme rovněž vlastnosti diferencí $s_n - 2^{n-1}$ (jsou to až na znaménka mocniny dvou). Pokusíme se proto najít závislost mezi čísly s_{n+4} a s_n. Podle (3) postupně určíme

$$\begin{aligned} s_{n+2} &= 2s_{n+1} - 2s_n + 2^n, \\ s_{n+3} &= 2s_{n+2} - 2s_{n+1} + 2^{n+1} = 2(2s_{n+1} - 2s_n + 2^n) - 2s_{n+1} + 2^{n+1} = \\ &= 2s_{n+1} - 4s_n + 2^{n+2}, \\ s_{n+4} &= 2s_{n+3} - 2s_{n+2} + 2^{n+2} = \\ &= 2(2s_{n+1} - 4s_n + 2^{n+2}) - 2(2s_{n+1} - 2s_n + 2^n) + 2^{n+2} = \\ &= -4s_n + 2^{n+3} - 2^{n+1} + 2^{n+2} = -4s_n + 2^{n+3} + 2^{n+1}, \end{aligned}$$

neboli

$$s_{n+4} - 2^{n+3} = -4(s_n - 2^{n-1}).$$

Z posledního vztahu okamžitě plyne, že rovnost $s_{n+4} = 2^{n+3}$ platí, právě když $s_n = 2^{n-1}$. Protože z hodnot $n = 1, 2, 3, 4$ poslední rovnost platí pouze pro $n = 3$, je hypotéza o tom, že hledaná n jsou právě čísla tvaru $n = 4m + 3$, dokázána.

A – I – 4

Pro $n = 1$ má daná nerovnost tvar

$$\sqrt{2} \leqq \frac{1}{2}(p+1), \quad \text{neboli} \quad p \geqq 2\sqrt{2} - 1.$$

Označme $p_1 = 2\sqrt{2} - 1$. Zjistili jsme, že žádné číslo p menší než p_1 požadovanou vlastnost nemá. Číslo p_1 je tedy hledané číslo, pokud ukážeme, že pro každé $n \geqq 1$ platí

$$\sqrt{1^2+1} + \sqrt{2^2+1} + \sqrt{3^2+1} + \ldots + \sqrt{n^2+1} \leqq \frac{1}{2}n(n+p_1). \qquad (1)$$

Důkaz provedeme matematickou indukcí.

(i) Pro $n = 1$ je nerovnost (1) splněna díky způsobu, jakým jsme číslo p_1 určili.

(ii) Předpokládejme, že nerovnost (1) platí pro určité přirozené číslo n, a ukážeme, že platí i pro přirozené číslo $n + 1$. Nechť tedy

$$F(n) = \sqrt{1^2+1} + \sqrt{2^2+1} + \sqrt{3^2+1} + \ldots + \sqrt{n^2+1} \leqq \leqq \frac{1}{2}n(n+p_1). \qquad (2)$$

Protože

$$F(n+1) = F(n) + \sqrt{(n+1)^2+1},$$

platí podle indukčního předpokladu (2) a definice p_1

$$F(n+1) \leqq \frac{1}{2}n\big(n+2\sqrt{2}-1\big) + \sqrt{(n+1)^2+1}. \qquad (3)$$

Nyní dokážeme nerovnost

$$\frac{1}{2}n\big(n+2\sqrt{2}-1\big) + \sqrt{(n+1)^2+1} \leqq \frac{1}{2}(n+1)\big(n+1+2\sqrt{2}-1\big). \qquad (4)$$

Její úpravou dostaneme nerovnost s ní ekvivalentní

$$\sqrt{(n+1)^2+1} \leqq n+\sqrt{2},$$

o jejíž platnosti se snadno přesvědčíme po umocnění obou stran na druhou:

$$\left(n+\sqrt{2}\right)^2 = n^2+2\sqrt{2}n+2 > n^2+2n+2 = (n+1)^2+1.$$

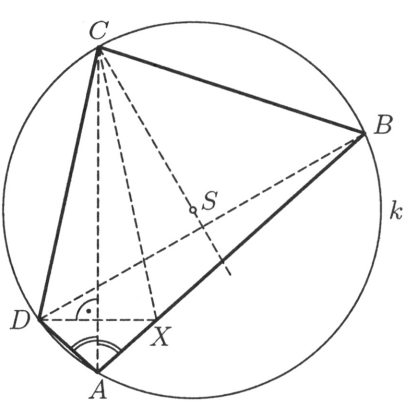
Podle (3) a (4) platí

$$F(n+1) \leqq \frac{1}{2}(n+1)\left(n+1+2\sqrt{2}-1\right) = \frac{1}{2}(n+1)(n+1+p_1),$$

což je nerovnost (1) pro hodnotu $n+1$.
Závěr: Hledaným reálným číslem je číslo $p = 2\sqrt{2}-1$.

A – I – 5

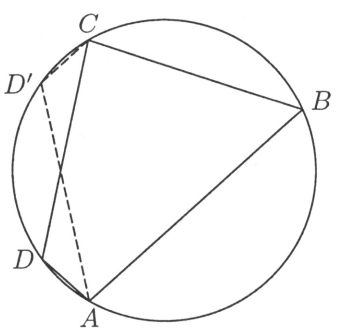
Nejprve zvažme, jak může takový tětivový čtyřúhelník $ABCD$ s šedesátistupňovým úhlem při vrcholu B a se shodnými stranami BC a CD vypadat. Označme k kružnici, jež je čtyřúhelníku $ABCD$ opsána. Protože $|\sphericalangle ABC| = 60°$, je už určena velikost úhlopříčky AC, která je tětivou odpovídající obvodovému úhlu 60°. Vrchol D pak musí být vnitřním bodem kratšího oblouku AC kružnice k (v polorovině opačné k ACB) a vrchol B je obrazem bodu D v souměrnosti podle přímky SC (obr. 29), kde S je střed kružnice k.



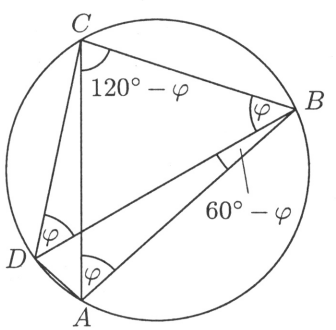
Obr. 29

Protože dle předpokladu $|BC| = |CD|$, jsou obvodové úhly BAC a CAD příslušné shodným tětivám shodné. Vidíme tedy, že polopřímky AD a AB jsou souměrně sdruženy dle osy AC. Označme X obraz bodu D v této souměrnosti (obr. 29). Bod X zřejmě leží uvnitř strany AB (obraz kratšího oblouku AC leží celý ve vnitřní oblasti kružnice k), a protože $|CX| = |CD| = |BC|$, je trojúhelník XBC rovnoramenný. Trojúhelník XBC je dokonce rovnostranný, protože velikost jeho úhlu při vrcholu B je $60°$. Je proto $|BX| = |BC| = |CD|$. Ze souměrnosti navíc plyne $|DA| = |XA|$, takže $|CD| + |DA| = |BX| + |XA| = |AB|$, což je požadovaná rovnost.

b) Snadno nahlédneme, že opačná implikace neplatí. Stačí vzít takový čtyřúhelník $ABCD$, který splňuje předpoklady úlohy, a zároveň v něm platí $|CD| \neq |DA|$ (takový určitě existuje, jak jsme naznačili hned v úvodu řešení). Prohodíme-li nyní strany CD a DA, tj. nahradíme-li vrchol D vrcholem D' souměrně sdruženým s vrcholem D podle osy úhlopříčky AC (obr. 30), dostaneme tětivový čtyřúhelník $ABCD'$ s šedesátistupňovým úhlem při vrcholu B, který bude i nadále splňovat rovnost $|CD'| + |D'A| = |DA| + |CD| = |AB|$, ale bude v něm platit $|BC| = |CD| = |D'A| \neq |D'C|$.



Obr. 30



Obr. 31

Jiné řešení. Uvažujme *sinovou větu* v následujícím tvaru, který plyne z věty o obvodových úhlech: *Je-li R poloměr kružnice opsané trojúhelníku ABC, je* $\sin\alpha = \frac{1}{2}a/R$, *kde* $a = |BC|$. (Doplníme-li cyklicky další dvě rovnosti, dostaneme odtud snadno běžné znění sinové věty ze školních učebnic.)

Označíme-li nyní φ obvodový úhel příslušný shodným tětivám BC a CD ($0° < \varphi < 60°$), snadno zjistíme, že tětivě DA přísluší obvodový

úhel $60° - \varphi$ a tětivě AB obvodový úhel $120° - \varphi$ (obr. 31). Dokazovaná rovnost je pak dle sinové věty ekvivalentní rovnosti

$$\sin\varphi + \sin(60° - \varphi) = \sin(120° - \varphi).$$

Protože $\sin(120° - \varphi) = \sin(60° + \varphi)$, je uvedená rovnost (po jednoduché úpravě) ekvivalentní rovnosti

$$\sin\varphi = 2\cos 60° \sin\varphi,$$

která triviálně platí.

Stejně jako v předchozím řešení si uvědomíme, že rovnost $|CD| + |DA| = |AB|$ zůstane zachována, i když v daném čtyřúhelníku vyměníme obě strany CD a DA. Nový čtyřúhelník zůstane tětivový, velikost jeho vnitřního úhlu při vrcholu B se nezmění, ale místo rovnosti $|BC| = |CD|$ bude splněna rovnost $|BC| = |DA|$.

Jiné řešení. Označme délky stran čtyřúhelníku $ABCD$, který splňuje podmínky úlohy, obvyklým způsobem a, b, c, d. Protože vnitřní úhly při vrcholech B a D mají velikost 60°, resp. 120°, z kosinové věty pro trojúhelníky ABC a CDA plyne dvojím vyjádřením hodnoty $|AC|^2$ rovnost

$$a^2 + b^2 - ab = c^2 + d^2 + cd. \tag{6}$$

a) Jestliže $b = c$, lze z rovnosti (6) postupně odvodit:

$$\begin{aligned} a^2 + c^2 - ac &= c^2 + d^2 + cd, \\ a^2 - d^2 &= ac + cd, \\ (a - d)(a + d) &= c(a + d), \\ a - d &= c. \end{aligned}$$

Rovnost $a = c + d$, kterou jsme měli dokázat, tedy platí.

b) Jestliže platí $a = c + d$, dostaneme po dosazení za a do rovnosti (6)

$$(c + d)^2 + b^2 - (c + d)b = c^2 + d^2 + cd.$$

Odtud po úpravě obdržíme vztah $(b - c)(b - d) = 0$, z něhož plyne, že platí $b = c$ nebo $b = d$. Opačná implikace tedy obecně neplatí.

Jsou-li čísla x, y, z řešením dané soustavy, zřejmě platí $xyz \neq 0$. Vynásobme proto jednotlivé rovnice činiteli yz, zx resp. xy a v oboru nenulových reálných čísel řešme ekvivalentní soustavu rovnic

$$x^2yz = y + z, \quad xy^2z = x + z, \quad xyz^2 = x + y. \tag{1}$$

Sečtením levých a pravých stran této soustavy rovnic získáme po úpravě rovnici

$$(xyz - 2)(x + y + z) = 0.$$

Odtud vidíme, že platí $xyz = 2$ nebo $x + y + z = 0$.

▷ Nechť $xyz = 2$. Po dosazení za součin xyz v soustavě (1) dostaneme

$$2x = y + z, \quad 2y = x + z, \quad 2z = x + y,$$

což je ekvivalentní se soustavou

$$3x = x + y + z, \quad 3y = x + y + z, \quad 3z = x + y + z.$$

Odtud plyne $x = y = z$. S ohledem na podmínku $xyz = 2$ dostáváme $x = y = z = \sqrt[3]{2}$. Zkouškou ověříme, že trojice $(\sqrt[3]{2}, \sqrt[3]{2}, \sqrt[3]{2})$ je skutečně řešení soustavy (1), a tedy i původní soustavy rovnic.

▷ Nechť $x + y + z = 0$. Z první rovnice soustavy (1) plyne $x^2yz = -x$, odkud s ohledem na podmínku $x \neq 0$ dostaneme $xyz = -1$. Ověřme, že každá trojice nenulových reálných čísel (x, y, z) splňující soustavu dvou rovnic

$$x + y + z = 0, \quad xyz = -1 \tag{2}$$

je řešením původní soustavy. Z rovností (2) totiž plyne

$$\frac{1}{y} + \frac{1}{z} = \frac{y + z}{yz} = \frac{-x}{-1/x} = x^2$$

(s ohledem na symetrii zadané soustavy stačilo ověřit jednu rovnici).

Soustava rovnic (2) má v oboru nenulových reálných čísel nekonečně mnoho řešení, která získáme například tak, že jednu proměnnou (např. z) zvolíme jako parametr. Tím dostaneme soustavu

$$x + y = -z, \quad xy = -\frac{1}{z}.$$

Po dosazení za x z první rovnice do druhé dostaneme

$$(y+z)y = \frac{1}{z},$$

tedy

$$y^2 + yz - \frac{1}{z} = 0. \tag{3}$$

Jedná se o kvadratickou rovnici s neznámou y a parametrem z. Její diskriminant je roven $D = z^2 + 4/z$. Nutnou a postačující podmínkou k tomu, aby tato rovnice měla reálné kořeny, je nerovnost $D \geqq 0$. Vyřešením nerovnice $(z^3+4)/z \geqq 0$ dostaneme pro parametr z podmínku

$$z \in \left(-\infty, -\sqrt[3]{4}\right\rangle \cup (0, \infty). \tag{4}$$

Za podmínky (4) má kvadratická rovnice (3) kořeny

$$y_1 = \frac{-z + \sqrt{z^2+4/z}}{2} \quad \text{a} \quad y_2 = \frac{-z - \sqrt{z^2+4/z}}{2},$$

kterým podle vztahu $x = -y - z$ odpovídají hodnoty

$$x_1 = \frac{-z - \sqrt{z^2+4/z}}{2} \quad \text{a} \quad x_2 = \frac{-z + \sqrt{z^2+4/z}}{2},$$

přitom pouze v případě $z = -\sqrt[3]{4}$ platí $(x_1, y_1) = (x_2, y_2)$.

Závěr: Daná soustava má řešení $x = y = z = \sqrt[3]{2}$. Všechna ostatní řešení jsou trojice (x, y, z) tvaru

$$(x, y, z) = \left(\frac{-z \pm \sqrt{z^2+4/z}}{2}, \frac{-z \mp \sqrt{z^2+4/z}}{2}, z\right),$$

kde z je libovolné číslo splňující podmínku (4).

A – S – 1

Protože $P(x)$ je kvadratický trojčlen s nezápornými koeficienty, je nutně $a > 0$.

Nechť x je libovolné kladné reálné číslo a n přirozené. Protože

$$0 \leqq \left(\sqrt{x^n} - \frac{1}{\sqrt{x^n}}\right)^2 = x^n + \frac{1}{x^n} - 2,$$

platí

$$x^n + \frac{1}{x^n} \geqq 2, \tag{1}$$

přitom rovnost nastává, právě když $\sqrt{x^n} = 1/\sqrt{x^n}$, tj. když $x = 1$.

Protože čísla ab, bc a ca jsou podle předpokladů úlohy nezáporná, užitím nerovnosti (1) dále platí

$$\begin{aligned} P(x) \cdot P\Big(\frac{1}{x}\Big) &= (ax^2 + bx + c)\Big(a\,\frac{1}{x^2} + b\,\frac{1}{x} + c\Big) = \\ &= a^2 + b^2 + c^2 + ab\Big(x + \frac{1}{x}\Big) + bc\Big(x + \frac{1}{x}\Big) + ca\Big(x^2 + \frac{1}{x^2}\Big) \geqq \\ &\geqq a^2 + b^2 + c^2 + 2ab + 2bc + 2ca = (a + b + c)^2 = \big(P(1)\big)^2. \end{aligned}$$

Rovnost nastává, právě když $x = 1$, nebo $ab = bc = ca = 0$, což s ohledem na podmínku $a > 0$ dává $b = c = 0$.

Pro libovolné kladné reálné číslo x tedy platí

$$P(x) \cdot P\Big(\frac{1}{x}\Big) \geqq \big(P(1)\big)^2,$$

přičemž rovnost nastává právě tehdy, je-li $x = 1$ nebo $b = c = 0$.

Poznámka. Úlohu lze řešit také užitím Cauchyovy nerovnosti:

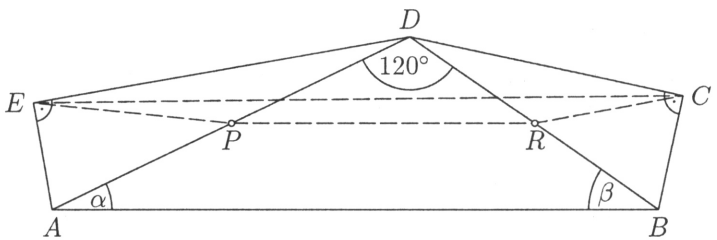
$$\begin{aligned} P(x) \cdot P\Big(\frac{1}{x}\Big) &= (ax^2 + bx + c)\Big(a\,\frac{1}{x^2} + b\,\frac{1}{x} + c\Big) = \\ &= \Big(\big(\sqrt{a}x\big)^2 + \big(\sqrt{bx}\big)^2 + \big(\sqrt{c}\big)^2\Big)\left(\left(\frac{\sqrt{a}}{x}\right)^2 + \left(\sqrt{\frac{b}{x}}\right)^2 + \big(\sqrt{c}\big)^2\right) \geqq \\ &\geqq \left(\sqrt{a}x \cdot \frac{\sqrt{a}}{x} + \sqrt{bx} \cdot \sqrt{\frac{b}{x}} + \sqrt{c} \cdot \sqrt{c}\right)^2 = (a + b + c)^2 = \big(P(1)\big)^2. \end{aligned}$$

A – S – 2

Nechť $ABCDE$ je libovolný konvexní pětiúhelník s uvažovanými vlastnostmi. Označme P, R po řadě středy stran AD, BD trojúhelníku ABD (obr. 32). Pak bude

$$|PR| = \frac{1}{2}|AB|, \qquad |CR| = \frac{1}{2}|BD|, \qquad |PE| = \frac{1}{2}|AD|, \tag{1}$$

protože PR je střední příčka trojúhelníku ABD a protože v pravoúhlém trojúhelníku je střed přepony zároveň středem jeho opsané kružnice (Thaletova věta).



Obr. 32

Z trojúhelníkové nerovnosti je zřejmé, že pro délku úhlopříčky CE platí

$$|CE| \leqq |CR| + |RP| + |PE| = s,$$

kde délka s lomené čáry $CRPE$ je podle (1) zároveň rovna polovině obvodu trojúhelníku ABD.

Dále zkoumejme, kdy bude mít trojúhelník ABD daných vlastností ($|AB| = 6\,\text{cm}$, $|\sphericalangle ADB| = 120°$) největší obvod. Označíme-li α a β (obr. 32) velikosti vnitřních úhlů při vrcholech A a B trojúhelníku ABD ($\alpha + \beta = 60°$), dostaneme ze sinové věty v trojúhelníku ABD

$$|BD| = |AB| \frac{\sin\alpha}{\sin 120°}, \quad |AD| = |AB| \frac{\sin\beta}{\sin 120°}.$$

Sečtením obou předchozích rovností vyjde

$$|AD| + |BD| = |AB| \frac{\sin\alpha + \sin\beta}{\sin 120°} =$$
$$= 2|AB| \frac{\sin 30°}{\sin 120°} \cos\frac{\alpha - \beta}{2} \leqq 2|AB| \frac{\sqrt{3}}{3},$$

přičemž rovnost v poslední nerovnosti nastává, právě když $\cos\frac{1}{2}(\alpha - \beta) = 1$, tj. pro $\alpha = \beta = 30°$. Trojúhelník ABD má tedy největší obvod, právě když je rovnoramenný a jeho úhly při základně AB mají velikost 30°. Vzhledem k tomu, že $|AB| = 6\,\text{cm}$, platí pro libovolný pětiúhelník $ABCDE$ požadovaných vlastností

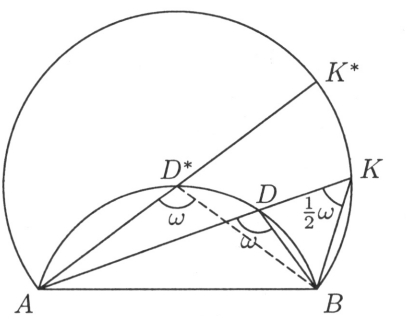
$$|CE| \leqq s = \frac{1}{2}(|AB| + |AD| + |BD|) \leqq \frac{1}{2}|AB| \left(1 + 2\frac{\sqrt{3}}{3}\right) =$$
$$= \left(3 + 2\sqrt{3}\right)\text{cm}.$$

Přitom pro uvažovaný pětiúhelník $ABCDE$ v situaci, kdy je trojúhelník ABD rovnoramenný a vrcholy C, E leží na přímce RP, skutečně platí $|CE| = (3 + 2\sqrt{3})$ cm.

Největší délka úhlopříčky CE pětiúhelníku $ABCDE$ vyhovujícího podmínkám úlohy je tedy $(3 + 2\sqrt{3})$ cm.

Poznámka. V druhé části řešení jsme (pro konkrétní hodnotu $\omega = 120°$) ukázali, že trojúhelník ABD s danou stranou AB a daným úhlem ω při vrcholu D má největší obvod, právě když je rovnoramenný se základnou AB. To plyne i z následující úvahy:

Bod D probíhá oblouk, z něhož je úsečka AB vidět pod úhlem ω. Na polopřímce opačné k DA (obr. 33) sestrojme bod K tak, aby $|DB| = |DK|$. Z rovnoramenného trojúhelníku BDK plyne, že $|\sphericalangle AKB| = \frac{1}{2}\omega$. Bod K proto leží na oblouku, z něhož je úsečka AB vidět pod úhlem $\frac{1}{2}\omega$. Délka $|AK| = |AD| + |BD|$ bude tudíž největší, právě když bude úsečka AK průměrem AK^* zmíněného oblouku; tehdy je bod D středem D^* příslušné kružnice, takže platí $|AD^*| = |BD^*| = |D^*K^*|$.



Obr. 33

A – S – 3

Odečtením první rovnice dané soustavy od druhé dostaneme rovnici

$$y^2 - x^2 + 2zx - 2yz = 6(z + x - 2) - 6(y + z - 2),$$

kterou upravíme na tvar

$$(x - y)(x + y - 2z + 6) = 0.$$

Podobně odečtením první rovnice soustavy od třetí dostaneme

$$(x - z)(x + z - 2y + 6) = 0.$$

Daná soustava je proto ekvivalentní se soustavou rovnic

$$\begin{aligned} x^2 + 2yz - 6(y + z - 2) &= 0, \\ (x - y)(x + y - 2z + 6) &= 0, \\ (x - z)(x + z - 2y + 6) &= 0. \end{aligned} \tag{2}$$

Vzhledem ke druhé a třetí rovnici této soustavy je možno rozlišit čtyři případy.

1. Nechť $(x - y = 0) \wedge (x - z = 0)$. Pak $x = y = z$ a dosazením za y a z do první rovnice soustavy (2) dostaneme rovnici

$$3x^2 - 12x + 12 = 0,$$

   která má dvojnásobný reálný kořen $x = 2$. Proto trojice $(x, y, z) = (2, 2, 2)$ je v tomto případě jediným řešením dané soustavy.
2. Nechť $(x - y = 0) \wedge (x + z - 2y + 6 = 0)$. Pak $y = x$ a $z = x - 6$. Dosazením do první rovnice soustavy (2) dostaneme po úpravě rovnici

$$3x^2 - 24x + 48 = 0,$$

   která má dvojnásobný reálný kořen $x = 4$. Proto trojice $(x, y, z) = (4, 4, -2)$ je v tomto případě jediným řešením dané soustavy.
3. Nechť $(x+y-2z+6 = 0) \wedge (x-z = 0)$. Podobně jako v předcházejícím případě dostaneme jediné řešení $(x, y, z) = (4, -2, 4)$.
4. Nechť $(x + y - 2z + 6 = 0) \wedge (x + z - 2y + 6 = 0)$. Odečtením druhé rovnice od první dostaneme, že $3y - 3z = 0$, tedy $y = z$. Z prvního předpokladu tak máme $y = x + 6$. Dosazením do první rovnice soustavy (2) dostaneme po úpravě rovnici

$$3x^2 + 12x + 12 = 0,$$

   která má dvojnásobný reálný kořen $x = -2$. Proto trojice $(x, y, z) = (-2, 4, 4)$ je v tomto případě jediným řešením dané soustavy.

Daná soustava má v oboru reálných čísel čtyři řešení (x, y, z), kterými jsou trojice $(2, 2, 2)$, $(4, 4, -2)$, $(4, -2, 4)$ a $(-2, 4, 4)$.

Poznámka. Pokud si všimneme, že sečtením všech tří rovnic dané soustavy dostaneme po úpravě

$$(x + y + z - 6)^2 = 0,$$

pak např. z podmínky $z + x - 2y + 6 = 0$ přímo plyne $y = 4$, což předcházející úvahy zjednoduší.

A – II – 1

Každý pětimístný palindrom p se dá zapsat ve tvaru $p = \overline{abcba}$, kde a, b, c jsou číslice v desítkové soustavě, $a \neq 0$. Z vyjádření

$$p = 10\,001a + 1\,010b + 100c = 37(270a + 27b + 3c) + 11(a + b - c)$$

plyne, že p je dělitelné číslem 37, právě když je číslem 37 dělitelné číslo $a + b - c$. Vzhledem k tomu, že a, b, c jsou číslice ($a \neq 0$), je $-8 \leqq a + b - c \leqq 18$. Proto je číslo $a + b - c$ dělitelné 37, právě když $a + b - c = 0$, neboli $c = a + b$. Číslice a, b tedy musejí splňovat podmínku $a + b \leqq 9$.

Ke každému $a \in \{1, 2, 3, \ldots, 9\}$ lze číslici b zvolit $10 - a$ způsoby tak, aby platilo $a + b \leqq 9$ ($b \in \{0, 1, 2, \ldots, 9 - a\}$). Číslice c je pak určena jednoznačně jako součet $a + b$. Palindromů s číslicí $a = 1$ je proto 9, palindromů s číslicí $a = 2$ je 8 atd.; konečně pro číslici $a = 9$ existuje právě jeden palindrom.

Počet všech pětimístných palindromů, které jsou dělitelné číslem 37, je tedy

$$9 + 8 + 7 + \ldots + 1 = 45.$$

A – II – 2

Počet vyhovujících slov délky $n \geqq 2$, která končí dvojicemi písmen AA, AB, BA označme postupně $(aa)_n$, $(ab)_n$, $(ba)_n$; počet vyhovujících slov délky $n \geqq 1$, která končí písmenem A, resp. B, označme a_n, resp. b_n. Pro všechna přirozená čísla $n \geqq 2$ platí:

$$\begin{aligned} a_n &= (aa)_n + (ba)_n, \\ b_n &= (ab)_n, \\ p_n &= a_n + b_n = (aa)_n + (ba)_n + (ab)_n. \end{aligned}$$

Existují právě dvě vyhovující slova délky jedna, a to slova A a B, a právě tři vyhovující slova délky dva, a to slova AA, AB, BA, proto $a_1 = b_1 = 1$, $p_1 = 2$, $(aa)_2 = (ab)_2 = (ba)_2 = 1$, $a_2 = 2$, $b_2 = 1$, $p_2 = 3$.

Každé vyhovující slovo délky $n \geqq 3$, které končí dvojicí písmen AA, dostaneme tak, že připíšeme písmeno A na konec slova délky $n-1$ končícího dvojicí BA. Proto platí

$$(aa)_n = (ba)_{n-1}.$$

Analogicky zjistíme, že pro každé $n \geqq 3$ platí rovněž vztahy

$$(ba)_n = (ab)_{n-1},$$
$$(ab)_n = (aa)_{n-1} + (ba)_{n-1}.$$

Protože nás zajímá pouze parita přirozeného čísla p_n a výrazů, pomocí kterých ho počítáme, můžeme na základě uvedených rovností sestavit tabulku ze symbolů S a L, jimž odpovídají sudá resp. lichá čísla. Dostaneme

n	1	2	3	4	5	6	7	8	9	...
$(aa)_n$		L	L	L	S	S	L	S	L	...
$(ba)_n$		L	L	S	S	L	S	L	L	...
$(ab)_n$		L	S	S	L	S	L	L	L	...
a_n	L	S	S	L	S	L	L	L	S	...
b_n	L	L	S	S	L	S	L	L	L	...
p_n	S	L	S	L	L	L	S	S	L	...

Tato tabulka je nutně periodická, protože existuje jen osm různých uspořádaných trojic písmen S a L, takže nejdéle po osmi sloupcích se vzhledem k dokázané rekurenci začnou hodnoty posloupností $((aa)_n)$, $((ba)_n)$, $((ab)_n)$ opakovat. Hodnoty posloupností (a_n), (b_n), (p_n) jsou z nich odvozeny, takže se začnou opakovat rovněž. Z tabulky vidíme, že její perioda je 7 (první dva shodné sloupce jsou pro $n = 2$ a $n = 9$). A protože v příslušném úseku tabulky je dvojice sousedních sudých čísel p_7, p_8 jediná, jsou obě čísla p_n a p_{n+1} sudá, právě když je číslo n dělitelné sedmi.

Poznámky. Z výše uvedených vztahů můžeme odvodit rekurentní rovnice pro čísla a_n a b_n. Pro všechna přirozená čísla $n \geqq 4$ platí

$$a_n = (aa)_n + (ba)_n = (ba)_{n-1} + (ab)_{n-1} =$$
$$= (ab)_{n-2} + (ab)_{n-1} = b_{n-2} + b_{n-1},$$
$$b_n = (ab)_n = (aa)_{n-1} + (ba)_{n-1} = a_{n-1}.$$

Tyto rovnice také můžeme odvodit následující úvahou. Vyhovující slovo končící písmenem A má koncovku BA nebo BAA, počet slov prvního

typu je b_{n-1}, slov druhého typu je b_{n-2}. Vyhovující slovo končící písmenem B má nutně koncovku AB a těchto slov je a_{n-1}.

Ze vztahů uvedených v předchozím odstavci se dá odvodit rekurentní rovnice přímo pro čísla p_n. Pro každé $n \geqq 4$ totiž platí

$$a_n = b_{n-1} + b_{n-2} = a_{n-2} + a_{n-3},$$
$$b_n = a_{n-1} = b_{n-2} + b_{n-3}.$$

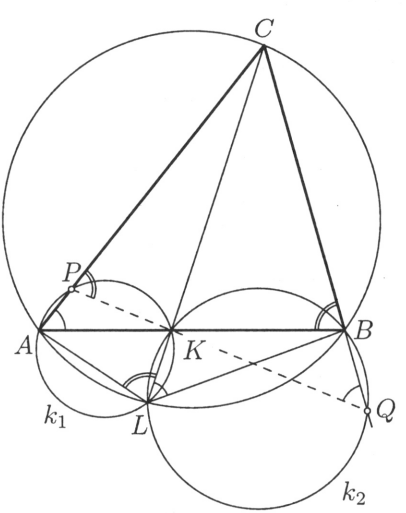
Vzhledem k tomu, že $p_n = a_n + b_n$, dostaneme sečtením těchto vztahů rovnici

$$p_n = p_{n-2} + p_{n-3},$$

kterou můžeme odvodit i takto: Každé vyhovující slovo délky n má právě jednu z koncovek $ABAA$, ABA, BAB, $BAAB$, přitom koncovky ABA a BAB má právě p_{n-2} slov, zatímco koncovky $ABAA$ a $BAAB$ má právě p_{n-3} slov.

A – II – 3

a) V tětivovém čtyřúhelníku $ALBC$ platí $\alpha = |\sphericalangle BAC| = |\sphericalangle BLC|$ a $\beta = |\sphericalangle ABC| = |\sphericalangle ALC|$ (obr. 34). Z rovnosti obvodového a příslušného úsekového úhlu pro tětivu AK v kružnici k_1 vyplývá, že přímka AC je



Obr. 34

tečnou ke kružnici k_1, právě když platí $|\sphericalangle CAK| = |\sphericalangle ALK|$, tj. právě když $\alpha = \beta$. Z analogických důvodů je přímka BC tečnou ke kružnici k_2, právě když $\beta = \alpha$. Přímka AC je proto tečnou ke kružnici k_1, právě když přímka BC je tečnou ke kružnici k_2, což jsme chtěli dokázat.

b) Podle části a) víme, že platí $\alpha = |\sphericalangle BAC| = |\sphericalangle BLK|$ a $\beta = |\sphericalangle ABC| = |\sphericalangle ALK|$. Bez újmy na obecnosti můžeme předpokládat, že platí $\alpha < \beta$. Tečna v bodě A ke kružnici k_1 svírá s tětivou AK úsekový úhel $\beta > \alpha$, proto leží bod P na polopřímce AC, zatímco bod Q leží analogicky na polopřímce opačné k BC. Z tětivových čtyřúhelníků $ALKP$ a $BQLK$ plynou rovnosti $|\sphericalangle KPC| = \beta$ a $|\sphericalangle BQK| = \alpha$ (obr. 34). Trojúhelníky APK a QBK se proto shodují ve dvou úhlech (u vrcholů A, Q a P, B). Shodují se tedy i v úhlu při společném vrcholu K:

$$|\sphericalangle AKP| = |\sphericalangle BKQ| \ (= \beta - \alpha).$$

Odtud plyne, že body P, K, Q leží na téže přímce. Tím je tvrzení části b) dokázáno.

Poznámka. Dokázali jsme vlastně následující tvrzení: Je-li trojúhelník ABC rovnoramenný s rameny AC, BC, dotýkají se obě ramena odpovídajících kružnic k_1 a k_2 ve vrcholech A a B; není-li rovnoramenný, protínají jeho strany AC a BC odpovídající kružnice k_1 a k_2 v dalších bodech P a Q ($P \neq A$, $Q \neq B$), přičemž jejich spojnice PQ prochází daným bodem K.

A – II – 4

Užitím sinové věty v trojúhelnících BKA a CKA dostaneme

$$\frac{|BK|}{|AK|} = \frac{\sin\frac{\alpha}{2}}{\sin\beta} \qquad \text{a} \qquad \frac{|CK|}{|AK|} = \frac{\sin\frac{\alpha}{2}}{\sin\gamma}.$$

Sečtením obou předešlých rovností vyjde

$$\frac{|BC|}{|AK|} = \frac{|BK|}{|AK|} + \frac{|CK|}{|AK|} = \sin\frac{\alpha}{2}\left(\frac{1}{\sin\beta} + \frac{1}{\sin\gamma}\right).$$

Vynásobíme-li obě strany poslední rovnosti výrazem $2\cos\frac{\alpha}{2}$, obdržíme po úpravě

$$2\,\frac{|BC|}{|AK|}\cos\frac{\alpha}{2} = \sin\alpha\left(\frac{1}{\sin\beta} + \frac{1}{\sin\gamma}\right). \tag{1}$$

Cyklickou záměnou získáme další dvě analogické rovnosti

$$2\frac{|CA|}{|BL|}\cos\frac{\beta}{2} = \sin\beta\left(\frac{1}{\sin\gamma}+\frac{1}{\sin\alpha}\right), \tag{2}$$

$$2\frac{|AB|}{|CM|}\cos\frac{\gamma}{2} = \sin\gamma\left(\frac{1}{\sin\alpha}+\frac{1}{\sin\beta}\right). \tag{3}$$

Sečtením rovností (1), (2) a (3) dostaneme po dělení dvěma rovnost

$$\frac{|BC|}{|AK|}\cos\frac{\alpha}{2}+\frac{|CA|}{|BL|}\cos\frac{\beta}{2}+\frac{|AB|}{|CM|}\cos\frac{\gamma}{2} =$$
$$\frac{1}{2}\left(\frac{\sin\alpha}{\sin\beta}+\frac{\sin\beta}{\sin\alpha}\right)+\frac{1}{2}\left(\frac{\sin\beta}{\sin\gamma}+\frac{\sin\gamma}{\sin\beta}\right)+\frac{1}{2}\left(\frac{\sin\gamma}{\sin\alpha}+\frac{\sin\alpha}{\sin\gamma}\right).$$

Vzhledem k tomu, že $\sin\alpha$, $\sin\beta$, $\sin\gamma$ jsou kladná čísla, můžeme každý ze tří výrazů v závorkách na pravé straně poslední rovnosti odhadnout zdola číslem dvě (využíváme známou nerovnost $a/b+b/a \geqq 2$, která je pro libovolná kladná čísla a, b ekvivalentní se zřejmou nerovností $(a-b)^2 \geqq \geqq 0$). Odtud plyne požadovaná nerovnost a důkaz je hotov.

A – III – 1

Vyhovuje-li nějaká trojice $(x, y, z) \in \mathbb{R}^3$ $(xyz \neq 0)$ podmínkám úlohy, je řešením následující soustavy nerovnic

$$\begin{aligned} x^2+y^2+z^2 &\leqq 6+x^2-\frac{8}{x^4}, \\ x^2+y^2+z^2 &\leqq 6+y^2-\frac{8}{y^4}, \\ x^2+y^2+z^2 &\leqq 6+z^2-\frac{8}{z^4}, \end{aligned} \qquad \text{tj.} \qquad \begin{aligned} \frac{8}{x^4}+y^2+z^2 &\leqq 6, \\ x^2+\frac{8}{y^4}+z^2 &\leqq 6, \\ x^2+y^2+\frac{8}{x^4} &\leqq 6. \end{aligned}$$

Sečtením všech tří nerovnic této soustavy dostaneme nerovnici

$$\left(\frac{8}{x^4}+x^2+x^2\right)+\left(\frac{8}{y^4}+y^2+y^2\right)+\left(\frac{8}{z^4}+z^2+z^2\right) \leqq 18.$$

Výrazy v každé ze tří závorek na levé straně lze odhadnout užitím nerovnosti mezi aritmetickým a geometrickým průměrem trojice kladných čísel. Obdržíme tak postupně

$$18 \geqq \left(\frac{8}{x^4}+x^2+x^2\right)+\left(\frac{8}{y^4}+y^2+y^2\right)+\left(\frac{8}{z^4}+z^2+z^2\right) \geqq$$
$$\geqq 3\sqrt[3]{\frac{8}{x^4}\cdot x^2\cdot x^2}+3\sqrt[3]{\frac{8}{y^4}\cdot y^2\cdot y^2}+3\sqrt[3]{\frac{8}{z^4}\cdot z^2\cdot z^2} = 18.$$

Odtud plyne, že v každé ze tří použitých nerovností mezi aritmetickým a geometrickým průměrem nastává rovnost, takže příslušná trojice čísel má vždy tři stejné složky. Musí tedy současně platit

$$\frac{8}{x^4} = x^2, \qquad \frac{8}{y^4} = y^2, \qquad \frac{8}{z^4} = z^2,$$

tj.

$$x^6 = y^6 = z^6 = 8.$$

Z poslední podmínky bezprostředně plyne

$$(x, y, z) = \left(\varepsilon_1\sqrt{2}, \varepsilon_2\sqrt{2}, \varepsilon_3\sqrt{2}\right), \quad \text{kde } \varepsilon_i \in \{-1; 1\} \text{ pro } i = 1, 2, 3. \quad (1)$$

Vzhledem k užité důsledkové úpravě je nutno provést zkoušku, pomocí níž zjistíme, že všech 8 trojic reálných čísel určených vztahem (1) vyhovuje podmínkám úlohy.

Jiné řešení. Nechť trojice $(x, y, z) \in \mathbb{R}^3$ $(xyz \neq 0)$ je řešením dané úlohy. Označme

$$A = \min\{x^2, y^2, z^2\} > 0.$$

Potom platí

$$\min\left\{x^2 - \frac{8}{x^4}, y^2 - \frac{8}{y^4}, z^2 - \frac{8}{z^4}\right\} = A - \frac{8}{A^2}.$$

Proto též

$$A + A + A \leqq x^2 + y^2 + z^2 \leqq 6 + \min\left\{x^2 - \frac{8}{x^4}, y^2 - \frac{8}{y^4}, z^2 - \frac{8}{z^4}\right\} =$$
$$= 6 + A - \frac{8}{A^2}.$$

Po úpravě dostaneme nerovnost, jejíž pravou stranu odhadneme užitím nerovnosti mezi aritmetickým a geometrickým průměrem:

$$6 \geqq A + A + \frac{8}{A^2} \geqq 3\sqrt[3]{A \cdot A \cdot \frac{8}{A^2}} = 6.$$

To znamená, že ve všech užitých nerovnostech musí nastat rovnost, proto

$$2 = A = x^2 = y^2 = z^2.$$

Zkouškou opět ověříme, že všechny trojice určené vztahem (1) jsou řešením zadané nerovnice.

A – III – 2

Počet vyhovujících slov délky n, která končí písmenem A, resp. B, označme a_n, resp. b_n. Platí

$$p_n = a_n + b_n. \tag{1}$$

Nechť $n \geqq 4$. Vyhovující slovo končící písmenem A má jednu z koncovek BA, BAA, nebo $BAAA$. Počet slov prvního typu je b_{n-1}, druhého typu b_{n-2}, třetího typu b_{n-3}. Proto

$$a_n = b_{n-1} + b_{n-2} + b_{n-3}. \tag{2}$$

Podobně pro $n \geqq 3$ má vyhovující slovo končící písmenem B jednu z koncovek AB, ABB, tudíž

$$b_n = a_{n-1} + a_{n-2}. \tag{3}$$

Nechť dále $n \geqq 6$; každé z čísel b_i ve vztahu (2) vyjádřeme pomocí (3), dostaneme tak

$$\begin{aligned} a_n &= b_{n-1} + b_{n-2} + b_{n-3} = \\ &= (a_{n-2} + a_{n-3}) + (a_{n-3} + a_{n-4}) + (a_{n-4} + a_{n-5}) = \\ &= a_{n-2} + 2a_{n-3} + 2a_{n-4} + a_{n-5}. \end{aligned} \tag{4}$$

Podobně dostaneme

$$\begin{aligned} b_n &= a_{n-1} + a_{n-2} = \\ &= (b_{n-2} + b_{n-3} + b_{n-4}) + (b_{n-3} + b_{n-4} + b_{n-5}) = \\ &= b_{n-2} + 2b_{n-3} + 2b_{n-4} + b_{n-5}. \end{aligned} \tag{5}$$

Sečtením vztahů (4) a (5) dostaneme dle (1)

$$p_n = p_{n-2} + 2p_{n-3} + 2p_{n-4} + p_{n-5}.$$

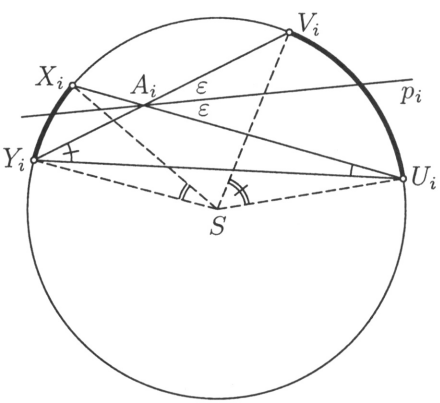
Proto pro libovolné přirozené číslo $n \geqq 6$ platí

$$\frac{p_n - p_{n-2} - p_{n-5}}{p_{n-3} + p_{n-4}} = 2,$$

tudíž zadaný zlomek má hodnotu 2 i pro $n = 2\,004$.

A – III – 3

Pro libovolné i, $1 \leqq i \leqq 121$, označme M_i množinu všech bodů X kružnice k, pro něž úsečka A_iX svírá s odpovídající přímkou p_i úhel velikosti menší než $\varepsilon = 21°$. Množina M_i je zřejmě tvořena dvěma oblouky X_iY_i a U_iV_i (obr. 35). Oběma uvažovaným obloukům kružnice k odpovídá dvojice středových úhlů X_iSY_i a U_iSV_i, kde S je střed dané kružnice k. Ukážeme, že pro každé $i \in \{1, 2, \ldots, 121\}$ platí $|\sphericalangle X_iSY_i| + |\sphericalangle U_iSV_i| = 4\varepsilon = 84°$.



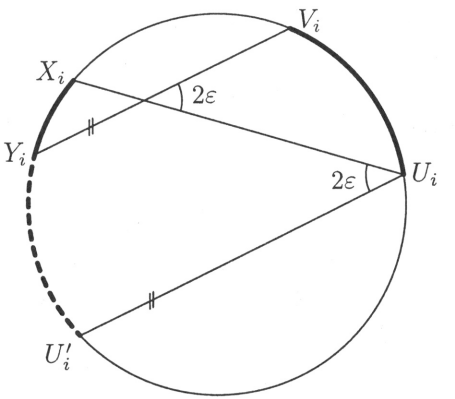
Obr. 35

V trojúhelníku $A_iY_iU_i$ je součet velikostí vnitřních úhlů při vrcholech Y_i a U_i roven velikosti vedlejšího úhlu při vrcholu A_i, tj. 2ε. Avšak součet obou uvažovaných vnitřních úhlů v trojúhelníku $A_iY_iU_i$ je roven součtu obvodových úhlů odpovídajících obloukům X_iY_i a U_iV_i. Ze vztahu mezi obvodovým a středovým úhlem dostáváme

$$|\sphericalangle X_iSY_i| + |\sphericalangle U_iSV_i| = 2 \cdot 2\varepsilon = 4\varepsilon = 84°.$$

Celkově tak 121 uvažovaným tětivám p_i a jejich bodům A_i odpovídá 121 dvojic oblouků X_iY_i a U_iV_i kružnice k s celkovou obloukovou délkou $121 \cdot 84° = 10\,164°$. Pokud každý bod X kružnice k náleží nejvýše 28 množinám M_i, musí být uvedený součet všech obloukových délek nejvýše roven $28 \cdot 360° = 10\,080°$, což neplatí. Proto existuje aspoň jeden bod kružnice k, který náleží současně aspoň 29 množinám M_i, což jsme měli dokázat.

Poznámka. Že oběma obloukům X_iY_i a U_iV_i odpovídá dohromady středový úhel 4ε, nahlédneme snadno i z obr. 36, neboť oblouky $U_i'Y_i$ a U_iV_i jsou shodné.



Obr. 36

A – III – 4

Pro $n = 1, 2, 3$ je daný součet roven celým číslům 1, 3, resp. 5. Předpokládejme proto dále, že $n > 3$. Jednoduchou úpravou dostaneme

$$\frac{n}{1!} + \frac{n}{2!} + \ldots + \frac{n}{(n-2)!} + \frac{n}{(n-1)!} + \frac{n}{n!} =$$
$$= \frac{n(n-1)\cdot\ldots\cdot 2 + n(n-1)\cdot\ldots\cdot 3 + \ldots + n(n-1) + n + 1}{(n-1)!}.$$

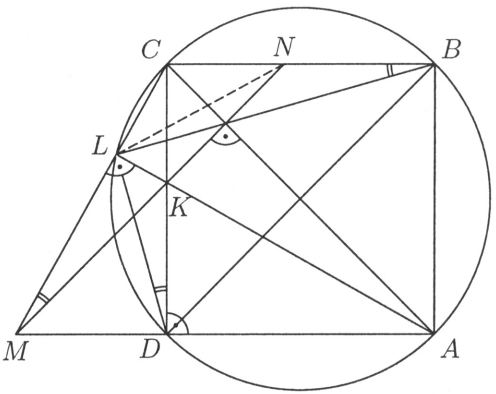
Je-li poslední zlomek celé číslo, je nutně číslo $n-1$ dělitelem jeho čitatele. Proto je číslo $n-1$ dělitelem čísla $n+1$. Protože největší společný dělitel dvou čísel je dělitelem i jejich rozdílu, je největší společný dělitel čísel $n-1$ a $n+1$ dělitelem čísla 2, takže $n-1 \in \{1, 2\}$, což je ve sporu s předpokladem $n > 3$.

Daný součet je celé číslo pro přirozená čísla n z množiny $\{1, 2, 3\}$.

A – III – 5

Úhlopříčka AC je průměrem kružnice opsané čtverci $ABCD$, takže podle Thaletovy věty je úhel ALC pravý (obr. 37). Bod K je tak průsečíkem výšek CD a AL v trojúhelníku ACM, takže i přímka MK je kolmá

na AC a protíná stranu BC daného čtverce v jejím vnitřním bodě N, neboť $MK \parallel DB$.



Obr. 37

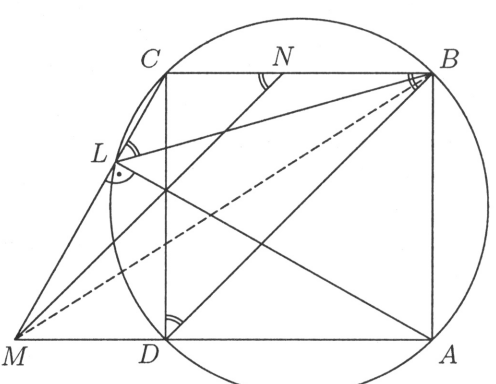
Tvrzení lze úlohy dokázat několika způsoby.

1. Čtyřúhelníky $BCLD$ a $KLMD$ jsou tětivové, proto podle věty o obvodových úhlech postupně platí

$$|\sphericalangle NBL| = |\sphericalangle CBL| = |\sphericalangle CDL| = |\sphericalangle KDL| = |\sphericalangle KML| = |\sphericalangle NML|.$$

Protože body B a M leží v téže polorovině vyťaté přímkou NL, leží body B, L, M, N na téže kružnici.

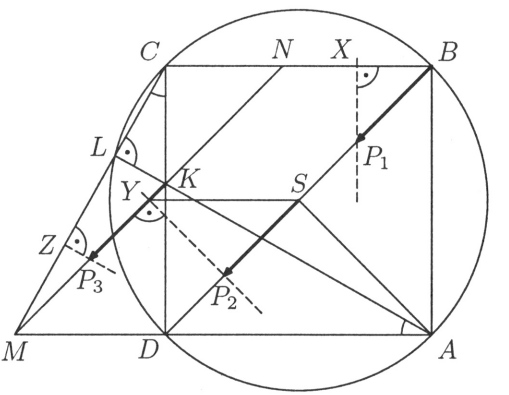
2. Protože $MN \parallel DB$, je $|\sphericalangle MNC| = 45°$, rovněž úhel BLC nad tětivou BC kružnice k má velikost $45°$ (obr. 38), je tedy $|\sphericalangle BLM| =$



Obr. 38

$= |\sphericalangle BNM| = 135°$. Body L a N zřejmě leží v téže polorovině vyťaté přímkou MB, proto leží body B, L, M, N na téže kružnici.

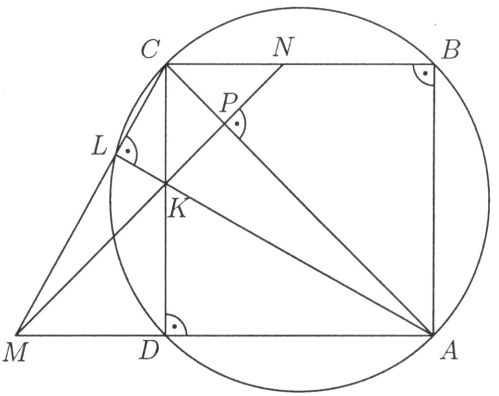
3. (Oceněné řešení *Martina Seleckého.*) Ukážeme, že osy úseček BN, MN a LM se protínají v jednom bodě. Označme X, Y a Z středy těchto úseček (obr. 39). Osa úsečky BN je zřejmě rovnoběžná s AB. Z rovnosti obvodových úhlů $|\sphericalangle DAL| = |\sphericalangle DCL|$ nad tětivou DL kružnice opsané



Obr. 39

danému čtverci plyne, že trojúhelníky DAK a DCM jsou shodné (*usu*), takže $|MD| = |DK|$. Trojúhelník MDK je tedy pravoúhlý rovnoramenný, takže $|\sphericalangle DMN| = 45°$, neboli $MN \parallel BD$. To znamená, že osa úsečky MN je rovnoběžná s přímkou SA. Konečně úhel CLA je pravý úhel nad průměrem AC kružnice opsané danému čtverci, takže osa úsečky LM je rovnoběžná s KA. Označme P_1, P_2 průsečíky os úseček NB a MN s úhlopříčkou BD a P_3 průsečík osy úsečky LM s přímkou MN (obr. 39), přitom zřejmě je P_3 středem úsečky KM. Protože $|MD| = |YS| = |NB|$, je také $|BP_1| = |SP_2| = |KP_3|$ (stačí si uvědomit shodnost příslušných pravoúhlých rovnoramenných trojúhelníků $BNP_1 \cong SYP_2 \cong DKP_3$). Zjistili jsme, že osy úseček BN, MN a LM dostaneme po řadě rovnoběžným posunutím přímek BA, SA a KA o vektor $\boldsymbol{BP_1}$, a protože uvedené přímky mají společný bod A, je společným bodem všech tří os bod, který dostaneme posunutím bodu A o týž vektor.

4. Označme P patu výšky z vrcholu M na stranu AC a uvažujme čtyřúhelníky $ABNP$, $APKD$ a $DKLM$ (obr. 40). Podle Thaletovy věty jsou všechny tři čtyřúhelníky tětivové. Vrchol C daného čtverce $ABCD$ leží vně každé ze tří kružnic opsaných uvažovaným tětivovým čtyřúhel-



Obr. 40

níkům, takže užitím věty o mocnosti bodu C ke kružnicím opsaným po řadě čtyřúhelníkům $ABNP$, $APKD$, $DKLM$ obdržíme následující tři rovnosti

$$|CN| \cdot |CB| = |CP| \cdot |CA|,$$
$$|CP| \cdot |CA| = |CK| \cdot |CD|,$$
$$|CK| \cdot |CD| = |CL| \cdot |CM|,$$

z nichž bezprostředně vyplývá rovnost

$$|CN| \cdot |CB| = |CL| \cdot |CM|.$$

Odtud již plyne, že body B, L, M, N leží na téže kružnici.

A – III – 6

Nechť f je libovolná z hledaných funkcí. Označme $f(1) = p$, vzhledem k podmínkám úlohy platí $p > 0$.

V daném vztahu položme $x = 1$, $y = 1$. Po úpravě dostaneme

$$p = f(p). \tag{1}$$

V daném vztahu dále položme $x = p$, $y = 1$. Potom

$$p^2\big(f(p) + p\big) = (p + 1)f\big(f(p)\big)$$

a podle (1) vyjde

$$2p^3 = (p + 1)p.$$

Tato algebraická rovnice má tři reálné kořeny $-\frac{1}{2}$, 0, 1. Jediný kořen vyhovující podmínce $p > 0$ je $p = 1$, tedy

$$f(1) = 1. \tag{2}$$

Nechť t je libovolné kladné reálné číslo. V daném vztahu položme $x = 1$, $y = t$, takže vzhledem k (2) dostaneme

$$1 + f(t) = (1 + t)f(t).$$

Odtud po úpravě

$$f(t) = \frac{1}{t}. \tag{3}$$

Dosazením snadno ověříme, že funkce $f(t) = 1/t$ vyhovuje rovnici ze zadání.

Funkce určená vztahem (3) je jediné řešení dané úlohy.

Jiné řešení. Předpokládejme, že existuje funkce daných vlastností, a libovolnou z takových funkcí označme f.

Nechť t je libovolné kladné reálné číslo. V daném vztahu položme $x = t$, $y = t$. Po úpravě dostaneme

$$tf(t) = f\big(tf(t)\big).$$

Odtud plyne, že množina $\mathsf{P} = \{p \in \mathbb{R}_+ : p = f(p)\}$ pevných bodů funkce f je neprázdná, protože pro každé kladné reálné číslo t je $tf(t) \in \mathsf{P}$.

Předpokládejme, že množina P obsahuje alespoň dvě různá čísla, a označme je a a b. V daném vztahu položme $x = a$, $y = b$, dostaneme tak

$$a^2\big(f(a) + f(b)\big) = (a + b)f\big(f(a)b\big),$$

a protože $a = f(a)$, $f(a) + f(b) = a + b \neq 0$, vyjde po úpravě

$$a^2 = f(ab). \tag{1}$$

Položíme-li naopak $x = b$, $y = a$, dostaneme obdobně

$$b^2 = f(ab). \tag{2}$$

Protože a a b jsou kladná čísla, plyne ze vztahů (1) a (2) rovnost $a = b$, což je spor s předpokladem, že množina P obsahuje dvě různá čísla. Množina P tedy obsahuje jediné číslo p ($p \in \mathbb{R}_+$). Z předcházejících vztahů

vyplývá, že pro každé kladné reálné číslo t platí $tf(t) = p$, proto je funkce f nutně tvaru

$$f(t) = \frac{p}{t}.$$

Dosadíme-li tento předpis do původního vztahu, dostaneme pro všechna $x, y \in \mathbb{R}_+$

$$x^2\left(\frac{p}{x} + \frac{p}{y}\right) = (x+y)\frac{p}{\frac{p}{x}\,y},$$

což po úpravě dává $p = 1$.

Funkce f daná pro všechna kladná reálná čísla t předpisem

$$f(t) = \frac{1}{t}$$

je tedy jediná funkce, která vyhovuje zadání.

Texty úloh

P – I – 1

Síť

Firma Připojse poskytovala svým zákazníkům velmi spolehlivé připojení na Internet. Vybudovala si pro tento účel svou vlastní datovou síť spojující několik největších měst v zemi. Síť se skládá z uzlů umístěných ve městech a z linek, které vedou mezi městy. Každá linka spojuje některé dva uzly. Síť měla tu vlastnost, že v případě přerušení jedné libovolné linky zůstala plně funkční, tzn. zbývající linky zajišťovaly propojení mezi každými dvěma městy.

Nedávno se firma rozhodla rozšířit své služby i pro zákazníky v dalších městech. Zřídila proto řadu nových linek, pomocí nichž byla tato města připojena k již existující síti. Nyní by ve firmě potřebovali vědět, zda je jejich síť stále ještě dostatečně spolehlivá, tj. zda i nadále existuje možnost spojení mezi každými dvěma městy i v případě výpadku jedné libovolné linky.

Soutěžní úloha. Napište program, který přečte ze vstupu seznam uzlů a linek tvořících síť a zjistí, zda má síť tu vlastnost, že pokud se libovolná jedna linka přeruší, všechna města v síti mohou mezi sebou nadále komunikovat pomocí zbývajících linek.

Formát vstupu: První řádek vstupního textového souboru `sit.in` obsahuje dvě kladná celá čísla n a m, kde n je počet měst propojených v síti a m je počet linek ($n \leqq 100$). Pro jednoduchost jsou města označena čísly $1, 2, \ldots, n$. Na každém z následujících m řádků vstupního souboru jsou zapsána dvě čísla, která určují města spojená linkou.

Můžete předpokládat, že je možné prostřednictvím existujících linek komunikovat mezi každými dvěma městy v síti a že žádné dvě linky nespojují stejnou dvojici měst.

Formát výstupu: Výstupní textový soubor `sit.out` obsahuje jediný řádek, na němž je zapsáno slovo „ANO“, jestliže lze v síti zadané na

vstupu komunikovat mezi libovolnými dvěma městy i po přerušení libovolné jedné linky, a slovo „NE“, pokud síť tuto vlastnost nemá.

Příklad 1:

```
sit.in      sit.out
5 6         ANO
1 2
2 3
3 1
3 4
4 5
2 5
```

Příklad 2:

```
sit.in      sit.out
4 4         NE
1 2         (Pokud se přeruší linka
2 3         mezi městy 3 a 4, tato
3 1         města spolu nebudou
3 4         moci komunikovat.)
```

P – I – 2

AttoSoft

Vašek si založil maličkou programátorskou firmu, kterou nazval příznačně AttoSoft.[1] Nedávno se mu konečně podařilo získat prvního klienta. Ten mu dal za úkol napsat N jednoduchých programů.

Vašek je však líný sám programovat, a proto si na tuto práci najal N programátorů. Když ráno všichni přišli do práce, ukázalo se, že každý z nich umí naprogramovat pouze jeden z potřebných programů. Naštěstí každý z požadovaných programů uměl někdo z nich naprogramovat, takže se mohli pustit do práce. Objevil se však další problém. Firma AttoSoft vlastní pouze jeden počítač a na něm může v jednom okamžiku pracovat jen jeden programátor.

Vašek si uvědomil, že je důležité zvolit správné pořadí, v němž budou jednotliví programátoři pracovat u počítače. Podle podepsané smlouvy totiž programátory neplatí za vykonanou práci. Každého programátora platí za čas, který uplyne od začátku práce na celé zakázce až do okamžiku, kdy tento programátor dokončí svůj program. Vašek o každém z programátorů ví, kolik mu musí zaplatit za jednu hodinu a jak dlouho mu napsání jeho programu bude trvat. Pomozte mu zjistit, v jakém pořadí má poslat programátory pracovat na počítači, aby jim dohromady mohl zaplatit co nejméně.

Příklad. Mějme tři programátory A, B, C. Programátor A chce 100 Kč za hodinu a na svůj program potřebuje 2 hodiny. B dostane 20 Kč za ho-

[1] Mikro je 10^{-6}, nano je 10^{-9}, piko je 10^{-12}, femto je 10^{-15}, atto je 10^{-18}.

dinu a potřebuje na práci 5 hodin času. Programátor C požaduje 500 Kč za hodinu a bude pracovat 20 hodin.

Pokud by programovali v pořadí A, B, C, musel by Vašek zaplatit programátorovi A za 2 hodiny, programátorovi B za 7 hodin a programátorovi C za 27 hodin, což by ho přišlo celkově na 13 840 Kč. Jestliže budou programovat v pořadí C, B, A, bude to Vaška stát jenom $500 \times 20 + 20 \times 25 + 100 \times 27 = 13\,200$ Kč, takže toto pořadí je výhodnější. (Ale není to ještě nejlepší možné řešení.)

Soutěžní úloha. Napište program, který přečte ze vstupu počet programátorů, jejich hodinové mzdy a časy potřebné na jejich práci a spočítá, v jakém pořadí má Vašek nechat programátory pracovat, aby jim dohromady zaplatil nejmenší možnou částku. Má-li úloha více různých řešení, program najde a vypíše jedno libovolné z nich.

Formát vstupu: První řádek vstupního souboru `attosoft.in` obsahuje jedno kladné celé číslo N ($1 \leqq N \leqq 10\,000$), které udává počet programátorů. Následuje dalších N řádků; i-tý z nich obsahuje dvě celá čísla m_i, t_i ($0 < m_i, t_i \leqq 30\,000$), kde m_i je hodinová mzda i-tého programátora a t_i je čas v hodinách, který i-tý programátor potřebuje na napsání svého programu.

Formát výstupu: Výstupní textový soubor `attosoft.out` obsahuje N řádků. Na j-tém z nich je zapsáno jedno celé číslo z rozmezí od 1 do N — číslo programátora, který bude programovat jako j-tý v pořadí.

Příklad:

`attosoft.in`	`attosoft.out`
`3`	`1`
`100 2`	`3`
`20 5`	`2`
`500 20`	*(Vašek zaplatí 11 740 Kč.)*

P – I – 3

Součty

Je dáno pole $A[1..N]$ celých čísel. Napište program, který bude umět co nejrychleji provádět následující příkazy:

- ▷ změň hodnotu $A[x]$ na y,
- ▷ vypiš součet prvků $A[x] + A[x+1] + \ldots + A[y]$.

Váš program si na začátku výpočtu může pole A v rozumném čase vhodně předzpracovat.

Formát vstupu: Vstupní textový soubor `soucty.in` obsahuje předem neznámý počet řádků. Na prvním řádku souboru je uvedeno je-

diné celé číslo N ($1 \leqq N \leqq 2\,000$). Druhý řádek obsahuje původní hodnoty uložené v poli A — celá čísla $a_1, a_2, \ldots, a_N$ oddělená mezerami ($0 \leqq a_i \leqq 1\,000\,000$).

Vstupní soubor pokračuje několika řádky s příkazy, z nichž každý má některý z následujících tvarů:

1 x y ($1 \leqq x \leqq N$, $0 \leqq y \leqq 1\,000\,000$)	změň hodnotu $A[x]$ na y
2 x y ($1 \leqq x \leqq y \leqq N$)	vypiš hodnotu $A[x] + \ldots + A[y]$

Vstup je ukončen řádkem obsahujícím jediné číslo 0.

Formát výstupu: Program musí vykonat všechny příkazy v pořadí, ve kterém jsou uvedeny na vstupu. Pro každý příkaz na vypsání součtu nějakých prvků pole musí do výstupního souboru `soucty.out` zapsat jeden řádek obsahující jedno celé číslo — součet příslušných prvků pole A v daném okamžiku.

Příklad:

```
soucty.in                    soucty.out
14                           26
1 4 3 1 1 3 1 1 3 1 4 1 1 1  8
2 1 14                       8
2 3 6                        5
1 2 0
2 3 6
1 3 0
2 3 6
0
```

P – I – 4

Registrový počítač

V této úloze se budeme zabývat *registrovými počítači*. *Registr* je něco podobného jako proměnná. V registru může být uloženo *libovolně velké* nezáporné celé číslo. Na rozdíl od proměnných, které mezi sebou můžeme sčítat, odčítat a násobit, s registrem lze provádět jen tři jednoduché operace: zvětšit jeho obsah o 1, zmenšit jeho obsah o 1 (pokud se pokusíme zmenšit obsah registru obsahujícího hodnotu 0, zůstane v něm 0) a otestovat registr, zda je v něm 0. Na začátku výpočtu jsou ve všech registrech nuly.

Registrový počítač může používat neomezený počet registrů označených R_0, R_1, R_2 atd. Vedle registrů má k dispozici ještě *konečně velkou* pomocnou paměť.

Program pro registrový počítač budeme zapisovat v jazyce velmi podobném programovacímu jazyku Pascal. Programovací jazyk registrového počítače bude oproti Pascalu rozšířen například o příkazy pro práci s registry, naopak některé příkazy z Pascalu v něm budou zakázány.

Registrový počítač bude řešit úlohy následujícího typu: počítač dostane na vstupu zadáno slovo (řetězec písmen) a po nějakém čase odpoví, zda je toto slovo *správné* nebo *špatné*. Aby nám mohl odpovědět, zavedeme do programovacího jazyka speciální příkazy *Accept* a *Reject*. Jakmile se během výpočtu vykoná příkaz *Accept*, vstupní slovo je správné a výpočet končí. Jestliže se provede příkaz *Reject*, slovo je špatné a výpočet končí. Pokud se výpočet zacyklí nebo pokud skončí, aniž by se provedl příkaz *Accept* nebo *Reject*, zadané vstupní slovo je rovněž špatné.

Příkaz „přičti 1 k obsahu registru R" budeme značit $Inc(R)$, „odečti 1 od obsahu registru R" budeme značit $Dec(R)$. Výraz $Zero(R)$ je pravdivý, jestliže je v registru R nula, v opačném případě je nepravdivý. Na začátku výpočtu jsou ve všech registrech nuly.

V každém programu můžeme použít jen konečně mnoho registrů. Kromě nich můžeme použít už jen konstantní počet pomocných proměnných typu *byte*[2] (nemůžeme tedy používat pole!) a jednu speciální proměnnou *vstup* typu *char*. Obsah proměnné *vstup* lze měnit pouze provedením příkazu *Read*(*vstup*). Jestliže počítač ještě nedočetl vstupní slovo, příkaz *Read*(*vstup*) z něj přečte jedno další písmeno a uloží ho do proměnné *vstup*. Pokud počítač již vstupní slovo dočetl, příkaz *Read*(*vstup*) uloží do proměnné *vstup* speciální znak $.

Jelikož registrový počítač má kromě registrů jen konečně mnoho paměti, nemůže si dovolit používat rekurzi (neměl by si kde pamatovat návratové adresy). My pro jistotu úplně zakážeme definovat a používat v programu procedury a funkce. Zakázáno je i volání všech standardních procedur a funkcí jazyka Pascal. V aritmetických výrazech lze používat pouze proměnné (tedy ne registry!), celočíselné konstanty, celočíselné operátory `+`, `-`, `*`, `div`, `mod` a závorky. V podmínkách se mohou používat výrazy $Zero(R_i)$, běžné relační operátory (`<`, `<=`, ...), logické spojky a závorky.

Z klíčových slov jazyka Pascal jsou tedy v programovacím jazyku registrového počítače povolena pouze následující: `var`, `begin`, `end`, `if`, `then`, `else`, `case`, `of`, `while`, `do`, `repeat`, `until`, `for`, `to`, `downto`, `div`, `mod`, `and`, `or`, `not` a `xor`.

[2] Taková proměnná obsahuje jedno celé číslo z rozmezí od 0 do 255.

Příklad 1. Napište program pro registrový počítač, který bude řešit následující úlohu. Na vstupu je zadán řetězec písmen a, b, c. Nechť α označuje počet těch písmen a ve vstupním řetězci, za kterými už není žádné c. Podobně nechť β je počet b, za nimiž není žádné c. Počítač má vstupní řetězec označit za správný právě tehdy, když $\alpha = \beta$.

Řešení. V registru R_1 si budeme pamatovat aktuální hodnotu α, v registru R_2 hodnotu β. Pokaždé, když přečteme ze vstupu písmeno c, oba registry vynulujeme. Na konci výpočtu jednoduše porovnáme hodnoty uložené v registrech. Rozmyslete si, že by stačilo použít jen jeden registr (v němž bychom měli hodnotu $\alpha - \beta$).

```
var vstup:char;
begin
   Read(vstup);
   while (vstup<>'$') do begin
      if (vstup='a') then Inc(R1);
      if (vstup='b') then Inc(R2);
      if (vstup='c') then begin
         while not Zero(R1) do Dec(R1);
         while not Zero(R2) do Dec(R2);
      end;
      Read(vstup);
   end;
   while not Zero(R1) do begin
      Dec(R1);
      if (Zero(R2)) then Reject;
      Dec(R2);
   end;
   if Zero(R2) then Accept;
end.
```

Příklad 2. Napište program pro registrový počítač, který bude řešit následující úlohu. Na vstupu bude zadán řetězec písmen a. Počítač ho označí za správný právě tehdy, když je jeho délka mocninou tří.

Řešení. Přečteme vstupní slovo, přičemž si do R_1 uložíme jeho délku. Potřebujeme zjistit, zda je to mocnina tří. Uloženou hodnotu proto budeme dělit třemi, dokud to půjde. Jestliže nakonec dostaneme podíl 0 a zbytek 1, původní číslo bylo mocninou tří, jinak nebylo.

```
var vstup:char;
    zbytek:byte;
```

```
begin
   Read(vstup);
   while (vstup<>'$') do begin Inc(R1); Read(vstup); end;
   if Zero(R1) then Reject;
   while true do begin
      zbytek:=0;
      while not Zero(R1) do begin
         Dec(R1);
         zbytek:=(zbytek+1) mod 3;
         if (zbytek=0) then Inc(R2);
      end;
      if (Zero(R2) and (zbytek=1)) then Accept;
      if (zbytek<>0) then Reject;
      while not Zero(R2) do begin Dec(R2); Inc(R1); end;
   end;
end.
```

Soutěžní úloha. Napište program pro registrový počítač, který bude řešit následující úlohu. Vstupem programu bude řetězec písmen `a`, `b`, `c`, `d`. Počítač ho označí jako správný právě tehdy, jestliže obsahuje nejvíce písmen `a` (tzn. počet písmen `a` obsažených ve vstupním slově je větší než počet písmen `b`, zároveň počet písmen `a` je větší než počet písmen `c` a zároveň také počet písmen `a` je větší než počet písmen `d`).

Například vstupní slovo `baacd` je tedy správné, zatímco vstupní slovo `baacdbb` je špatné (neboť obsahuje více písmen `b` než `a`) a ani vstup `ac` není správný (neboť písmen `a` a `c` je v něm stejný počet).

Základním kritériem hodnocení kvality navrženého programu bude počet registrů, které program používá. Pokuste se napsat program, kterému jich stačí co nejméně. Druhým kritériem pak bude doba výpočtu programu.

P – II – 1

Síť

Firma Truhlík a syn má ve městě N budov a chce všechny svoje budovy propojit počítačovou sítí. Vedení firmy rozhodlo, že pro K ($1 \leqq K \leqq N$) budov zakoupí vysokorychlostní připojení na Internet. Kromě toho mezi některými dvojicemi budov vybudují propojení optickým kabelem.

Dvě budovy se nacházejí v téže komponentě sítě, pokud lze mezi nimi komunikovat pomocí optických kabelů (buď mají přímé spojení, nebo

jsou spojeny nepřímo přes několik jiných budov). Aby bylo možné komunikovat mezi dvěma budovami ležícími v různých komponentách sítě, musí každá z těchto komponent obsahovat aspoň jeden počítač připojený na Internet.

Soutěžní úloha. Na vstupu jsou dána čísla N a K a pro každou dvojici budov jedno kladné celé číslo — cena za vybudování optického kabelu, který by propojil tuto dvojici budov. Navrhněte efektivní algoritmus, jenž určí, kterých K budov se má připojit na Internet a které dvojice budov se mají propojit optickým kabelem tak, aby mezi každými dvěma budovami bylo možné komunikovat a přitom aby celková cena vybudovaných optických kabelů byla co nejmenší.

Příklad:

Vstup:	*Výstup*:
$N = 4$, $K = 2$	Na Internet připojíme budovy 1 a 2,
Ceny spojení:	kabelem spojíme dvojice budov $(1, 3)$, $(2, 4)$ a $(3, 5)$.
(1,2): 100	Cena kabelů bude 47.
(1,3): 10	
(1,4): 100	
(1,5): 300	
(2,3): 100	
(2,4): 10	
(2,5): 300	
(3,4): 47	
(3,5): 27	
(4,5): 74	

P – II – 2

AttoSoft

Vaškova programátorská firma AttoSoft je známa z úlohy P–I–2. Vaškovi se nyní podařilo získat druhého klienta. Ten mu dal opět za úkol naprogramovat N jednoduchých programů.

Vašek chce tentokrát ušetřit ještě více, a proto místo programátorů zaměstnal N studentů, na každý program jednoho studenta. Firma AttoSoft vlastní stále jen jeden počítač a na něm může v každém okamžiku pracovat jen jeden student. Hlavní problém ale spočívá v tom, že studenti mohou pracovat jen ve volných chvílích mezi přednáškami.

Student i potřebuje p_i hodin času na napsání přiděleného programu, přijde do firmy v čase s_i a musí odejít nejpozději v čase t_i. Svůj program nemusí psát najednou, může občas práci přerušit a počítač uvolnit jiným studentům.

Soutěžní úloha. Napište program, jenž určí, kdy má počítač používat který student, aby všichni stihli napsat své programy za jeden den, nebo zjistí, že to není možné.

Příklad 1:

Vstup:	*Výstup*:
$N = 3$	Student 1 pracuje od 3 do 4.
$p_1 = 1$, $s_1 = 3$, $t_1 = 4$	Student 2 pracuje od 2 do 3 a od 4
$p_2 = 2$, $s_2 = 2$, $t_2 = 5$	do 5.
$p_3 = 5$, $s_3 = 1$, $t_3 = 10$	Student 3 pracuje od 5 do 10.

Příklad 2:

Vstup:	*Výstup*:
$N = 2$	Nelze.
$p_1 = 200$, $s_1 = 300$, $t_1 = 500$	
$p_2 = 200$, $s_2 = 400$, $t_2 = 600$	

P – II – 3

Bageta

Kleofáš dostal dnes ráno hlad a rozhodl se připravit si obloženou bagetu se sýrem. Bagetu si můžeme představit jako úsečku dlouhou N cm, nebo přesněji jako uzavřený interval $\langle 0, N \rangle$. Každý kousek sýra tvoří rovněž uzavřený interval celočíselné délky. Jelikož Kleofáš je pedant, pokládá na bagetu kousky sýra tak, aby souřadnice jejich začátků i konců byla celá čísla. Kleofáš by chtěl, aby bageta byla pokryta sýrem přesně podle jeho představ. Na to ale potřebuje jednoduchý počítačový program, který by mu pomohl.

Soutěžní úloha. Na vstupu je dána velikost bagety N (N je celé číslo, $1 \leqq N \leqq 10^6$). Následuje P příkazů ($1 \leqq P \leqq 10^9$), přičemž každý z nich má jeden z následujících možných tvarů:

`PRIDEJ` a b	Kleofáš přidal kousek sýra sahající od a do b.
`KOLIK` c	Program vypíše zprávu, kolik kusů sýra leží na pozici c.

Váš program musí zpracovávat příkazy v pořadí, v jakém jsou uvedeny na vstupu. Pro každý příkaz `KOLIK` vypište jedno číslo — počet dosud

položených kusů sýra, které leží nad souřadnicí c. (Jelikož kousky sýra jsou uzavřené intervaly, počítají se i ty kousky, pro něž je c souřadnice jejich začátku nebo konce.)

Příklad:

Vstup:	*Výstup*:
$N = 20$	
PRIDEJ 1 10	
PRIDEJ 6 12	
KOLIK 5	1
KOLIK 6	2
PRIDEJ 4 14	
KOLIK 5	2
KOLIK 16	0

P – II – 4

Registrový počítač

V této úloze se budeme zabývat tzv. *dvousměrnými* registrovými počítači. Od registrových počítačů z úlohy P–I–4 se liší tím, že se při čtení vstupního slova dovedou vracet zpět. Oproti původní definici registrových počítačů nyní proměnná *vstup* vždy obsahuje hodnotu aktuálního písmene ze vstupu. Na začátku výpočtu je aktuálním písmenem první písmeno zadané na vstupu. Polohu aktuálního písmene můžeme v programu změnit provedením příkazů *Left* (aktuálním se stane předcházející písmeno) a *Right* (aktuálním se stane následující písmeno). Pokud by se po provedení jednoho z těchto příkazů mělo aktuální písmeno nacházet mimo zadané vstupní slovo, proměnná *vstup* bude obsahovat speciální znak `$`. (Můžeme si představovat, že před i za vstupním slovem je zapsáno dostatečně mnoho znaků `$`.)

Příklad 1. Napište program pro dvousměrný registrový počítač, který bude řešit následující úlohu: Na vstupu bude zadán řetězec písmen `a`, `b`, `c`. Nechť α označuje počet písmen `a` ve vstupním řetězci, β nechť je počet `b` a γ počet `c`. Počítač má vstupní řetězec označit za správný právě tehdy, když $\alpha = \beta = \gamma$.

Řešení. Projdeme vstupní slovo zleva doprava a v registrech R_1 a R_2 si přitom spočítáme hodnoty α a β. Když vstupní slovo dočteme, porovnáme obsahy obou registrů. Vrátíme se na začátek, při druhém průchodu vstupním slovem spočítáme v R_1 a R_2 hodnoty β a γ a opět je porovnáme. Rozmyslete si, že by stačilo použít jediný registr, v němž bychom měli hodnotu $\alpha - \beta$, resp. $\beta - \gamma$.

```
var vstup: char;
begin
   while (vstup<>'$') do begin
      if (vstup='a') then Inc(R1);
      if (vstup='b') then Inc(R2);
      Right;
   end;
   while not Zero(R1) do begin
      Dec(R1); if Zero(R2) then Reject else Dec(R2);
   end;
   if not Zero(R2) then Reject;
   { bylo stejně 'a' a 'b', vrátíme se na začátek }
   Left; while (vstup<>'$') do Left;
   Right;

   while (vstup<>'$') do begin
      if (vstup='b') then Inc(R1);
      if (vstup='c') then Inc(R2);
      Right;
   end;
   while not Zero(R1) do begin
      Dec(R1); if Zero(R2) then Reject else Dec(R2);
   end;
   if Zero(R2) then Accept;
end.
```

Příklad 2. Napište program pro dvousměrný registrový počítač, který bude řešit následující úlohu: Na vstupu bude zadán řetězec písmen a. Počítač ho označí za správný právě tehdy, když je jeho délka mocninou tří.

Řešení. Řešení bude vypadat stejně jako na jednosměrném registrovém počítači. Projdeme vstupním slovem zleva doprava, přičemž si do registru R_1 uložíme délku tohoto slova. Potřebujeme zjistit, zda je to mocnina tří. Uloženou hodnotu proto budeme dělit třemi, dokud to půjde. Jestliže nakonec dostaneme podíl 0 a zbytek 1, původní číslo bylo mocninou tří, jinak nebylo.

```
var vstup: char;
    zbytek: byte;
begin
```

```
   while (vstup<>'$') do begin Inc(R1); Right; end;
   if Zero(R1) then Reject;
   while true do begin
      zbytek:=0;
      while not Zero(R1) do begin
         Dec(R1);
         zbytek:=(zbytek+1) mod 3;
         if (zbytek=0) then Inc(R2);
      end;
      if (Zero(R2) and (zbytek=1)) then Accept;
      if (zbytek<>0) then Reject;
      while not Zero(R2) do begin Dec(R2); Inc(R1); end;
   end;
end.
```

Soutěžní úloha. Napište program pro dvousměrný registrový počítač, který bude řešit následující úlohu:

Vstupem programu bude řetězec písmen a, b, c, d. Počítač ho označí jako správný právě tehdy, jestliže je to palindrom, tzn. je stejný při čtení zepředu i zezadu. Formálně řečeno: slovo $a_1a_2a_3\ldots a_{n-1}a_n$ je palindrom, jestliže $a_1 = a_n$, $a_2 = a_{n-1}$, $\ldots$, $a_{\lfloor n/2 \rfloor} = a_{\lceil n/2 \rceil}$. Tedy například vstupy `bacab` a `dd` jsou palindromy, zatímco vstupy `baacdbb` a `bacabdccc` nejsou.

P – III – 1

Agenti

Jistá nejmenovaná tajná společnost má N agentů. Z důvodu utajení může každý agent vydávat rozkazy jen několika dalším agentům. Agent, který dostane rozkaz, pošle tento rozkaz všem agentům, jimž může vydávat rozkazy. Šéfem společnosti je takový agent, který když vydá rozkaz, tak ho časem dostanou všichni agenti. (Společnost může mít i více šéfů, případně nemusí mít žádného šéfa.)

Soutěžní úloha. Na vstupu je dán počet agentů N. Agenti jsou označeni čísly od 7 (přesněji 007) do $N + 6$. Pro každého agenta je také dán seznam agentů, kterým může vydat rozkaz. Navrhněte efektivní algoritmus, který určí šéfa tajné společnosti (pokud jich existuje více, stačí nalézt jednoho libovolného z nich) nebo zjistí, že tajná společnost žádného šéfa nemá.

Příklad 1:

Vstup:
$N = 3$
Agent 7 rozkazuje agentovi 8.
Agent 8 rozkazuje agentovi 9.
Agent 9 rozkazuje agentovi 7.

Výstup:
Šéfem je agent 7.

Příklad 2:

Vstup:
$N = 4$
Agent 7 nerozkazuje nikomu.
Agent 8 nerozkazuje nikomu.
Agent 9 rozkazuje agentům 7 a 8.
Agent 10 rozkazuje agentům 7 a 8.

Výstup:
Žádný agent není šéfem.

P – III – 2

Teploty

Meteorologická stanice měří každou minutu teplotu vzduchu. Meteorologové by potřebovali program, který by jim v každém okamžiku sděloval, jaká nejnižší teplota byla naměřena během posledních K minut. Vaším úkolem bude napsat tento program.

Na vstupu je dáno číslo K, následuje posloupnost naměřených teplot ukončená hodnotou $-1\,000$. Váš program musí po přečtení každé teploty ihned vypsat nejnižší z posledních K načtených teplot (resp. nejnižší ze všech načtených teplot, jestliže jich dosud bylo méně než K).

Příklad:

Vstup:
$K = 3$
teploty:
9,0
4,7
5,3
2,1
9,0
9,8
17,0
9,5
-1000

Výstup:
9,0
4,7
4,7
2,1
2,1
2,1
9,0
9,5

P – III – 3

Registrový počítač

V této úloze se budeme zabývat tzv. *jednosměrnými registrovými počítači* — stejnými jako v úloze P–I–4 (tam je obsažena i jejich definice).

Soutěžní úloha. a) Nechť R je řetězec tvořený písmeny `a`, `b`, `c`, `A`, `B`, `C`. Označme $m(R)$ řetězec tvořený malými písmeny obsaženými v R (ve stejném pořadí, v jakém se vyskytují v R). Analogicky označme $v(R)$ řetězec tvořený velkými písmeny v R. Řetězec $upcase(R)$ dostaneme z R tak, že nahradíme všechna malá písmena odpovídajícími velkými písmeny.

Např. jestliže R = `aaAcB`, potom $m(R)$ = `aac`, $v(R)$ = `AB` a $upcase(R)$ = `AAACB`.

Napište program pro jednosměrný registrový počítač, který bude řešit následující úlohu: Na vstupu dostane řetězec R tvořený písmeny `a`, `b`, `c`, `A`, `B`, `C`. Počítač ho má označit za správný právě tehdy, když $upcase(m(R)) = v(R)$. (Vyjádřeno slovně: když velká písmena obsažená v R tvoří „stejné“ slovo jako malá.)

Například vstupní řetězce `aA`, `Aa` a `abAcBaCABb` jsou tedy správné, zatímco řetězce `aa`, `BcbC` a `acACa` jsou špatné.

Váš program může použit libovolný konečný počet registrů, hodnotí se jen jeho správnost. Pokud si myslíte, že takový program neexistuje, dokažte to.

b) Dokažte, že pro libovolnou úlohu platí: Jestliže umíme sestrojit program pro registrový počítač, který řeší danou úlohu pomocí tří registrů, potom dokážeme sestrojit také program, který tuto úlohu řeší pomocí dvou registrů.

Jinými slovy: Ukažte postup, kterým lze libovolný existující program používající tři registry přepsat na ekvivalentní program, jenž potřebuje pouze dva registry. Nezapomeňte zdůvodnit správnost svého postupu.

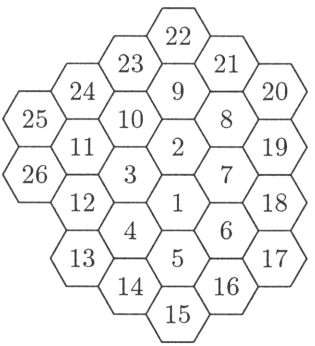
P – III – 4

Psíci

Program: `psici.pas` / `psici.c` / `psici.cpp`
Vstup: `psici.in`
Výstup: `psici.out`

„Tak, a teď budete skákat vždycky, když zapískám. A každý jinak!“ rozkázal Konrád svým dvěma psíkům. Chudáci malí, musí teď oba poskakovat po louce tak dlouho, dokud se oba současně neschovají.

Svět je nekonečná šestiúhelníková síť. Políčka tvořící svět jsou očíslována přirozenými čísly počínaje od 1 po spirále. Louku tvoří prvních N políček světa. Na obrázku je příklad louky pro $N = 26$:



Na louce stojí naši dva psíci na políčkách S_1, S_2. Skrýš pro prvního psíka je na políčku T_1, pro druhého na políčku T_2. Na M políčkách louky rostou bodláky a psíci tam za žádnou cenu neskočí.

Na jedno písknutí přeskočí každý psík na libovolné sousední políčko, pokud na něm nerostou bodláky. Oba psíci nemohou nikdy skočit současně stejným směrem a nemohou také oba dopadnout najednou na stejné políčko. Při každém písknutí musí každý z nich přeskočit na jiné políčko (i kdyby už stál ve své skrýši).

Vaším úkolem je zjistit, na kolik nejméně písknutí se mohou oba psíci dostat současně do svých skrýší.

Vstup: Ve vstupním souboru `psici.in` následuje po sobě popis několika (maximálně pěti) problémů. Každý problém má na svém prvním řádku dvě čísla N ($2 \leqq N \leqq 500$), M ($0 \leqq M \leqq N-2$), na druhém řádku čísla políček S_1, T_1, S_2, T_2 (v uvedeném pořadí, $1 \leqq S_1, T_1, S_2, T_2 \leqq N$). Následuje dalších M řádků s čísly políček, kde rostou bodláky. Vstupní soubor je ukončen řádkem obsahujícím dvě nuly ($M = N = 0$).

Můžete předpokládat, že vždy $S_1 \neq S_2$, $T_1 \neq T_2$ a že na políčkách S_1, S_2 nejsou bodláky. Na políčkách T_1 nebo T_2 bodláky být mohou — v takovém případě však úloha jistě nemá řešení.

Výstup: Do výstupního souboru `psici.out` zapište pro každý problém jeden řádek s nejmenším počtem písknutí, po němž mohou oba psíci

současně stát ve svých skrýších. Pokud není možné tohoto výsledného stavu dosáhnout, vypište do výstupního souboru místo počtu písknutí řádek se slovem „nelze“.

Příklad:

Vstupní soubor `psici.in`

```
11 0
3 10 6 7
11 1
3 10 6 7
1
10 2
3 10 7 8
2
9
0 0
```

Výstupní soubor `psici.out`

```
2
3
nelze
```

P – III – 5

AttoSoft

Program: `attosoft.pas` / `attosoft.c` / `attosoft.cpp`
Vstup: `attosoft.in`
Výstup: `attosoft.out`

Programátorské firmě AttoSoft se podařilo získat dalšího klienta, který potřebuje naprogramovat N programů. Vaškovi a jeho programátorům se však do práce moc nechtělo, a tak když přišel termín odevzdání, programy ještě stále nebyly hotové. Vašek se lekl a začal studovat smlouvu, kterou se zákazníkem podepsal.

Ve smlouvě byl pro každý program uveden vzorec, podle kterého se počítá pokuta za opožděné odevzdání programu v závislosti na délce zdržení. Naštěstí není třeba zaplatit součet pokut za všechny opožděné programy, ale jen nejvyšší pokutu ze všech. Vašek se proto nyní snaží naplánovat práci na programech tak, aby zaplatil co možná nejnižší pokutu. Stejně jako dříve má i nyní k dispozici jen jeden počítač, a proto není možné pracovat na více programech najednou. Započatou práci na programu není možné přerušit.[3]

[3] Šikovnější z vás si po přečtení zbytku zadání uvědomí, že i kdyby se to smělo, stejně by se to nevyplatilo.

Soutěžní úloha. Na vstupu je pro každý z nedokončených programů uveden vzorec na výpočet pokuty a údaj, kolik dní práce je zapotřebí na jeho dokončení. Napište program, který určí rozvrh na dokončení programů, při němž Vašek zaplatí nejmenší pokutu. Vzorec na výpočet pokuty má tvar polynomu nejvýše třetího stupně $ax^3 + bx^2 + cx + d$, ve kterém jsou koeficienty a, b, c, d celočíselné nezáporné a x je počet dní, o něž se odevzdání programu opozdilo.

Vstup: První řádek vstupního souboru `attosoft.in` obsahuje kladné celé číslo N ($1 \leqq N \leqq 5\,000$) — počet programů. Následuje N řádků, i-tý z nich obsahuje pět celých čísel l_i, a_i, b_i, c_i, d_i ($1 \leqq l_i \leqq 100$, $0 \leqq \leqq a_i, b_i, c_i, d_i \leqq 5\,000$) kde l_i je počet dní potřebný na dokončení i-tého programu a a_i, b_i, c_i, d_i jsou koeficienty vzorce na výpočet pokuty. Můžete předpokládat, že za 100 000 dní se stihnou napsat všechny programy.

Výstup: Výstupní soubor `attosoft.out` obsahuje N čísel oddělených mezerami nebo konci řádků. Tato čísla představují čísla jednotlivých programů v pořadí, v němž je třeba programy dokončit, aby byla pokuta nejmenší možná. Pokud má úloha více řešení, vypište jedno libovolné z nich.

Příklad:

Vstupní soubor `attosoft.in`	*Výstupní soubor* `attosoft.out`
3	1
10 1 0 0 0	3
3 0 0 0 10	2
1 0 0 5 0	

Zde pokuta za program číslo 1 dokončený po deseti dnech je $10^3 = \\ = 1\,000$, za program číslo 2 dokončený po 14 dnech je pokuta 10 a za program číslo 3 dokončený po 11 dnech je $5 \cdot 11 = 55$. Vašek tedy zaplatí pokutu 1 000.

P – I – 1

Na úvod pár slov pro ty, kdo dosud neměli příležitost seznámit se alespoň se základy *teorie grafů*. V našem chápání je graf tvořen několika body, které budeme nazývat *vrcholy* grafu, některé dvojice bodů jsou spojeny čárami, kterým budeme říkat *hrany* grafu. Formálněji řečeno, (neorientovaný) graf je dvojice $G = (V, E)$, kde V je množina vrcholů a $E \subseteq \{\{x, y\} : x, y \in V\}$ je množina neuspořádaných dvojic vrcholů, tj. hran. Právě takový graf máme v naší úloze zadán na vstupu — města v zemi představují vrcholy grafu a linky jsou hrany vedoucí mezi nimi.

Řekneme, že graf je *souvislý*, jestliže se dá po jeho hranách přejít z libovolného vrcholu do libovolného jiného. Podle zadání naší úlohy je graf zadaný na vstupu souvislý. Máme zjistit, zda odstranění některé hrany souvislost grafu poruší. Hranu s touto vlastností nazýváme *most*.

Jakým způsobem můžeme zjistit, zda je zkoumaný graf souvislý? Existuje na to více různých algoritmů. Nejčastějšímu algoritmu řešícímu tento problém se říká *obarvování vrcholů* nebo také *prohledávání* grafu. Základní myšlenka algoritmu je následující. Začneme v nějakém (libovolně zvoleném) vrcholu grafu a postupně obarvujeme všechny vrcholy, kam se dokážeme po hranách grafu dostat. Když už není možné obarvit žádný další vrchol, stačí se podívat, zda jsou obarveny všechny vrcholy grafu. Vrcholy je samozřejmě třeba obarvovat systematicky tak, abychom žádný z dostupných vrcholů nevynechali. Můžeme postupovat například prohledáváním do hloubky.

Prohledávání do hloubky je podobné postupu, jakým člověk zkoumá neznámé město. Začneme tím, že se postavíme do nějakého vrcholu a obarvíme ho. Nadále budeme barvit všechny vrcholy i hrany grafu, které navštívíme. Jestliže z vrcholu, kde právě jsme, vede nějaká ještě nepoužitá (tj. neobarvená) hrana, vydáme se po ní. Pokud přijdeme do dosud nenavštíveného (tj. neobarveného) vrcholu, obarvíme ho a rekurzívně zavoláme prohledávání z něj (tedy opět se snažíme najít nepoužitou hranu, atd.). Když přijdeme do již navštíveného, a tedy obarveného vrcholu, okamžitě se vrátíme po té hraně, kterou jsme do něj přišli. Jsme-li ve vrcholu, z něhož vedou samé obarvené hrany, vrátíme se zpět tou hranou, po které jsme do vrcholu přišli poprvé. Až se tímto způsobem budeme chtít vracet z vrcholu, kde jsme začínali, prohledávání končí.

Popsaným postupem projdeme právě dvakrát (tam a zpět) po každé z hran, k nimž se dokážeme dostat, a navštívíme všechny vrcholy, ke kterým lze dojít z počátečního vrcholu. Algoritmus je tedy korektní a jeho časová složitost je $O(M + N)$. Algoritmus je možné snadno rekurzívně implementovat, jak dokládá program uvedený na konci tohoto řešení.

Nejjednodušším řešením zadané úlohy by bylo postupně vyzkoušet odstranit každou jednotlivou hranu z grafu a vždy se podívat, zda je výsledný graf ještě stále souvislý. Takové řešení by mělo časovou složitost $O(M \cdot (M + N))$ — pro každou hranu potřebujeme spustit jedno prohledávání.

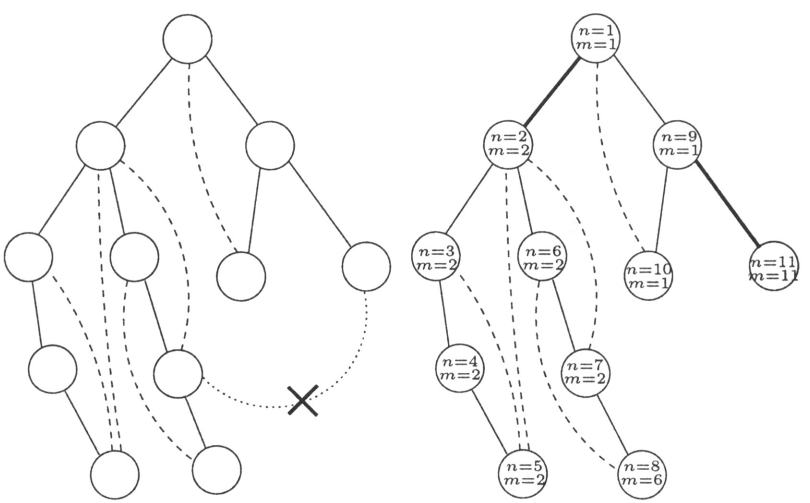
Ukážeme si však jiný algoritmus, který úlohu vyřeší v čase $O(M+N)$ (tedy s optimální časovou složitostí) a vyhledá přitom v grafu všechny mosty. Tento algoritmus je drobnou modifikací prohledávání do hloubky. Dříve než vysvětlíme samotné řešení, seznámíme se s několika potřebnými vlastnostmi prohledávání do hloubky. Začneme tedy s prohledáváním do hloubky v našem souvislém grafu. Všimněte si těch hran grafu, jimiž jsme během prohledávání přišli do dosud nenavštíveného vrcholu. Takových hran je přesně $N - 1$ (jedna pro každý vrchol grafu kromě toho, ve kterém jsme začínali s prohledáváním). Graf jimi tvořený je strom, neboť je souvislý a neobsahuje kružnice. Tento strom budeme nazývat *DFS strom* (DFS = depth-first search = prohledávání do hloubky). Vrchol, z něhož jsme graf začínali prohledávat, nazveme *kořenem* DFS stromu. Z každého jiného vrcholu x vede po *stromových* hranách (tj. po hranách DFS stromu) do kořene právě jedna cesta. Vrcholy ležící na této cestě budeme nazývat předky vrcholu x, zatímco o vrcholu x budeme říkat, že je jejich potomkem. Speciálně každý vrchol je sám sobě předkem i potomkem. Všichni potomci vrcholu x a stromové hrany vedoucí mezi nimi tvoří podstrom s kořenem x.

Ostatní hrany mohou být teoreticky dvou typů. Jestliže hrana spojuje vrchol s nějakým jeho předkem nebo potomkem, budeme ji nazývat *zpětná*, ostatní hrany nazveme *příčné*. Nechť uv je hrana, která není stromová. Všimněte si podstromů s kořeny u, v. Jsou dvě možnosti — pokud je jeden z nich podgrafem druhého, hrana uv je zpětná, jinak musí být tyto podstromy disjunktní a hrana uv je příčná. V DFS stromu však žádné příčné hrany nemohou být. To snadno zdůvodníme sporem. Nechť uv je příčná hrana. Bez újmy na obecnosti můžeme předpokládat, že během prohledávání jsme do u přišli dříve než do v. Všimněte si nyní okamžiku, kdy se při prohledávání chceme vrátit z vrcholu u zpět. Je-li uv příčná hrana, nesměli jsme dosud vrchol v navštívit (jinak by v byl

potomkem u a hrana uv by byla zpětná). Vrchol v je tedy dosud nenavštívený soused vrcholu u, proto bychom se z u ještě neměli vracet zpět, ale měli bychom se vydat do v, což je spor.

Všechny hrany grafu tedy můžeme rozdělit na stromové a zpětné. Je zřejmé, že leží-li hrana na nějaké *kružnici* (cyklu), po jejím odstranění graf zůstane souvislý. Každá zpětná hrana uv leží na kružnici tvořené hranou uv a cestou z u do v po hranách DFS stromu. Mosty se proto mohou nacházet jen mezi stromovými hranami. Každý most rozděluje graf na dvě části, přičemž v jedné z nich se nachází kořen DFS stromu.

Představte si, že náš graf zavěsíme za kořen. Nyní se vydáme z kořene dolů po stromových hranách. Uvažujme jednu konkrétní stromovou hranu uv, kde u je vrchol ležící blíže ke kořeni než v. Kdy je hrana uv mostem? Tehdy, když ji nedokážeme obejít. Jinými slovy řečeno když se z podstromu s kořenem v nemůžeme dostat do vrcholu u (nebo ekvivalentně: do u nebo libovolného jeho předka) bez použití hrany uv.



Budeme tedy chtít pro každou hranu uv určit, zda existuje cesta z v do u nebo do nějakého jeho předka, která nepoužívá hranu uv. Hledejme takovou cestu, která používá nejmenší počet zpětných hran a ze všech takových cest je nejkratší. Co o ní umíme říci? Její poslední hrana bude určitě zpětná, neboť po stromových hranách se do vrcholu u či nad u nedostaneme. Všechny její vrcholy kromě posledního budou ležet v podstromu s kořenem v, protože jakmile se dostaneme nad u, skončíme. Do všech vrcholů ležících v podstromu s kořenem v se ale jistě můžeme do-

stat z v stromovými hranami. Ukázali jsme tedy, že pokud nějaká hledaná cesta existuje, pak existuje i taková, při níž jdeme nejprve několika stromovými hranami a potom jednou zpětnou hranou. Stačí nám proto pro každou stromovou hranu v grafu ověřit, zda existuje takováto cesta. Jak to uděláme?

Během prohledávání budeme číslovat vrcholy v pořadí, v jakém do nich budeme poprvé vstupovat. Číslo vrcholu x označíme $num(x)$. Je zřejmé, že všechny vrcholy ležící v podstromu s kořenem u mají číslo větší než $num(u)$. Na druhé straně všichni předci vrcholu u mají číslo menší než $num(u)$. Kdybychom pro v znali nejmenší číslo vrcholu, do kterého se můžeme dostat bez použití hrany uv (což musí být předek vrcholu v, neboť příčné hrany neexistují), měli bychom vyhráno — hrana uv je mostem právě tehdy, když je toto číslo větší než $num(u)$. Ukázali jsme si ale, že nám stačí uvažovat cesty, které vedou nejprve několika stromovými hranami „dolů" a potom jednou zpětnou hranou „nahoru". Budeme si tedy pro každý vrchol přímo během prohledávání počítat nejmenší číslo vrcholu, do kterého se z něj dokážeme dostat takovouto cestou.

Tím máme algoritmus řešení úlohy téměř hotov, zbývá už jen celý postup shrnout. Budeme prohledávat zkoumaný graf do hloubky a zároveň si pro každý vrchol x budeme počítat dvě čísla: $num(x)$ (kolikátý navštívený vrchol to je) a $up(x) = \min\{num(y)$:do y vede z x cesta výše uvedeného tvaru$\}$. Jak vypočítat $num(x)$ je zřejmé. Hodnota $up(x)$ je rovna minimu z $num(x)$, ze všech hodnot $up(x_i)$ pro syny vrcholu x a ze všech hodnot $num(y_i)$ vrcholů, do nichž vede z x zpětná hrana. Hodnotu $up(x)$ tedy umíme spočítat v okamžiku, kdy se při prohledávání vracíme z vrcholu x. V tomto okamžiku dokážeme také rozhodnout o hraně vedoucí z vrcholu x do jeho otce y, zda je mostem — stačí porovnat hodnoty $up(x)$ a $num(y)$ (resp. $up(x)$ a $num(x)$).

```
program Sit;
var G : array[1..100,1..100] of integer; {graf}
 deg,num,up: array[1..100] of integer; {stupně vrcholů a obě čísla pro ně}
 visited : array[1..100] of boolean;    {byl jsem už v tomto vrcholu?}
 N,M,C : integer;                {počet vrcholů, hran, navštívených vrcholů}
 ok : boolean;

procedure Load;
var i,x,y : integer;
begin
   read(N,M); fillchar(deg,sizeof(deg),0);
   for i:=1 to M do begin
      read(x,y);
      inc(deg[x]); G[x][deg[x]]:=y;
      inc(deg[y]); G[y][deg[y]]:=x;
```

```
   end;
end;

procedure DFS(v,parent : integer);
var i : integer;
begin
   visited[v]:=true;
   num[v]:=C; up[v]:=C; inc(C); { nastavíme obě čísla ve vrcholu }
   for i:=1 to deg[v] do if not visited[G[v][i]] then begin
      DFS(G[v][i],v);
      if up[G[v][i]]<up[v] then up[v]:=up[G[v][i]];
   end else begin { zpětná hrana }
      if G[v][i]<>parent then
        if num[G[v][i]]<up[v] then up[v]:=num[G[v][i]];
   end;
   if num[v]=up[v] then ok:=false; { hrana v-parent je most }
end;

begin
   Load;
   fillchar(visited,sizeof(visited),0); C:=1; ok:=true;
   DFS(1,1);
   if ok then writeln('ANO') else writeln('NE');
end.
```

P – I – 2

Uvažujme libovolné pořadí, v němž budou programátoři pracovat, a podívejme se na dva po sobě napsané programy — nechť jsou to programy i a j. Napsání programu budeme nadále označovat jako událost. První z našich událostí, tedy i, začne v čase T_0, bude trvat po dobu t_i a Vašek za ni proto zaplatí částku $(T_0 + t_i) \cdot m_i$. Druhá událost, j, začne v čase $T_0 + t_i$ (tzn. ihned po skončení události i) a bude stát $(T_0 + t_i + t_j) \cdot m_j$ — každého programátora platíme nejen za dobu, kdy pracuje, ale od úplného začátku.

Po sečtení zjistíme, že když se obě uvažované události vykonají v pořadí i, j, Vašek za ně bude muset zaplatit částku $S_{i,j} = T_0 \cdot (m_i + m_j) + + t_i \cdot m_i + (t_i + t_j) \cdot m_j$.

Co by se stalo, kdybychom zaměnili pořadí událostí i a j? Podobně jako v předchozím případě můžeme spočítat, kolik bude muset Vašek zaplatit za tyto dvě události. Za první z nich (tedy j) to bude $(T_0 + t_j) \cdot m_j$ a za druhou $(T_0 + t_j + t_i) \cdot m_i$, což dohromady činí $S_{j,i} = T_0 \cdot (m_i + m_j) + + t_j \cdot m_j + (t_j + t_i) \cdot m_i$.

Porovnejme nyní tyto dva výsledky. Označme si pro jednoduchost jejich společnou část $A := T_0 \cdot (m_i + m_j) + t_i \cdot m_i + t_j \cdot m_j$. Po snadných

úpravách dostáváme:

$$S_{i,j} = A + m_i \cdot m_j \cdot \frac{t_i}{m_i}, \qquad S_{j,i} = A + m_i \cdot m_j \cdot \frac{t_j}{m_j}.$$

Zajímá nás, která z těchto hodnot je menší, ale to je zjevně ta, která má menší poměr t_k/m_k. To tedy znamená, že pokud $t_i/m_i > t_j/m_j$, výměnou pořadí těchto událostí dosáhneme nižší výsledné částky. (Je zřejmé, že záměna pořadí dvou po sobě následujících událostí neovlivní částku, kterou zaplatíme ostatním programátorům.)

Z uvedených úvah vyplývá, že pokud v nějakém pořadí událostí najdeme dvě po sobě jdoucí takové, že první z nich má větší poměr t_k/m_k než druhá, jejich vzájemnou výměnou získáme nové pořadí událostí, které je levnější. Optimální pořadí událostí bude proto takové, v němž jsou poměry t_k/m_k uspořádány od nejmenšího po největší.

Samotný program je potom už velmi jednoduchý — stačí události utřídit vzestupně podle poměru t_k/m_k, což dokážeme provést v průměrném čase $O(n \cdot \log n)$ například algoritmem QuickSort.

```
program AttoSoft;
type Tprg = record
            m,t,idx: integer;
            tm: real;
            end;

var N,i: integer;
    prg: array[1..10000] of Tprg;
    sum,t: integer;

procedure QSort(l,r: integer);
var y: Tprg;
    x: real;
    i,j: integer;
begin
    i:=l; j:=r; x:=prg[(l+r) div 2].tm;
    repeat
    while prg[i].tm<x do inc(i);
    while x<prg[j].tm do dec(j);
    if i<=j then
    begin
        y:=prg[i]; prg[i]:=prg[j]; prg[j]:=y;
        inc(i); dec(j);
    end;
    until i>j;

    if l<j then QSort(l, j);
    if i<r then QSort(i, r);
end;
```

```
begin
    assign(input,'attosoft.in'); reset(input);
    assign(output,'attosoft.out'); rewrite(output);

    read(N);
    for i:=1 to N do
    begin
      read(prg[i].m, prg[i].t);
      prg[i].idx:=i;
      prg[i].tm:=prg[i].t/prg[i].m;
    end;

    QSort(1,N);

    for i:=1 to N do writeln(prg[i].idx);

    {Výsledná částka:
    sum:=0; T:=0;
    for i:=1 to N do
    begin
      sum:=sum+(T+prg[i].t)*prg[i].m;
      T:=T+prg[i].t;
    end;
    writeln('Výsledná částka: ',sum);
    }

    close(input); close(output);
end.
```

P – I – 3

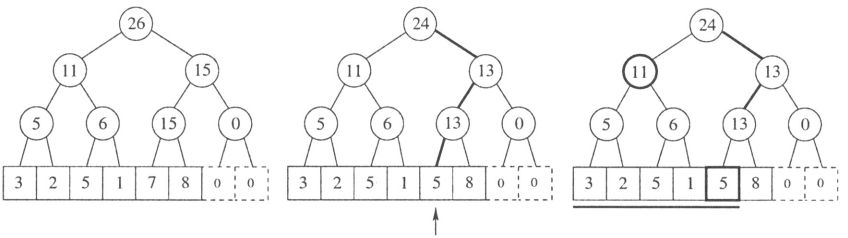
Pro zjednodušení dalších úvah zvětšíme nejprve pole A tak, aby jeho velikost byla rovna nejbližší vyšší mocnině dvou. Tím se pole A prodlouží maximálně na dvojnásobek původní délky, takže tato úprava neovlivní časovou složitost výsledného algoritmu. Nadále tedy předpokládejme, že prodloužené pole má délku $N = 2^K$.

Představme si, že nad polem A vybudujeme úplný binární strom. Jeho listy budou odpovídat jednotlivým prvkům pole A, každý vyšší vrchol tohoto stromu odpovídá nějakému intervalu v poli A (přesněji řečeno odpovídá prvkům pole určeným listy z jeho podstromu). V každém vrcholu stromu si budeme pamatovat součet čísel v příslušném intervalu pole. Tuto datovou strukturu budeme nazývat intervalový strom.

V nejspodnější vrstvě našeho stromu se nachází N vrcholů, v předcházející vyšší vrstvě jich je $N/2$, ve třetí odspodu $N/4$, atd. V celém stromě je tedy $2N - 1$ vrcholů, proto budeme potřebovat na jeho uložení paměť velikosti $\Theta(N)$ (čti: lineární). V průběhu předzpracovaní pole A musíme tuto paměť naplnit, proto na předzpracování bude zapotřebí čas $\Omega(N)$

(čti: aspoň lineární). Snadno zjistíme, že v lineárním čase dokážeme náš strom skutečně vytvořit — stačí ho zaplňovat po vrstvách zdola nahoru.

Co se stane s naším stromem, když změníme hodnotu prvku $A[j]$? Musíme změnit zapamatované hodnoty pro všechny intervaly, v nichž je změněný prvek pole obsažen. Ty ale odpovídají právě vrcholům intervalového stromu ležícím na cestě z j-tého listu do kořene. Je jich tedy $K + 1 = O(\log N)$. Změnit hodnotu v poli A tudíž dokážeme v logaritmickém čase.



Zbývá ukázat, jak lze pomocí intervalového stromu odpovídat na otázky ze zadání. Řešme nejprve jednodušší úlohu: Jakou hodnotu má součet $S(x) = A[1] + \ldots + A[x]$? Začneme v kořeni našeho stromu. Mohou nastat dvě možnosti: Jestliže interval od 1 do x leží celý v levém podstromu, zavoláme rekurzívní výpočet pro levého syna. Pokud ne, tak tento interval zabírá celý levý podstrom a ještě část pravého. Vezmeme proto součet všech prvků pole odpovídajících levému podstromu (ten máme spočítaný v levém synovi) a zavoláme rekurzívní výpočet pro pravého syna a zbytek intervalu.

Takto postupně v našem stromu procházíme dolů po cestě od kořene do x-tého listu, přitom na každé úrovni vykonáme jen konstantní počet operací. Proto pro libovolné x dokážeme hodnotu $S(x)$ spočítat v čase $O(\log N)$. To je ale vše, co potřebujeme vědět, neboť $A[x] + \ldots + A[y] = S(y) - S(x-1)$ (dodefinujeme $S(0) = 0$).

Pomocí intervalového stromu tedy dokážeme každý příkaz ze zadání úlohy zpracovat v logaritmickém čase. Naše řešení potřebuje lineární paměť a lineární čas na předzpracování.

Nejjednodušší implementací intervalového stromu je uložit ho v jednom poli podobně jako haldu. Kořen stromu bude umístěn v poli na pozici 1, synové vrcholu x jsou na pozicích $2x$ a $2x+1$. Prvky původního pole A odpovídají listům stromu a začínají v poli na pozici N. V praxi se někdy paměťová složitost snižuje na polovinu tím, že si ukládáme jen součty v levých synech, implementace je potom ale o něco náročnější.

```
program Soucty;

var T : array[1..10000] of longint; { strom }
    oldN,N,prikaz,i : longint;
    x,y : longint;
    pom : longint;

function Soucet(delka, koren, interval : longint) : longint;
{delka - délka intervalu, jehož součet počítáme
 koren - kořen podstromu, ve kterém počítáme
 interval - délka intervalu odpovídajícího kořenu
            (abychom ji nemuseli počítat)}

begin
   if delka=0 then begin Soucet:=0; exit; end;
   if interval=1 then begin Soucet:=T[koren]; exit; end;
   if delka<=(interval div 2)
      then Soucet:=Soucet(delka,2*koren,interval div 2)
      else Soucet:=T[2*koren]+
         Soucet(delka-(interval div 2),2*koren+1,interval div 2);
end;

begin
   fillchar(T,sizeof(T),0);
   read(oldN);
   N:=1; while N<oldN do N:=N*2; { upravíme velikost pole }
   for i:=1 to oldN do read(T[N+i-1]);
   for i:=N-1 downto 1 do T[i]:=T[2*i]+T[2*i+1];
   read(prikaz);
   while prikaz>0 do begin
      if prikaz=1 then begin
         { měníme hodnotu }
         read(x,y); i:=x+N-1; pom:=y-T[i];
         while i>=1 do begin Inc(T[i],pom); i:=i div 2; end;
      end else begin
         { počítáme součet }
         read(x,y);
         writeln(Soucet(y,1,N)-Soucet(x-1,1,N));
      end;
      read(prikaz);
   end;
end.
```

P – I – 4

Nejjednodušším řešením je použít čtyři registry a v každém si počítat počet písmen jednoho typu. Když dočteme slovo, v R_0 máme počet přečtených písmen a, v R_1 počet b, atd. Nyní budeme najednou zmenšovat hodnoty ve všech čtyřech registrech. *Accept* zavoláme právě tehdy, když registr R_0 zůstane nejdéle nenulový.

Počet použitých registrů lze snadno snížit na tři: Nechť jsme dosud přečetli α písmen a, β písmen b, γ písmen c a δ písmen d. V registrech si budeme ukládat absolutní hodnoty výrazů $\alpha - \beta$, $\alpha - \gamma$, $\alpha - \delta$, ve třech proměnných si budeme pamatovat jejich znaménka (např. 0, pokud je v příslušném registru nula, 1, pokud tam je kladné číslo, a 255, když je záporné.) V každém okamžiku výpočtu pak dokážeme snadno určit, zda bylo dosud na vstupu písmen a nejvíce — to platí právě tehdy, když jsou všechny tři zapamatované hodnoty kladné (tzn. všechna tři jejich znaménka rovna 1).

Naše řešení bude potřebovat jen dva registry. Je možné ukázat (v tomto vzorovém řešení to ale neuděláme), že jeden registr na vyřešení této úlohy nestačí. Naše řešení bude tudíž vzhledem k počtu registrů optimální.

V průběhu výpočtu si v R_0 budeme pamatovat číslo $2^\alpha 3^\beta 5^\gamma 7^\delta$, registr R_1 budeme používat pouze na pomocné výpočty. Když například přečteme ze vstupu jako další písmeno b, pomocí registru R_1 vynásobíme obsah registru R_0 třemi. Po dočtení vstupu potřebujeme porovnat hodnoty α, β, γ a δ. Podobně jako v prvním řešení je budeme najednou zmenšovat (což v tomto případě znamená dělit obsah R_0 vhodným číslem) a akceptujeme právě tehdy, když nám na konci zůstane kladná mocnina 2.

Samotný program je sice trochu delší, ale je jen přímočarou implementací uvedené myšlenky.

```
var c:char;
    d,e,f:byte;

begin
 { čteme vstup a kódujeme do R0, kolik v něm čeho je }
 Inc(R0);
 Read(c);
 while c<>'$' do begin
    case c of
       'a': d:=2;
       'b': d:=3;
       'c': d:=5;
       'd': d:=7;
       end;
    while not Zero(R0) do begin   { R1 := R0 * d, R0 := 0 }
       Dec(R0);
       for e:=1 to d do Inc(R1);
 end;
    while not Zero(R1) do begin   { R0 := R1, R1 := 0 }
       Dec(R1);
       Inc(R0);
   end;
```

```
    Read(c);
    end;

{ v každé iteraci z R0 odebereme jedno "a" }
{ a po jednom z dosud zbývajících ostatních písmen }
while true do begin
 e := 0;                    { e := R0 mod 210, R1 := R0 div 210, R0 := 0 }
 while not Zero(R0) do begin   { (210 = 2*3*5*7) }
    Dec(R0);
    e := (e+1) mod 210;
 if e=0 then Inc(R1);
    end;
 d := 1;                    { zjistíme, čím vším bylo R0 ještě dělitelné }
                            { ale stačí, když budeme testovat e místo R0 }
 if e mod 2 = 0 then d := d*2;
 if e mod 3 = 0 then d := d*3;
 if e mod 5 = 0 then d := d*5;
 if e mod 7 = 0 then d := d*7;
 if d=2 then Accept;           { už zbývají jen a-čka, což je dobré }
 if e mod 2 <> 0 then Reject;
                          { a-čka došla, ale zbyla jiná písmena => špatné }
 while not Zero(R1) do begin { V R0 má být původní R0 div d, což získáme}
    Dec(R1);              { tak, že nejprve spočteme (210 div d) * R1 ... }
    for f := 1 to 210 div d do Inc(R0);
 end;
 for f := 1 to e div d do Inc(R0);
                            { ... a pak přičteme e div d; R1 máme nulové }
 end;
end.
```

P – II – 1

Zadanou úlohu si převedeme do řeči teorie grafů. Budovy firmy představují *vrcholy* našeho grafu, *hrany* grafu odpovídají možným propojením optickým kabelem. Úlohu vyřešíme nejprve pro případ $K = 1$. V tomto případě je naším úkolem vybrat takovou množinu hran, aby všechny vrcholy byly navzájem propojeny (ne nutně přímo). Taková množina hran se nazývá *kostra grafu*, a jelikož chceme, aby součet cen hran v kostře byl co nejmenší, řešíme problém hledání *minimální kostry*.

Rozmyslete si, že minimální kostra grafu neobsahuje žádný cyklus — kdyby totiž nějaký obsahovala, mohli bychom jeho libovolnou hranu odstranit. Na druhé straně když do minimální kostry přidáme libovolnou hranu, vznikne nám cyklus, neboť vrcholy, mezi nimiž vede přidaná hrana, byly už spojeny pomocí nějakých hran kostry.

Hledání minimální kostry (Primův algoritmus). Algoritmus je založen na následující myšlence. Vrcholy grafu rozdělíme na dvě skupiny: na připojené a nepřipojené. Na začátku algoritmu zvolíme libovolný vrchol

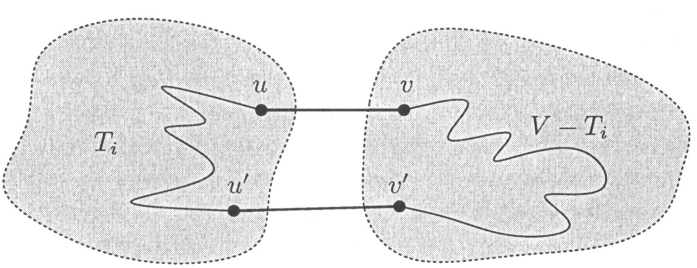
a prohlásíme ho za připojený, ostatní vrcholy jsou zatím nepřipojené. V každém kroku algoritmu připojíme jeden vrchol k dosud vytvořené síti následujícím způsobem. Najdeme nejkratší hranu spojující připojený a nepřipojený vrchol. Tuto hranu přidáme do sítě a její druhý konec se stane připojeným vrcholem. Skončíme ve chvíli, když jsou všechny vrcholy připojeny.

Aby byl algoritmus efektivní, potřebujeme umět rychle nalézt nejkratší hranu spojující připojený a nepřipojený vrchol. To zařídíme tak, že pro každý dosud nepřipojený vrchol si budeme pamatovat, ze kterého připojeného vrcholu k němu vede nejkratší hrana. Pokaždé, když přidáme k připojeným vrcholům další vrchol, musíme si informaci o nejbližších připojených vrcholech aktualizovat. Projdeme všechny nepřipojené vrcholy a pokud je nově připojovaný vrchol bližší, naši informaci změníme.

Skutečnost, že výsledná množina hran tvoří kostru, je zřejmá. Je však třeba dokázat, že je tato kostra minimální. Představme si libovolnou minimální kostru (dále ji budeme označovat MK) a porovnávejme ji s výsledkem našeho algoritmu (dále VNA).

Jestliže MK a VNA jsou shodné, VNA je minimální kostra. Předpokládejme tedy, že MK a VNA nejsou shodné. Nechť T_i je množina připojených vrcholů po i-tém kroku našeho algoritmu. Seřadíme hrany ve VNA podle toho, jak jsme je přidávali, a najdeme první hranu, která se vyskytuje ve VNA, ale není obsažena v MK. Nechť tato hrana byla přidána v kroku $i+1$ a nechť spojuje vrchol $u \in T_i$ a vrchol $v \notin T_i$.

Přidejme hranu (u, v) do MK. Tím vznikne v MK cyklus, který začíná v T_i, přejde po hraně (u, v) ven z T_i a potom se vrátí nějakou cestou zpět do T_i (obr. 41). Na této cestě musí existovat aspoň jedna hrana (u', v'),



Obr. 41. Přidáním hrany (u, v) vznikne v MK cyklus, který začíná v T_i, přejde po hraně (u, v) ven z T_i a potom se vrátí nějakou cestou zpět do T_i.

která má jeden konec v T_i a druhý konec mimo T_i. Cena této hrany musí být aspoň taková, jako je cena hrany (u, v). V opačném případě by si

náš algoritmus v kroku $i+1$ musel vybrat hranu (u', v') namísto hrany (u, v). Proto pokud hranu (u', v') odebereme z MK a přidáme tam místo ní hranu (u, v), cena MK se nezvýší. Nemůže se však ani snížit, neboť MK je minimální. Upravená MK bude tedy nadále minimální kostrou v grafu. Navíc VNA a MK se nyní shodují v prvních $i+1$ hranách. Stejným způsobem postupně přeměníme MK na VNA, přičemž nezvýšíme její cenu, takže VNA musí být také minimální kostrou.

Řešení pro obecné K. Dosud jsme předpokládali $K = 1$. Jestliže $K > 1$, nemusíme hranami pospojovat všechny vrcholy. Ke komunikaci totiž můžeme využít také Internet. Stačí, když se naše síť bude skládat z K souvislých částí, v každé z těchto souvislých částí vybereme jeden vrchol, který připojíme na Internet, a tak bude moci komunikovat každý vrchol s každým.

Takovouto síť můžeme získat například odebráním $K-1$ nejdražších hran z minimální kostry MK. Tím se nám totiž MK rozpadne právě na K souvislých částí. Jediným problémem je ukázat, že toto řešení je skutečně nejlevnější možné.

Označme tedy symbolem P množinu $K-1$ nejdražších hran kostry MK. Jejich odebráním z MK dostaneme množinu hran Q, která se skládá z K souvislých částí. Nechť existuje levnější množina hran T, která rovněž tvoří síť složenou z K souvislých částí.

Budeme uvažovat graf tvořený kostrou MK a hranami z množiny T. Jelikož už MK je souvislá, tento graf je jistě souvislý. Proto lze zvolit několik hran z MK, jimiž se dají jednotlivé komponenty T pospojovat. Každá přidaná hrana spojí dvě komponenty do jedné větší, takže stačí přidat $K-1$ hran. Množinu těchto přidaných hran označíme symbolem S.

Všimněte si následujících dvou skutečností:

- Množina hran S určitě není dražší než P, neboť obě obsahují $K-1$ hran z kostry MK, ale P jsme vybrali tak, aby obsahovala nejdražší hrany.
- Podle našeho předpokladu množina hran T je levnější než množina hran Q.

Z toho ale vyplývá, že kostra $S \cup T$ je levnější než MK $= P \cup Q$, což je spor s tím, že MK je minimální kostra. Tím jsme ukázali, že k vyřešení úlohy stačí z MK odebrat $K-1$ nejdražších hran.

Časová složitost. Při hledání minimální kostry se v každém kroku přidá jeden vrchol do množiny připojených, vykoná se tedy celkem $N-1$ kroků. V každém kroku nejprve v čase $O(N)$ najdeme nejkratší hranu

spojující připojený a nepřipojený vrchol. Potom aktualizujeme informaci o nejbližším připojeném vrcholu pro všechny dosud nepřipojené vrcholy. Tato aktualizace představuje opět provedení $O(N)$ operací. Celková časová složitost je proto kvadratická, tj. $O(N^2)$.

Z výsledné minimální kostry potom potřebujeme odebrat $K-1$ nejdražších hran. To můžeme udělat tak, že hrany kostry setřídíme. Třídění lze provést v čase $O(N \log N)$, v tomto případě nám ovšem stačí použít jednoduché třídění pracující v čase $O(N^2)$. Celková časová složitost je $O(N^2)$.

```
program Sit_P_II_1;
const
   maxn = 1000;
   nekonecno = 10000;

var
   N,K: integer; { počet budov, počet komponent }
   a: array[1..maxn,1..maxn] of integer; { ceny spojení }
   sit: array[1..maxn,1..2] of integer; { seznam hran výsledné sítě }

procedure nacti_vstup;
var
   i,j: integer;
begin
   write('Počet budov N:'); readln(N);
   write('Počet internetových připojení K:'); readln(K);
   for i:=1 to N do
      for j:=i+1 to N do begin
         write('Cena (',i,',',j,'):'); readln(a[i,j]);
         a[j,i]:=a[i,j];
      end;
end; {nacti_vstup}

procedure minimalni_kostra;
{najde minimální kostru a uloží její hrany do pole sit}
var
   pripojene: array[1..maxn] of boolean;
   nej: array[1..maxn] of integer;
   i,j: integer;
   min, nejlepsi: integer;

begin
   {na začátku je jenom vrchol 1 připojený}
   pripojene[1]:=true;
   for i:=2 to N do begin
      {v poli nej si budeme pro každý dosud nepřipojený vrchol
       udržovat nejbližší připojený vrchol}
      pripojene[i]:=false;
      nej[i]:=1;
   end;

   for i:=1 to N-1 do begin
```

```
                        {najdeme nejkratší hranu, která spojuje
                         připojený a nepřipojený vrchol}
      min:=nekonecno;
      for j:=1 to N do begin
         if not pripojene[j] then begin
            if a[j,nej[j]]<min then begin
               nejlepsi:=j; min:=a[j,nej[j]]
            end;
         end;
      end;

      {nalezený vrchol připojíme}
      pripojene[nejlepsi]:=true;
      {spojení (nejlepsi,nej[nejlepsi]) patří do sítě}
      sit[i,1]:=nejlepsi; sit[i,2]:=nej[nejlepsi];

      {přepočítáme pole nej: pro každý vrchol zjistíme, zda právě
      připojený vrchol nezkrátí jeho vzdálenost k připojeným vrcholům}
      for j:=1 to N do begin
         if not pripojene[j] then begin
            if a[j,nej[j]]>a[j,nejlepsi] then nej[j]:=nejlepsi;
         end;
      end;
   end;
end; {minimalni_kostra}

procedure utrid_hrany_site;
{setřídí hrany sítě od nejlevnější po nejdražší}
var
   i,j,k,min: integer;
begin
   for i:=1 to N-1 do begin
      min:=i;
      for j:=i+1 to N-1 do begin
         if a[sit[j,1],sit[j,2]]<a[sit[min,1],sit[min,2]] then
            min:=j;
      end;
      k:=sit[min,1]; sit[min,1]:=sit[i,1]; sit[i,1]:=k;
      k:=sit[min,2]; sit[min,2]:=sit[i,2]; sit[i,2]:=k;
   end;
end; {utrid_hrany_site}

procedure vypis_vysledek;
{vypíše výsledné hrany sítě}
var
   i: integer;
begin
   writeln('Je třeba vybudovat spojení mezi následujícími budovami:');
   for i:=1 to N-K do begin
      writeln('(',sit[i,1],',',sit[i,2],')');
   end;
end; {vypis_vysledek}

begin
   nacti_vstup;
```

```
    minimalni_kostra;
    utrid_hrany_site;
    vypis_vysledek;
end.
```

P – II – 2

Uvažujme, který ze studentů má pracovat u počítače v daném okamžiku t. Zřejmě to musí být jeden z těch studentů, kteří už přišli do firmy, ale ještě nedokončili svůj program. O těchto studentech řekneme, že jsou *aktivní* v čase t. Dokážeme, že v optimálním řešení vždy můžeme poslat pracovat na počítači toho z aktivních studentů, který musí odejít nejdříve (nechť je to student a). Kdyby totiž existoval rozvrh, ve kterém v čase t pracuje nějaký jiný student b, pak se student a musí dostat k počítači ještě v čase t' někdy mezi t a časem odchodu t_a. Student b však odchází v čase $t_b \geqq t_a$. Proto můžeme sestrojit nový rozvrh, v němž necháme studenta a chvíli pracovat na počítači v čase t a stejně dlouhou dobu potom necháme studenta b pracovat na počítači v čase t'. Jestliže byl původní rozvrh správný, je správný i takto modifikovaný rozvrh, neboť každý pracuje stejně dlouho jako v původním rozvrhu a každý pracuje před svým odchodem.

Dokázali jsme, že v každém okamžiku může pracovat ten z aktivních studentů, který musí nejdříve odejít. Kdy tedy může dojít ke změně obsazení počítače? Buď tehdy, když přijde nový student a potřebuje odejít dříve, než student právě pracující (v tom případě se vystřídají u počítače), nebo když nějaký student dokončí svůj program a uvolní počítač.

Náš algoritmus bude sestrojovat rozvrh obsazení počítače postupně od začátku do konce. Studenty seřadíme podle času jejich příchodu a u každého si zaznamenáme, jak dlouho ještě potřebuje pracovat. Budeme si také udržovat množinu aktivních studentů. V každém kroku algoritmu najdeme nejbližší událost, jež může ovlivnit rozvrh. Touto událostí je buď příchod studenta nebo ukončení práce právě pracujícího studenta. V obou případech zaktualizujeme datové struktury a potom najdeme aktivního studenta s nejbližším odchodem a přidělíme mu počítač. Pokud tento student již nemá dost času na dokončení svého programu, oznámíme, že všechny programy nelze dokončit. Správnost tohoto tvrzení přímo vyplývá z důkazu uvedeného výše.

Seřazení studentů je možné provést v čase $O(N \log N)$. Počet událostí je $2N$, neboť každý student jednou přijde a jednou dokončí program. Při každé události potřebujeme najít aktivního studenta s nejmenším časem

odchodu. Kdybychom kvůli tomu vždy procházeli všechny aktivní studenty, dostali bychom algoritmus s časovou složitostí $O(N^2)$. Algoritmus se však dá zefektivnit, jestliže uložíme aktivní studenty do haldy uspořádané podle času jejich odchodu. V takovém případě zpracování jedné události trvá jen $O(\log N)$ — když někdo přišel, vložíme ho do haldy, když někdo dokončil práci, z haldy ho odstraníme. Poté se podíváme na minimum v haldě — studenta, který bude od této chvíle pracovat u počítače. Celková časová složitost algoritmu s použitím haldy je tedy pouze $O(N \log N)$.

```
program AttoSoft_P_II_2;
type student =
   record
      prichod, odchod, zbyva: integer;
   end;

const Nekonecno = 10000;

var A: array [1..1000] of student;       { pole studentů }
    N: integer;                          { počet studentů }
    Halda: array [1..1000] of integer;   { halda podle času odchodu }
    Halda_N: integer;                    { počet studentů v haldě }

procedure trid;
begin
   { vynechána kvůli úspoře místa }
end; { trid }

procedure vloz_do_haldy(student: integer);
var i, rodic, tmp:  integer;
begin
   { vlož studenta na konec haldy a posouvej ho nahoru }
   Halda_N := Halda_N + 1;
   Halda[Halda_N] := student;
   i := Halda_N;
   while i>1 do begin
      rodic := i div 2;
      if A[Halda[i]].odchod < A[Halda[rodic]].odchod then begin
         tmp := Halda[i];
         Halda[i] := Halda[rodic];
         Halda[rodic] := tmp;
      end;
      i := rodic;
   end;
end; { vloz_do_haldy }

procedure vyber_z_haldy;
var i, dite, tmp:  integer;
begin
   { první prvek nahraď posledním a posouvej ho dolů }
   Halda[1] := Halda[Halda_N];
   Halda_N := Halda_N - 1;
```

```
   i := 1;
   while i*2<=Halda_N do begin
      dite := i*2;
      if (i*2+1<=Halda_N)
         and (A[Halda[i*2+1]].odchod < A[Halda[dite]].odchod) then begin
            dite := i*2+1;
      end;
      if A[Halda[i]].odchod > A[Halda[dite]].odchod then begin
         tmp := Halda[i];
         Halda[i] := Halda[dite];
         Halda[dite] := tmp;
      end;
      i := dite;
   end;
end; { vyber_z_haldy }

procedure nacti;
var i: integer;
begin
   readln(N);
   for i := 1 to N do begin
      readln(A[i].prichod, A[i].odchod, A[i].zbyva);
   end;
end; { nacti }

function min(a,b: integer): integer;
begin
   if a<b then min := a
   else min := b;
end; { min }

var Pracuje, Skonci: integer;       { kdo právě pracuje a kdy skončí }
    Stary_cas, Novy_cas: integer;   { aktuální a předcházející událost }
    i: integer;                     { index do pole A }

begin
   nacti;                           { načti studenty do pole A }
   A[N+1].prichod := Nekonecno;     { zarážka }
   trid;                    { seřaď prvky pole A podle položky "prichod" }
   Pracuje := -1;                   { nikdo nepracuje }
   Skonci := Nekonecno;
   Halda_N := 0;                    { inicializace haldy }
   i := 1;
   Stary_cas := 0;
   while (i <= N) or (i = N+1) and (Pracuje > 0) do begin
      { Najdi novou událost }
      Novy_cas := min(A[i].prichod, Skonci);
      writeln('Čas ',Novy_cas);
      { Pokud někdo pracoval u počítače, vyhoď ho }
      if Pracuje > 0 then begin
         writeln(Novy_cas, ': student ', Pracuje, ' od počítače.');
         A[Pracuje].zbyva := A[Pracuje].zbyva - (Novy_cas - Stary_cas);
         if A[Pracuje].zbyva = 0 then vyber_z_haldy;
      end;
```

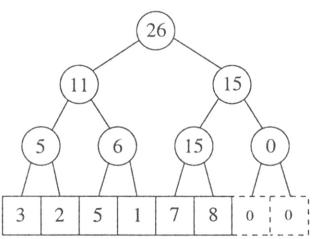
```
      { Jestliže událostí je příchod, }
      { vlož do haldy a jdi na další příchod }
      if (i <= N) and (A[i].prichod < Skonci) then begin
         vloz_do_haldy(i);
         i := i+1;
      end;

      { Najdi nového studenta k počítači }
      if Halda_N > 0 then begin
         Pracuje := Halda[1];
         Skonci := Novy_cas + A[Pracuje].zbyva;
         if Skonci > A[Pracuje].odchod then begin
            writeln('Rozvrh neexistuje!');
            exit;
         end;
         writeln(Novy_cas, ': student ', Pracuje, ' k počítači.');
      end
      else begin
         Pracuje := -1;
         Skonci := Nekonecno;
      end;
      Stary_cas := Novy_cas;
   end;
end.
```

P – II – 3

Na příkaz KOLIK c musí náš program odpovídat, v kolika dosud zadaných intervalech c leží. Na úvod uvedeme dvě triviální řešení. První je založeno na tom, že si budeme jednoduše pamatovat všechny dosud zadané intervaly a při každém příkazu KOLIK je všechny projdeme. Paměťová složitost tohoto řešení je $O(P)$, časová v nejhorším případě až $O(P^2)$. (Příkaz PRIDEJ dokážeme zpracovat v čase $O(1)$, ale na KOLIK potřebujeme v nejhorším případě až $O(P)$.) Trochu lepší řešení využívá pomocné pole velikosti $N+1$, v němž si pro každou celočíselnou pozici budeme pamatovat počet intervalů, které ji obsahují. Jeden interval přidáme v čase $O(N)$, na otázku odpovíme v čase $O(1)$. Výsledná časová složitost tohoto algoritmu je $O(N \cdot P)$, paměťová $O(N)$.

Uvědomte si, co vlastně potřebujeme zjistit, když nám přijde příkaz KOLIK c. Potřebujeme určit S — počet intervalů, které začínají na pozici $\leqq c$ a končí na pozici $\geqq c$. Nechť $Z(x)$ je počet intervalů, které začínají na pozici $\leqq x$, a $K(x)$ je počet intervalů, které končí na pozici $\leqq x$. Potom $S = Z(c) - K(c-1)$. (Intervaly, které končí před c, jsou započteny v $Z(c)$ i v $K(c-1)$, a proto je do S nezapočítáváme.) Stačilo by nám tedy umět rychle zjišťovat hodnoty $Z(x)$ a $K(x)$.

V řešení úlohy budeme využívat myšlenku z úlohy P–I–3 — datovou strukturu, kterou jsme nazvali *intervalový strom*.[4] Připomeňme si, o co šlo: Představte si, že nad polem A (jehož délku N jsme zvětšili na nejbližší mocninu dvou) vybudujeme úplný binární strom. Jeho listy budou odpovídat jednotlivým prvkům pole A, každý vyšší vrchol tohoto stromu bude odpovídat nějakému intervalu v poli A (přesněji bude odpovídat prvkům určeným listy z jeho podstromu). V každém vrcholu stromu si budeme pamatovat součet čísel v příslušném intervalu pole. Změnit hodnotu v poli A (a příslušně upravit součty ve vrcholech stromu) dokážeme v čase $O(\log N)$, zjistit součet libovolného intervalu v poli A dokážeme rovněž v čase $O(\log N)$.



$Z(c)$ je vlastně součet počtů intervalů začínajících na pozicích $0, 1, 2, \ldots, c$. Budeme mít pole, ve kterém si tyto počty budeme pamatovat, a nad ním vybudovaný *intervalový strom*. Každé přidání intervalu změní jednu hodnotu v poli, tuto změnu dokážeme uskutečnit v čase $O(\log N)$. Analogicky budeme používat druhé pole (a druhý *intervalový strom*) pro počty intervalů, které na jednotlivých pozicích končí. Pomocí těchto datových struktur dokážeme každou hodnotu Z a K spočítat v čase $O(\log N)$.

Detailnější popis obou operací s *intervalovým stromem* a jeho implementaci v poli najdete v řešeních P–I–3. Časová složitost našeho vzorového řešení je $O(P \log N)$ a paměťová $O(N)$. Všimněte si, že by nám stačilo udržovat jedno pole. Přidání intervalu $\langle a, b \rangle$ by znamenalo např. zvýšení hodnoty na pozici a a snížení hodnoty na pozici $b + 1$.

```
program Bageta_P_II_3;
var ZZ,KK: array[0..2100000] of longint; { stromy pro Z a K }
    N,oldN,a,b,c,kde: longint;
    prikaz,pom: char;

function
Soucet(var T: array of longint; delka, koren, interval: longint): longint;
```

[4] Neplést si s intervaly ze zadání!

```
{T - pole, v němž počítáme součty (všimněte si: "var T", ne "T" - proč?)
 delka - délka intervalu, jehož součet počítáme
 koren - kořen podstromu, ve kterém ho počítáme
 interval - délka intervalu odpovídajícího kořenu
 (abychom ji nemuseli počítat)}
begin
   if (delka=0) then begin Soucet:=0; exit; end;
   if (interval=1) then begin Soucet:=T[koren]; exit; end;
   if (delka<=(interval div 2))
      then Soucet:=Soucet(T,delka,2*koren,interval div 2)
      else Soucet:=T[2*koren]+Soucet(T,delka-(interval div 2),
                  2*koren+1,interval div 2);
end;

begin
 fillchar(ZZ,sizeof(ZZ),0);
 fillchar(KK,sizeof(KK),0);
 readln(oldN); N:=1; while (N<oldN+1) do N:=N*2; {upravíme velikost pole}

 while not eof do begin
   read(prikaz); pom:=prikaz; while (pom<>' ') do read(pom);
   if (prikaz='P') then begin
      readln(a,b);
      kde:=a+N; while (kde>=1) do begin Inc(ZZ[kde]); kde:=kde div 2; end;
      kde:=b+N; while (kde>=1) do begin Inc(KK[kde]); kde:=kde div 2; end;
   end else begin
      readln(c);
      writeln(Soucet(ZZ,c+1,1,N) - Soucet(KK,c,1,N));
   end;
 end;
end.
```

P – II – 4

Představte si, že bychom kromě registrů měli k dispozici ještě jeden *zásobník*[5]. Potom bychom již úlohu dokázali snadno vyřešit: Procházíme vstupním slovem zleva doprava a přečtená písmena vkládáme do zásobníku. Až potom budeme ze zásobníku písmena odebírat, budeme je dostávat v opačném pořadí, než v jakém byla do zásobníku vložena. Vrátíme se proto na začátek slova a budeme porovnávat, zda je slovo stejné odpředu jako odzadu. Vždy přečteme jedno písmeno ze vstupu, vyzvedneme jedno písmeno ze zásobníku a porovnáme je. Skončíme, když někdy dostaneme dvě různá písmena (slovo je špatné) nebo když dočteme celé vstupní slovo (slovo je správné).

Kdybychom tedy měli k dispozici zásobník, máme úlohu vyřešenou. Zásobník si však dokážeme simulovat v jednom registru (s pomocí dru-

[5] Zásobník je datová struktura, která podporuje operace „vlož prvek“ a „odeber naposledy vložený prvek“.

hého)! Jak na to? Písmena `a`, `b`, `c`, `d` budou odpovídat číslicím 1, 2, 3, 4. Číslo uložené v registru R_1 bude představovat obsah zásobníku — když ho zapíšeme v poziční soustavě o základu 5, jednotlivé cifry budou představovat hodnoty vložené do zásobníku (cifra na místě jednotek bude naposledy vložená hodnota). Například když do prázdného zásobníku vložíme postupně písmena `a`, `c`, `b`, `a`, bude v R_1 hodnota $a\times5^3+c\times5^2+b\times5+a = 1\times5^3+3\times5^2+2\times5+1 = 125+75+10+1 = 211$.

Jak ale s takovýmto registrem-zásobníkem pracovat? Vložit novou hodnotu x je jednoduché — pomocí registru R_2 vynásobíme obsah R_1 pěti a potom ho x-krát zvětšíme o 1. Vyzvednout naposledy vloženou hodnotu také není těžké — je to přesně opačná operace. Vydělíme obsah registru R_1 pěti, zbytek po dělení je naposledy vložená hodnota, podíl (který dostaneme v R_2) je nový obsah zásobníku bez této hodnoty.

Máme tedy funkční řešení úlohy, které potřebuje dva registry. Pokusíme se však nalézt řešení ještě lepší. Jen s jedním registrem se nám už nepodaří simulovat zásobník a musíme proto vymyslet něco jiného.

Nejprve trochu terminologie: aktuální písmeno se bude v našem řešení pohybovat sem a tam po vstupním slově. Kvůli názornosti místo „aktuální je i-té písmeno vstupního slova“, resp. „přesuneme aktuální písmeno doleva/doprava“ budeme říkat „stojíme na pozici i“, resp. „jdeme doleva/doprava“. Délku vstupního slova budeme značit n.

Představte si, že stojíme na pozici i (přičemž ale i si nijak nepamatujeme, v R_1 je nula). Chtěli bychom písmeno na této pozici porovnat s jemu odpovídajícím písmenem na pozici $n+1-i$. Náš program ovšem nezná n ani i. Jak na to? Písmeno na naší pozici si zapamatujeme v proměnné. Nyní si zjistíme i. Jdeme doleva, dokud nepřijdeme na začátek vstupního slova, a zvyšujeme R_1. Odpovídající písmeno je i-té od konce. Není tedy těžké dojít k němu — přejdeme na konec slova, potom zmenšujeme R_1 a jdeme doleva, dokud v R_1 není nula. Písmena porovnáme, a jsou-li různá, končíme. Jinak se potřebujeme vrátit zpět na pozici, kde jsme začínali. K tomu použijeme úplně stejný postup: Cestou doprava spočítáme v registru R_1 potřebný počet kroků, přesuneme se na začátek slova a vykonáme stejný počet kroků směrem doprava. Tím jsme se dostali do stejné situace, v níž jsme začínali, jen máme porovnané aktuální písmeno s jemu odpovídajícím písmenem. Celý tento postup budeme nazývat *porovnání*.

Chtěli bychom postupně porovnat všechny navzájem si příslušející dvojice písmen. To ale není problém. Začínáme na prvním písmenu vstupu a provedeme *porovnání*. Pokud není první písmeno stejné jako poslední, skončili jsme, jinak pokračujeme. Přesuneme se doprava (na druhé

písmeno) a vykonáme další *porovnání*. Takto pokračujeme tak dlouho, dokud neporovnáme n-té písmeno s prvním (a nezjistíme, že právě porovnané písmeno bylo již posledním písmenem vstupního slova).

```
var vstup: char;
    pismeno: byte;
begin
   while (vstup<>'$') do begin
         { v R1 máme nulu, začínáme porovnání }
      if (vstup='a') then pismeno:=1;
      if (vstup='b') then pismeno:=2;
      if (vstup='c') then pismeno:=3;
      if (vstup='d') then pismeno:=4;
         { spočítáme, kde jsme }
      while (vstup<>'$') do begin Inc(R1); Left; end;
         { přejdeme na pravý konec }
      Right;
      while (vstup<>'$') do Right;
         { přejdeme na odpovídající pozici }
      while not Zero(R1) do begin Dec(R1); Left; end;
         { kontrola }
      if (pismeno=1) and (vstup<>'a') then Reject;
      if (pismeno=2) and (vstup<>'b') then Reject;
      if (pismeno=3) and (vstup<>'c') then Reject;
      if (pismeno=4) and (vstup<>'d') then Reject;
         { návrat zpět }
      while (vstup<>'$') do begin Inc(R1); Right; end;
      Left;
      while (vstup<>'$') do Left;
      while not Zero(R1) do begin Dec(R1); Right; end;
         { posun na další písmeno, které je třeba zkontrolovat }
      Right;
   end;
   Accept; { všechno správně }
end.
```

P – III – 1

Úlohu si můžeme reprezentovat pomocí orientovaného grafu. Agenti představují vrcholy grafu. Skutečnost, že agent a může vydat rozkaz agentovi b, vyjádříme orientovanou hranou (a, b). Naším úkolem je nalézt v tomto grafu vrchol, z něhož se můžeme dostat do všech ostatních vrcholů.

Začneme prohledáváním grafu do hloubky z libovolného zvoleného vrcholu. Jestliže při tomto prohledávání navštívíme všechny vrcholy, našli jsme šéfa — je jím vrchol, kterým jsme prohledávání začali. V opačném případě pokračujeme tak, že zahájíme nové prohledávání do hloubky v jednom z vrcholů, které jsme dosud nenavštívili (dříve navštívené vrcholy grafu přitom necháme označené jako navštívené). To opakujeme

tak dlouho, dokud nenavštívíme všechny vrcholy našeho grafu. Nechť r je vrchol, v němž jsme zahájili poslední prohledávání.

Tvrzení. *Má-li náš graf aspoň jednoho šéfa, potom vrchol r je šéfem.*

Důkaz. Předpokládejme, že náš graf má šéfa a že vrchol r není šéfem. Nechť je šéfem vrchol s. Musíme uvažovat dvě možnosti:

- ▷ *Vrchol s byl navštíven při posledním prohledávání.* To by ale znamenalo, že se do tohoto vrcholu můžeme dostat z vrcholu r (neboť vrchol r je počátkem posledního prohledávání), tudíž se můžeme dostat z vrcholu r do libovolného jiného vrcholu grafu přes s, což je však v rozporu s naším předpokladem, že r není šéfem.
- ▷ *Vrchol s byl navštíven dříve než při posledním prohledávání.* Jelikož se však z vrcholu s dá dojít do libovolného vrcholu, museli bychom také vrchol r navštívit ve stejném prohledávání jako s, takže vrchol r nemůže být začátkem posledního prohledávání.

Zbývá tedy už jen ověřit (opět prohledáváním do hloubky), zda r je skutečně šéfem grafu; v opačném případě graf nemá žádného šéfa. Časová složitost celého algoritmu je $O(M + N)$, kde N je počet vrcholů a M je počet hran grafu.

```
program Agenti;

const MAXM = 10000;
      MAXN = 100;

var rozkazuje: array [1..MAXM] of integer;
    ind_od, ind_do: array [1..MAXN] of integer;
    {agent i+6 rozkazuje agentům rozkazuje[ind_od[i]]...
               rozkazuje[ind_do[i]]}
    N: integer;
    navstiven: array [1..MAXN] of boolean;

procedure Nacti;
var i,M,agent: integer;
begin
   write('Počet agentů:'); readln(N);

   M:=0;
   for i:=1 to N do begin
      write('Agent ',i+6,' rozkazuje: (ukonči -1)');
      ind_od[i]:=M+1;
      read(agent);
      while (agent>0) do begin
           M:=M+1;
           rozkazuje[M]:=agent-6;
           read(agent);
      end;
      ind_do[i]:=M;
```

```
   end;
end; {procedure Nacti}

procedure Prohledej(i: integer);
var j: integer;
begin
   if not navstiven[i] then begin
      navstiven[i]:=true;
      for j:=ind_od[i] to ind_do[i] do begin
         Prohledej(rozkazuje[j]);
      end;
   end;
end; {procedure Prohledej}

var i,posledni: integer;
    je_sef: boolean;

begin
   Nacti;
   for i:=1 to N do navstiven[i]:=false;
   for i:=1 to N do begin
      if not navstiven[i] then begin
         posledni:=i;
         Prohledej(i);
      end;
   end;

   for i:=1 to N do navstiven[i]:=false;
   Prohledej(posledni);

   je_sef:=true;
   for i:=1 to N do je_sef:=je_sef and navstiven[i];

   if je_sef then
      writeln('Šéfem je agent ', posledni+6)
   else
      writeln('Žádný agent není šéfem');
end. {program Agenti}
```

P – III – 2

Snadno sestrojíme řešení, které potřebuje čas $O(K)$ na zpracování jedné hodnoty ze vstupu. Stačí si v cyklicky přepisovaném poli pamatovat posledních K vstupních hodnot. Pokaždé, když přečteme další číslo ze vstupu, pole jednoduše projdeme a vypíšeme nejmenší z hodnot uložených v poli.

Vzorové řešení vystačí s časem $O(\log K)$ na zpracování jednoho čísla. Představme si, že bychom si aktuálních K hodnot udržovali v haldě. Novou hodnotu do této haldy lehce přidáme v čase $O(\log K)$. Předtím, než vypíšeme minimum (které je uloženo v kořeni haldy), potřebujeme

však ještě z haldy odstranit nejstarší hodnotu. Jak ale máme vědět, která z nich to je?

Pomůžeme si tím, že hodnoty, které nám budou přicházet, vložíme nejen do haldy, ale také do fronty. Mezi těmito dvěma datovými strukturami si budeme udržovat vzájemné odkazy, abychom v každém okamžiku dokázali o každém prvku fronty říci, kde je v haldě, a naopak.

Když tedy přijde nová hodnota, vložíme ji do haldy a na konec fronty. Následně ze začátku fronty odstraníme nejstarší hodnotu, pomocí odkazu ji najdeme v haldě a odstraníme ji také odtamtud. Nyní už jen vypíšeme hodnotu uloženou v kořeni haldy.

Obě operace s haldou mají časovou složitost $O(\log K)$, zbývající operace dokážeme provést v konstantním čase. Paměť spotřebovaná haldou i frontou je $O(K)$.

```
#include <stdio.h>
#define MAXK 100047
#define INFTY 1e10
#define SWAP(x,y) pom=(x); (x)=(y); (y)=pom

typedef struct { double val; int ptr; } tZaznam;
int K;                    // ze zadání
tZaznam H[MAXK],Q[MAXK]; // halda a fronta
int qs;                   // začátek fronty
tZaznam pom;

void init(void) { // naplníme haldu i frontu "nekonečně" velkými hodnotami
  int i;
  qs=0; H[0].val=-10000;
  for (i=0;i<K;i++)
  { Q[i].val=H[i+1].val=INFTY; Q[i].ptr=i+1; H[i+1].ptr=i; }
}
void bubbleup(int idx) { // bublej prvkem nahoru v haldě
  int next=idx,p1,p2;
  if (H[idx/2].val > H[idx].val) next=idx/2;
  if (next!=idx) {
    p1=H[idx].ptr; p2=H[next].ptr;
    SWAP(H[idx],H[next]);
    Q[p1].ptr=next; Q[p2].ptr=idx;
    bubbleup(next);
  }
}
void bubbledown(int idx) { // bublej prvkem dolů v haldě
  int next=idx,p1,p2;
  if (2*idx<=K)   if (H[2*idx  ].val < H[next].val) next=2*idx;
  if (2*idx+1<=K) if (H[2*idx+1].val < H[next].val) next=2*idx+1;
  if (next!=idx) {
    p1=H[idx].ptr; p2=H[next].ptr;
    SWAP(H[idx],H[next]);
    Q[p1].ptr=next; Q[p2].ptr=idx;
    bubbledown(next);
```

```
  }
}

int main(void) {
 int delptr;
 double x;
 scanf("%d ",&K); init();
 while (1) {
   scanf("%lf ",&x); if (x==-1000) break;
   delptr=Q[qs].ptr; // najdeme hodnotu, kterou je třeba vymazat z haldy
   H[delptr].val=H[K].val; H[delptr].ptr=H[K].ptr; Q[H[K].ptr].ptr=delptr;
   K--; bubbledown(delptr); bubbleup(delptr); K++;
                                                  // smažeme a upravíme haldu
   Q[qs].val=H[K].val=x; Q[qs].ptr=K; H[K].ptr=qs; bubbleup(K); // vložíme
   qs=(qs+1)%K;
   printf("%g\n",H[1].val);
 }
 return 0;
}
```

P – III – 3

a) Na úvod si připomeňme, že v řešení krajského kola jste se mohli kromě jiného dočíst, jak lze pomocí dvou počítadel simulovat zásobník. Pro jistotu si zopakujeme, jak na to:

Zásobník si můžeme simulovat v jednom registru (s pomocí druhého). Písmena a, b, c budou odpovídat číslům 1, 2, 3. Číslo uložené v registru R_1 bude představovat náš zásobník — když ho zapíšeme v poziční soustavě o základu 4, jednotlivé cifry budou představovat vložené hodnoty (cifra na místě jednotek bude naposledy vložená hodnota). Například když do prázdného zásobníku vložíme postupně písmena a, c, b, a, bude v R_1 hodnota $\mathtt{a} \times 4^3 + \mathtt{c} \times 4^2 + \mathtt{b} \times 4 + \mathtt{a} = 1 \times 4^3 + 3 \times 4^2 + 2 \times 4 + 1 = 64 + 48 + 8 + 1 = 121$.

Jak ale s takovýmto registrem-zásobníkem pracovat? Vložit novou hodnotu x je jednoduché — pomocí registru R_2 vynásobíme obsah R_1 čtyřmi a potom ho x-krát zvětšíme o 1. Rovněž odebrání naposledy vložené hodnoty není těžké — je to přesně opačná operace. Vydělíme obsah registru R_1 čtyřmi. Zbytek po dělení je naposledy vložená hodnota, podíl (který dostaneme v R_2) je obsah zásobníku bez této hodnoty.

Nyní můžeme již přikročit k řešení zadané úlohy. Jednou možností je simulovat (pomocí dvou zásobníků) frontu, v níž si udržujeme ta písmena, jejichž pár jsme ještě neviděli. Toto řešení je poměrně komplikované a jeho základní myšlenka spočívá v tom, že přicházející písmena vkládáme do prvního zásobníku, písmena na kontrolu vybíráme z druhého zásobníku

a vždy, když se nám druhý zásobník vyprázdni, do něj přesypeme obsah prvního zásobníku.

Ukážeme si raději jednodušší řešení. Budeme opět používat dva zásobníky. Do prvního budeme vkládat všechna přicházející malá písmena (jako hodnoty 1, 2, 3), do druhého velká (také jako hodnoty 1, 2, 3). Po dočtení vstupního slova jednoduše porovnáme obsahy obou zásobníků. Vstupní slovo bylo správné právě tehdy, je-li jejich obsah stejný. To již snadno ověříme.

```
var vstup: char;
    i,co: byte;
begin
   Read(vstup);
   while vstup<>'$' do begin
      if vstup>='a' then begin
         if vstup='a' then co:=1;
         if vstup='b' then co:=2;
         if vstup='c' then co:=3;
         while not Zero(R1) do
            begin Dec(R1); for i:=1 to 4 do Inc(R0); end;
         while not Zero(R0) do begin Dec(R0); Inc(R1); end;
         while co>0 do begin Inc(R1); co:=co-1; end;
      end else begin
         if vstup='A' then co:=1;
         if vstup='B' then co:=2;
         if vstup='C' then co:=3;
         while not Zero(R2) do
            begin Dec(R2); for i:=1 to 4 do Inc(R0); end;
         while not Zero(R0) do begin Dec(R0); Inc(R2); end;
         while co>0 do begin Inc(R2); co:=co-1; end;
      end;
      Read(vstup);
   end;
   while not Zero(R1) and not Zero(R2) do begin Dec(R1); Dec(R2); end;
   if Zero(R1) and Zero(R2) then Accept;
end.
```

b) Pro zvýšení přehlednosti označíme původní registry R_1, R_2, R_3 a nové registry Q_1, Q_2.

Použijeme myšlenku, kterou známe již z řešení úlohy P–I–4. Zakódujeme obsah všech tří registrů do jediného, druhý registr budeme používat jako pomocný při práci s prvním. Místo tří registrů s obsahem a, b, c budeme mít tedy jeden registr Q_1 s obsahem $2^a 3^b 5^c$. Při simulování každé operace použijeme Q_2 jako pomocný registr. Na začátku i po provedení každé operace v něm bude uložena nula.

Operaci $Inc\,(R_x)$ v původním programu nahradíme tím, že obsah nového registru Q_1 vynásobíme 2, 3, resp. 5. Podobně příkaz $Dec\,(R_x)$ nahradíme příslušným dělením.

Nahradit podmínku *Zero* (R_x) bude trochu komplikovanější. Během vyhodnocování nějaké složené podmínky totiž nemůžeme provádět operace s registry — zjistit, zda je v R_x nula, tedy musíme *před* vyhodnocením příslušné podmínky. Navíc drobné problémy způsobí skutečnost, že tato podmínka se může vyskytovat i v podmínce pro příkaz `while`, kde bude vyhodnocována při každé iteraci (nejen před prvním voláním `while`), a že v jedné podmínce můžeme testovat více proměnných.

Definujme si „makro" (kus výpočtu) *SpocitejZ*, které bude fungovat následovně: Pro každý registr R_x nastaví proměnnou z_x tak, aby v ní byla kladná hodnota právě tehdy, je-li v R_x nula, jinak bude $z_x = 0$. Výpočet makra začne tím, že obsah Q_1 vydělíme příslušným prvočíslem, přičemž si (v proměnné z_x) zapamatujeme zbytek, který jsme dostali při tomto dělení. Vrátíme obsah Q_1 do původního stavu. Jestliže obsah Q_1 byl dělitelný příslušným prvočíslem (tedy neplatí *Zero* (R_x)), bude v z_x nula, jinak tam bude kladný zbytek. Výraz *Zero*(R_x) má tedy v tomto okamžiku stejnou pravdivostní hodnotu jako výraz $(z_x > 0)$.

Každý příkaz „`if` P `then` *příkazy*;" nahradíme makrem *SpocitejZ* a příkazem „`if` P' `then` *příkazy*;", kde podmínka P' vznikla z P tak, že jsme v ní místo všech výskytů výrazu *Zero* (R_x) dali výraz $(z_x > 0)$.

Každý příkaz „`while` P `do` *příkazy*;" nahradíme voláním makra *SpocitejZ* před cyklem a na konci každé iterace, tedy následujícím kusem výpočtu: „*SpocitejZ*; `while` P' `do begin` *příkazy*; *SpocitejZ*; `end`;"

Nová „makra" *Inc*, *Dec* (jimiž nahradíme každý výskyt těchto příkazů v původním programu) a *SpocitejZ* (na simulaci *Zero*) budou tedy vypadat následovně:

```
var x,y,z1,z2,z3,i: byte;
{nové proměnné, které nebyly v původním programu}

{ Inc(Rx) — x je v proměnné x, předpokládáme, že Q2 = 0 }
  if x=1 then y:=2 else if x=2 then y:=3 else y:=5;
  {vynásobíme obsah Q1 číslem y}
  while not Zero(Q1) do begin
     Dec(Q1);
     for i:=1 to y do Inc(Q2);
  end;
  while not Zero(Q2) do begin
     Dec(Q2); Inc(Q1);
  end;

{ Dec(Rx) — x je v proměnné x, předpokládáme, že Q2 = 0 }
  if x=1 then y:=2 else if x=2 then y:=3 else y:=5;
  z:=0;
  {vydělíme obsah Q1 číslem y}
  while not Zero(Q1) do begin
```

```
      Dec(Q1);
      if Zero(Q1) then begin z:=1; break; end; {v Rx byla nula}
      for i:=1 to y-1 do Dec(Q1);
      Inc(Q2);
   end;
   if z=1 then begin
      {obnovíme původní stav Q1 - nic se nemění}
      while not Zero(Q2) do begin
         Dec(Q2); for i:=1 to y do Inc(Q1);
      end;
      Inc(Q1);
   end else begin
      {přesuneme do Q1 podíl}
      while not Zero(Q2) do begin
         Dec(Q2); Inc(Q1);
      end;
   end;

{ SpocitejZ — předpokládáme, že Q2 = 0 }
   {nejdříve chceme spočítat z1, čili budeme dělit dvěma}
   y:=2;
   {vydělíme obsah Q1 číslem y}
   z1:=0;
   while not Zero(Q1) do begin
      Dec(Q1);
      z1:=z1+1;
      if z1=y then begin z1:=0; Inc(Q2); end;
   end;
   {vrátíme zpět původní hodnotu do Q1}
   while not Zero(Q2) do begin
      Dec(Q2); for i:=1 to y do Inc(Q1);
   end;
   for i:=1 to z1 do Inc(Q1);
   {a v proměnné z1 máme hledaný zbytek}
   {opakujeme totéž pro z2, y:=3 a z3, y:=5}
```

Dva důležité detaily, kterých jste si mohli povšimnout:

1. Nesmíme zapomenout ošetřit situaci, že registr R_i obsahuje nulu. V tom případě i po provedení *Dec* (R_i) musí v R_i (podle definice) zůstat nula.
2. Když jsme při simulování příkazů *Inc*, *Dec* a *Zero* potřebovali použít proměnné, muselo se jednat o *nové* proměnné, které se dosud v programu nevyskytovaly. (Co kdyby například původní program obsahoval část „`for i:=1 to 3 do` *Inc* (R_1)“ a my bychom při simulaci *Inc* (R_1) použili proměnnou `i`?) Volných jmen pro nové proměnné máme k dispozici nekonečně mnoho, lehce tedy najdeme nějaké nepoužité.

Snadno nahlédneme, že když tímto způsobem upravíme libovolný program, bude upravený program ekvivalentní s původním — tj. bude dá-

vat pro každý vstup stejný výstup jako původní program. Přitom pokud původní program používal tři registry, upravený program už používá jen dva.

Uvědomte si, že aplikováním tohoto postupu na program používající $k > 3$ registrů dostaneme program používající $k - 1$ registrů. Proto platí poměrně překvapivý výsledek: K libovolné úloze, kterou dokážeme řešit na registrovém počítači, existuje program, jemuž na její řešení stačí dva registry.

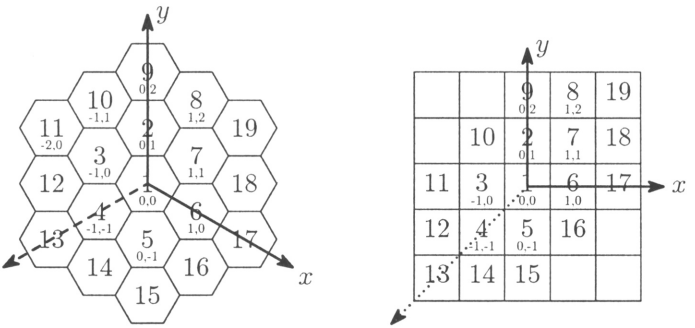
P – III – 4

Na poskakování psíků se můžeme dívat jako na hru. Stav hry lze jednoznačně popsat pozicí obou psíků. Skok psíků představuje tah. Když oba psíci skočí, změní stav hry. Povolené stavy hry budou ty, které odpovídají povoleným pozicím psíků. Pro každý stav hry budeme zkoumat, kolika tahy se do něho dá dostat z počátečního stavu (když jsou oba psíci na výchozích místech). Tento počet tahů budeme označovat jako vzdálenost daného stavu.

Vzdálenost počátečního stavu je 0. Všechny stavy, do nichž se lze z něho dostat jedním tahem, budou ve vzdálenosti 1. Nyní projdeme všechny stavy ve vzdálenosti 1 a hledáme, do kterých nových stavů se z nich dostaneme — ty budou zjevně ve vzdálenosti 2. Takto můžeme analogicky pokračovat pro stavy ve vzdálenosti $3, 4, \ldots$ Skončíme, když najdeme koncový stav (oba psíci jsou na svých cílových místech), nebo když už nenajdeme žádný nový stav. Tato technika prohledávání stavů se nazývá prohledávání do šířky.

Otázkou zůstává, jak pro každý stav určit, do kterých dalších (sousedních) stavů se z něho lze dostat jedním tahem. Pomohlo by nám, kdybychom uměli pro každé políčko na louce určit čísla jeho sousedů. Sousední stavy bychom potom určili snadno. Ze všech možných pohybů oběma psíky 6 směry (36 možností) vyškrtáme skákání stejným směrem, skákání na bodláky, skok některého psíka mimo louku a současný skok obou psíků na totéž políčko.

Abychom našli sousední políčka snadněji, ukážeme si, jak se dá šestiúhelníkový plán louky reprezentovat v obyčejném dvojrozměrném poli. Na políčku 1 si zvolíme dva směry. Jeden určuje rostoucí směr první souřadnice, druhý směr druhé souřadnice. Takto jsme přiřadili každému políčku souřadnice x, y, kterým odpovídají indexy v obyčejném dvojrozměrném poli. V dvojrozměrném poli je už nalezení sousedů lehké. Konkrétně



při naší volbě souřadnicových os budou mít sousedi políčka (x, y) souřadnice $(x, y+1)$, $(x-1, y)$, $(x-1, y-1)$, $(x, y-1)$, $(x+1, y)$ a $(x+1, y+1)$.

Jak zjistíme pro políčko s číslem k jeho souřadnice? Všimněte si, že spirálu můžeme rozložit na vrstvy šestiúhelníkového tvaru. Nejprve určíme, na kolikáté vrstvě spirály se k nachází, potom stranu na této vrstvě, pozici políčka na straně a je to.

Nultá vrstva spirály obsahuje 1 políčko, i-tá vrstva pak $6i$, jelikož každá vrstva má 6 stran a na každé straně je i políček. Celkový počet políček ve spirálách $0 \ldots v$ je tedy $1 + \sum_{i=1}^{v} 6i = 3v^2 + 3v + 1$.

A jak zjistíme, kde se nachází políčko k? Nejdříve spočteme vrstvu — najdeme kladné řešení rovnice $k = 3v^2+3v+1$ a zaokrouhlíme ho nahoru. Po vyřešení dostaneme $v = \left\lceil -\frac{1}{2} + \sqrt{\frac{1}{3}k - \frac{1}{12}} \right\rceil$. (Jednodušší a trochu pomalejší postup: zvyšujeme v, dokud počet políček nedosáhne k.)

Když už víme vrstvu v, kde se políčko nachází, pořadové číslo políčka ve vrstvě (číslováno od 0) dopočítáme snadno: odečteme od k celkový počet políček na dřívějších vrstvách a ještě 1. Na vrstvě v má každá strana v políček, rozdíl tedy stačí vydělit v. Číslo strany s bude podíl, pozice na straně p bude zbytek po tomto dělení.

Již umíme pro dané políčko k spočítat jeho vrstvu v, stranu s a pozici na straně p. Z těchto hodnot dostaneme souřadnice podle následující tabulky (platí pro $v \geqq 1$):

s	x	y
0	$+v-1-p$	$+v$
1	$-1-p$	$+v-1-p$
2	$-v$	$-1-p$
3	$-v+1+p$	$-v$
4	$+1+p$	$-v+1+p$
5	$+v$	$+1+p$

Implementace: Stav hry je jednoznačně reprezentován souřadnicemi x_1, y_1, x_2, y_2 obou psíků. Při prohledávání do šířky si pamatujeme seznam stavů, které jsme již dosáhli, ale ještě jsme z nich nezkoušeli prohledávat nové vrcholy. K tomu nám poslouží fronta. Dále potřebujeme umět pro každý stav rychle zjistit, zda jsme ho ještě neviděli, už viděli, nebo zda se do něj nedá jít (bodláky). K tomu používáme čtyřrozměrné pole, kde je to přímo zapsáno. (Přesně totéž se dalo reprezentovat dvojrozměrným polem, které bychom indexovali původními souřadnicemi. Navíc bychom ale potřebovali umět ze souřadnic určit původní číslo políčka.)

Abychom nemuseli při prohledávání do šířky stále kontrolovat, zda se nedostaneme ven z louky, postavíme okolo celé louky bodláky. Zakážeme také stavy, v nichž by byli oba psíci na stejném místě.

Prohledáváme prostor velikosti řádově $O(N^2)$, kde N je velikost louky. Časová i paměťová složitost prohledávání je lineární vzhledem k velikosti tohoto prostoru, tedy $O(N^2)$.

```
program Psici;
const
   EPS = 1.0E-6;    {max. chyba vzniklá v reálných číslech}
   MAX_V = 16;      {největší možná vrstva (i se zarážkou)}
   MAX_STAVU = MAX_V*MAX_V*MAX_V*MAX_V;
   INF = 299999;    {největší možný počet skoků}
   MAX_SMER = 6;    {počet směrů, jimiž se mohou psíci hýbat}

type
   TSour =  record x,y : integer; end; {souřadnice jednoho psíka}
   TStav = record p1, p2 : TSour; end; {souřadnice dvou psíků}
   {prostor pro 2 psíky: [x1,y1,x2,y2] říká, zda tam mohou být}
   TMrizka = array [-MAX_V..MAX_V,-MAX_V..MAX_V,
                    -MAX_V..MAX_V,-MAX_V..MAX_V] of integer;

function dekVrstvu(k: integer): integer; {číslo vrstvy, kde se k nachází}
begin
   dekVrstvu:= trunc(0.5 + sqrt(k/3.0 - 1.0/12.0 - EPS) );
end;

function zakVrstvu(v: integer): integer; {poslední prvek na dané vrstvě}
begin
   zakVrstvu:= 3*v*v - 3*v +1;
end;

function dekoduj(k: integer): TSour; {Dekóduj číslo políčka na souřadnice}
var v, pv, s, ps: integer;
    sour: TSour;
begin
   if k=1 then begin  {pro k=1 (vrstva 0) naše vzorce nefungují}
      sour.x:= 0;
      sour.y:= 0;
   end else begin
```

```
      v:= dekVrstvu(k);
      pv:= k - (3*v*v - 3*v + 1); {pozice ve vrstvě}
      s:= pv div v; {strana šestiúhelníka, kde se pv nachází}
      ps:= pv mod v; {pozice na straně šestiúhelníka}

      case s of
      0: begin sour.x:= +v-1-ps; sour.y:= +v     ; end;
      1: begin sour.x:=   -1-ps; sour.y:= +v-1-ps; end;
      2: begin sour.x:= -v     ; sour.y:=   -1-ps; end;
      3: begin sour.x:= -v+1+ps; sour.y:= -v     ; end;
      4: begin sour.x:=   +1+ps; sour.y:= -v+1+ps; end;
      5: begin sour.x:= +v     ; sour.y:=   +1+ps; end;
      else writeln('Velká chyba v dekoduj');
      end;
   end;
   dekoduj:=sour;
end;

var
   n, m, s1, t1, s2, t2: integer; {vstup}
   v: integer; {max. použitá vrstva pro dané n}
   A: TMrizka; {prohledávaný prostor}
   F: array [0..MAX_STAVU] of TStav; {fronta}

{zakáže všechny situace, kde se nachází bodlák na daném místě}
procedure pridejBodlak(b: TSour);
var x, y: integer;
begin
   for x:= -v to v do for y:= -v to v do begin
      A[x, y, b.x, b.y]:= INF;        {2. psík stojí na bodláku}
      A[b.x, b.y, x, y]:= INF;        {1. psík stojí na bodláku}
   end;
end;

{vyčistí celý prohledávaný prostor}
procedure inicializace;
var x1, y1, x2, y2, i, last: integer;
begin
   {vyčištění prostoru}
   for x1:= -v to v do for y1:= -v to v do
      for x2:= -v to v do for y2:= -v to v do
         A[x1, y1, x2, y2]:= -1;
   {zarážky při okrajích - přidáme umělé bodláky}
   last:= zakVrstvu(v+1);
   for i:=n+1 to last do pridejBodlak(dekoduj(i));
   {zakážeme být oběma psíkům na stejném místě}
   for x1:= -v to v do for y1:= -v to v do A[x1, y1, x1, y1]:= INF;
end;

{posunutí souřadnice daným směrem}
function pohyb(a: TSour; s: integer): TSour;
const smer: array [1..MAX_SMER] of TSour = (
   (x: 0; y: 1), (x:-1; y: 0), (x:-1; y:-1),
   (x: 0; y:-1), (x: 1; y: 0), (x: 1; y: 1) );
begin
```

```
   a.x:= a.x+smer[s].x; a.y:= a.y+smer[s].y; pohyb:=a;
end;

function pracuj: integer;
{prohledávání prostoru, kde se mohou psíci nacházet}
var
   zac: integer;     {pozice prvního prvku ve frontě}
   kon: integer;     {pozice za posledním prvkem ve frontě = volné místo}
   i, j, vzd: integer;
   p1, p2, q1, q2: TSour;
   pt1, pt2: TSour; {dekódované pozice skrýší pro psíky}
begin
   zac:=0; kon:=1; {přidáme počátek do fronty}
   p1:= dekoduj(s1);
   p2:= dekoduj(s2);
   F[zac].p1:= p1; F[zac].p2:= p2;
   A[p1.x, p1.y, p2.x, p2.y]:= 0; {začínáme ve vzdálenosti 0}

   pt1:=dekoduj(t1); pt2:=dekoduj(t2);

   while zac<>kon do begin
      {vybereme stav z fronty a najdeme k němu vzdálenost}
      p1:=F[zac].p1; p2:=F[zac].p2;
      vzd:= A[p1.x, p1.y, p2.x, p2.y];
      inc(zac);

      {zkoušíme všechny kombinace směrů, kam mohou skákat}
      for i:=1 to MAX_SMER do for j:=1 to MAX_SMER do begin
        if i=j then continue; {nemohou skákat stejně}

        {nové pozice psíků}
        q1:= pohyb(p1, i); q2:= pohyb(p2, j);

        {již navštívená pozice resp. zarážka ?}
        if A[q1.x, q1.y, q2.x, q2.y] >= 0 then continue;

        {nově objevený stav -> přidáme do fronty}
        A[q1.x, q1.y, q2.x, q2.y]:= vzd+1;
        F[kon].p1:=q1; F[kon].p2:=q2;
        inc(kon);

        {našli jsme koncový stav?}
        if (pt1.x=q1.x) and (pt1.y=q1.y) and (pt2.x=q2.x) and (pt2.y=q2.y)
           then begin pracuj:=vzd+1; exit; end;
      end;
   end;
   pracuj:= -1;
end;

var x, i: integer;
begin
   while true do begin
      read(n, m);
      if (n=0) and (m=0) then break;
```

```
      v:= dekVrstvu(n);
      inicializace();

      read(s1, t1, s2, t2);
      for i:=1 to m do begin read(x); pridejBodlak(dekoduj(x)); end;
      if (s1=t1) and (s2=t2) then x:=0 {speciální případ} else x:=pracuj;
      if x>=0 then writeln(x) else writeln('nelze');
   end;
end.
```

P – III – 5

Označme $S = \sum_{i=1}^{N} l_i$ počet dní, které AttoSoft potřebuje na dokončení všech programů. Poslední program tedy dokončíme po S dnech.

Pro každý program spočítáme pokutu, kterou bychom za něj zaplatili, kdybychom ho dokončili až po S dnech. Program s nejmenší takovou pokutou zařadíme do rozvrhu jako poslední. Je-li to program číslo i, zbývá nám naplánovat všechny zbývající programy na prvních $S - l_i$ dní, což provedeme stejným způsobem (tzn. opět vybereme jako poslední program s nejnižší pokutou po $S - l_i$ dnech, atd.)

Správnost uvedeného algoritmu dokážeme indukcí vzhledem k počtu programů, které potřebujeme dokončit. Pokud je třeba dokončit jeden program, existuje jen jediný možný rozvrh, a náš algoritmus tedy funguje jistě správně.

Nechť tedy počet programů, které je třeba dokončit, je N a nechť pro libovolný menší počet programů náš algoritmus funguje správně. Označme G řešení získané naším algoritmem (v tomto řešení je posledním programem program číslo i). Nechť existuje jiné, levnější řešení O, které končí programem číslo j. Jestliže $i = j$, pak rozdíl mezi G a O musí být v pořadí prvních $N - 1$ programů. Podle indukčního předpokladu však toto pořadí v řešení G je optimální, proto řešení O nemůže být levnější.

V opačném případě vytvoříme nový rozvrh O' následujícím způsobem. Nechť rozvrh O dokončuje programy v pořadí $o_1, o_2, \ldots, o_N$ a nechť $o_k = i$. Podle rozvrhu O' dokončíme programy v následujícím pořadí: $o_1, o_2, \ldots, o_{k-1}, o_{k+1}, \ldots, o_N, i$. Všimněte si, že řešení O' je nejvýše tak drahé, jako řešení O. Pokuta za programy $o_{k+1}, \ldots, o_N$ je totiž nižší než v řešení O, neboť je dokončíme dříve (pokuta roste s počtem dní po termínu). Navíc pokuta za program i dokončený po S dnech určitě nepřesahuje pokutu za program o_N dokončený po S dnech, jelikož program i jsme vybrali tak, aby tato pokuta byla nejmenší možná.

Řešení O' ale nemůže být levnější než řešení G (platí tu stejný argument, jako v předchozím případě). Proto ani řešení O nemůže být levnější než G. Dokázali jsme, že žádné levnější řešení než G neexistuje, řešení určené naším algoritmem je tedy optimální.

V každém kroku algoritmu musíme spočítat příslušnou pokutu pro každý program, který jsme dosud nezařadili do rozvrhu. Proto časová složitost algoritmu je $O(N^2)$.

Při výpočtu si ještě musíme dávat pozor na to, že pokuta může být až $5\,000 \cdot 100\,000^3 + 5\,000 \cdot 100\,000^2 + \ldots + 5\,000$, tedy přibližně $5 \cdot 10^{18}$, a tak velké číslo se již nevejde do *longint*u a nemůžeme si dovolit použít ani typ *real*, jelikož potřebujeme i u tak velkých čísel rozlišovat rozdíly na řádu jednotek. Většina překladačů naštěstí nabízí 64bitový celočíselný typ — v Turbo Pascalu je to typ *comp*, ve Free Pascalu například typ *QWord*.

```
program Attosoft;

const MAXN = 10000;

var a,b,c,d,l: array [1..MAXN] of longint;
    pouzite: array [1..MAXN] of boolean;
    N,S: longint;

function Cena(prog:integer; den: comp): comp;
begin
   cena:=((a[prog]*den+b[prog])*den+c[prog])*den+d[prog];
end; {function Cena}

procedure Nacti;
var i: integer;
begin
   readln(N);
   S:=0;
   for i:=1 to N do begin
      readln(l[i],a[i],b[i],c[i],d[i]);
      S:=S+l[i];
   end;
end; {procedure Nacti}

procedure Spocitej_rozvrh(d: longint);
var minprog: integer;
    min,cc: comp;
    i: integer;
begin
   {najdi nepoužitý program, který má nejnižší pokutu po d dnech}
   minprog:=-1;
   for i:=1 to N do begin
      if not pouzite[i] then begin
         cc:=Cena(i,d);
         if (minprog=-1) or (cc<min) then begin
```

```
            min:=cc;
            minprog:=i;
         end;
      end;
   end;

   if minprog>-1 then begin
      {označ program jako použitý}
      pouzite[minprog]:=true;
      {sestav zbytek rozvrhu}
      Spocitej_rozvrh(d-l[minprog]);
      {vypiš poslední program na konci rozvrhu}
      writeln(minprog);
   end;
end; {procedure Spocitej_rozvrh}

var i:integer;

begin
   Nacti;
   for i:=1 to N do pouzite[i]:=false;
   Spocitej_rozvrh(S);
end.
```

Přípravná soustředění před 45. MMO

V průběhu 53. ročníku se konalo výběrové soustředění pro přípravu na mezinárodní matematickou olympiádu bezprostředně po skončeném celostátním kole kategorie A, a to od 5. do 9. dubna 2004 v Kostelci nad Černými lesy nedaleko Prahy. Na soustředění bylo pozváno 9 nejlepších řešitelů III. kola kategorie A. Soustředění bylo zaměřeno na přípravu reprezentantů a ke konečné nominaci šestičlenného družstva.

Úspěšnost jednotlivých studentů ukazuje následující tabulka:

Vítězslav Kala	4/4 G Brno, tř. Kpt. Jaroše 14	90
Jan Moláček	4/4 GJKT, Hradec Králové	90
Marek Pechal	6/8 G Zlín, Lesní čtvrť 1364	87,5
Jaromír Kuben	2/4 G Brno, tř. Kpt. Jaroše 14	85,5
František Konopecký	7/8 GLJ Holešov, Palackého 524	84,5
Alexandr Kazda	8/8 G Praha 6, Nad Alejí	84
Pavel Kocourek	3/4 SPŠST Praha 1, Panská	82
Tomáš Gavenčiak	4/4 GMK Bílovec	80,5
Sven Dražan	4/4 G Brno, tř. Kpt. Jaroše 14	67

Na základě uvedených výsledků, v nichž jsou započítány i výsledky oblastního a celostátního kola, bylo prvních šest vybráno do reprezentačního družstva a sedmý byl určen jako náhradník. Toto družstvo nás reprezentovalo i na již tradičním střetnutí s družstvy Slovenska a Polska.

Jednotlivé semináře vedli a úlohy připravili:
dr. *Jaroslav Zhouf* (5. 4.),
dr. *Karel Horák* (6. 4.),
dr. *Pavel Calábek* (7. 4.),
dr. *Jaroslav Švrček* (8. 4.)
a doc. *Jaromír Šimša* (9. 4.).

Úlohy zadané na přípravném soustředění

1. Na stole leží $n \geqq 3$ listů papíru, které jsou očíslovány od 1 do n. Pak jsou listy libovolně rozděleny na dvě hromádky (jedna hromádka může být prázdná) a hledáme v aspoň jedné hromádce právě dva listy, jejichž čísla dávají v součtu druhou mocninu nějakého přirozeného čísla. Dokažte, že

a) když je $n \geqq 15$, pak takové dva listy mohou být vždy nalezeny,

b) když je $n < 14$, pak takové dva listy nemusejí být nutně nalezeny.

2. Najděte všechna přirozená čísla n taková, že zároveň platí

(a) n má právě šest různých dělitelů 1, d_1, d_2, d_3, d_4, n,

(b) $1 + n = 5(d_1 + d_2 + d_3 + d_4)$.

3. Nechť $ABCD$ je tečnový čtyřúhelník. Průsečík přímek AB a CD označme E, průsečík přímek DA a BC označme F. (Nechť B leží mezi A a E, C leží mezi D a E, A leží mezi D a F, B leží mezi C a F.) Středy kružnic vepsaných trojúhelníkům AFB, BEC a ABC označme postupně I_1, I_2 a I_3. Průsečíky přímky I_1I_3 s přímkami EA a ED označme K a L, průsečíky přímky I_2I_3 s přímkami FC a FD označme M a N. Dokažte, že platí $|EK| = |EL|$, právě když platí $|FM| = |FN|$.

4. Dokažte, že

$$\sum_{k=1}^{n-1} \frac{n}{(n-k)2^{k-1}} < 4$$

pro každé přirozené číslo $n \geqq 2$.

5. V rovině jsou dány dvě kružnice k_1, k_2 s poloměry r_1, r_2 ($r_1 < r_2$), jež se vně dotýkají. Jejich společná tečna t_1 se dotýká k_1 v bodě A a k_2 v bodě D. Označme t_2 druhou tečnu kružnice k_1 rovnoběžnou s t_1 a E, F její průsečíky s kružnicí k_2. Jestliže polopřímka z bodu D protne protne přímku t_2 v bodě B a kružnici k_2 v bodě C, je t_1 tečnou kružnice opsané trojúhelníku ABC. Dokažte.

6. Uvažujme abecedu $\{a, b, c, d\}$ a slova složená z n písmen této abecedy. Slovo považujeme za *složité*, jestliže obsahuje dvě stejné skupiny písmen za sebou (např. *caab* nebo *cababdc* jsou složitá slova, ale slovo *abcab* není). Slovo, které není složité, nazveme *jednoduché*. Dokažte, že jednoduchých slov složených z n písmen je více než 2^n.

7. V daném trojúhelníku ABC označme O střed opsané a V střed vepsané kružnice. Dále označme K a M body dotyku připsané kružnice ω_a,

která leží v úhlu BAC a strany BC se dotýká v bodě N. Jestliže střed P úsečky KM leží na opsané kružnici trojúhelníku ABC, leží body O, V a N v přímce. Dokažte.

8. Nechť $\mathbb{R}$ je množina všech reálných čísel. Určete všechny funkce f, g: $\mathbb{R} \to \mathbb{R}$ takové, že pro všechna reálná čísla x, y platí

$$f(x) - f(y) = (x - y)g(x + y).$$

9. Určete počet všech nekonečných posloupností $a_1, a_2, a_3, \ldots$ čísel $+1$ a -1, které vyhovují současně podmínkám:

(i) Pro všechna přirozená čísla m, n platí

$$a_{mn} = a_m a_n.$$

(ii) V každé trojici po sobě jdoucích členů (a_n, a_{n+1}, a_{n+2}) se vyskytuje současně jak číslo $+1$, tak číslo -1.

10. Určete všechny funkce $f\colon (1, +\infty) \to (1, +\infty)$ takové, že pro všechna reálná čísla $x, y > 1$ a pro všechna kladná reálná čísla m, n platí

$$f(x^m y^n) \leqq f(x)^{\frac{1}{4m}} f(y)^{\frac{1}{4n}}.$$

11. Určete všechna přirozená čísla n mající tuto vlastnost: Nechť p je mnohočlen s celočíselnými koeficienty takový, že $0 \leqq p(k) \leqq n$ pro $k = 0, 1, 2, \ldots, n + 1$, potom platí

$$p(0) = p(1) = p(2) = \ldots = p(n + 1) = 0.$$

12. Nechť I je středem kružnice vepsané danému trojúhelníku ABC. Uvažujme kružnici se středem v bodě I, která protíná strany BC, CA, AB po řadě vždy ve dvou vnitřních bodech D a P, E a Q, F a R. Nechť dále dvojice úseček EF a QR, FD a RP, DE a PQ se protínají po řadě v bodech S, T, U. Dokažte, že kružnice opsané trojúhelníkům FRT, DPU a EQS procházejí společným bodem.

13. Dokažte, že každém tětivovém čtyřúhelníku $ABCD$ platí nerovnost

$$\big||AC| - |BD|\big| < \big||AB| - |CD|\big|.$$

Určete, kdy nastane rovnost.

14. V obdélníku o rozměrech 20×25 je umístěno 120 jednotkových čtverců. Dokažte, že v tomto obdélníku existuje kružnice o poloměru $\frac{1}{2}$, která neprotíná žádný z jednotkových čtverců.

15. Čísla $x - 2004$ a $y - 2004$ jsou druhé mocniny dvou po sobě jdoucích celých čísel. Určete největší možnou hodnotu největšího společného dělitele čísel x a y.

16. Určete nejmenší hodnotu výrazu $x^2 + 4xy + 4y^2 + 2z^2$, kde x, y, z jsou kladná reálná čísla splňující podmínku $xyz = 32$.

17. Rovnoběžník $ABCD$ má vnitřní úhel $60°$ u vrcholu A. Označme O střed kružnice opsané trojúhelníku ABD a předpokládejme, že polopřímka AO protne osu vnějšího úhlu při vrcholu C rovnoběžníku v bodě $K \neq C$. Najděte možné hodnoty poměru $|AO| : |OK|$.

18. Určete nejmenší přirozené číslo n, pro které platí: Z libovolné n-tice navzájem různých celých čísel lze vybrat čtyři různá čísla a, b, c, d, pro která je číslo $a + b - c - d$ dělitelné dvaceti.

19. Šachovnice $(n - 1) \times (n - 1)$ má $(n - 1)^2$ čtvercových polí, která mají dohromady n^2 vrcholů. Kolika způsoby lze těchto n^2 bodů obarvit červenou a modrou barvou tak, aby každé pole mělo dva modré a dva červené vrcholy?

BÍLOVEC, 21.–22. ČERVNA 2004

V rámci závěrečné přípravy před MMO se uskutečnilo již čtvrté mezinárodní střetnutí mezi týmy České republiky, Polska a Slovenska. Jednotlivé země reprezentovala šestice účastníků, kteří si vybojovali ve svých zemích postup na 45. MMO v Athénách.

Soutěž se uskutečnila v termínu 21.–22. 6. 2004 v severomoravském Bílovci. Všechna tři reprezentační družstva přicestovala na místo konání již v neděli večer 20. 6. 2004. Organizace a průběh soutěže zůstal zachován z předešlých ročníků — je přizpůsoben stylu III. kola naší MO a podmínkám na MMO. Soutěžícím byly ve dvou dnech předloženy dvě trojice soutěžních úloh, přitom za každou z úloh mohli získat nejvýše 7 bodů, tj. celkově (stejně jako na MMO) 42 body. Na každou trojici úloh měli soutěžící vyhrazeno 4,5 hodiny.

Pořadí	Jméno	Země	body	Součet
1.	Mateusz Michałek	POL	6 7 6 7 7 5	38
2.–3.	*Vítězslav Kala*	CZE	6 7 1 7 7 7	35
	František Simančík	SVK	7 6 1 7 7 7	35
4.–5.	Kamil Duszenko	POL	6 7 0 7 7 7	34
	Tomáš Váňa	SVK	5 7 1 7 7 7	34
6.	*František Konopecký*	CZE	5 7 0 7 7 7	33
7.–8.	*Jaromír Kuben*	CZE	7 7 0 7 1 7	29
	Jan Moláček	CZE	7 0 1 7 7 7	29
9.	Alexandr Kazda	CZE	3 7 0 4 7 7	28
10.–11.	*Marek Pechal*	CZE	6 1 0 5 7 7	26
	Michał Pilipczuk	POL	4 0 1 7 7 7	26
12.	Andrzej Grzesik	POL	5 7 0 7 1 3	23
13.	Ondrej Budáč	SVK	4 0 1 6 6 3	20
14.	Hana Budáčová	SVK	2 0 1 7 1 6	17
15.	Peter Černo	SVK	5 0 0 2 1 7	15
16.	Piotr Danilewski	POL	0 0 0 7 0 6	13
17.	Jakub Kallas	POL	0 0 0 2 1 7	10
18.	Jozef Bodnár	SVK	3 0 1 2 0 3	9

Návrh všech šesti úloh (a jejich vzorová řešení) připravili členové úlohové komise z České republiky — dr. *Jaroslav Švrček* a doc. *Jaromír Šimša*. Úlohy koordinovala mezinárodní komise ve složení *Jaromír Šimša*, *Jaroslav Švrček* a *Karel Horák* za Českou republiku, *Pavol Novotný* a *Ján Mazák* za Slovensko a *Waldemar Pompe* a *Adam Osękowski* za Polsko.

Texty soutěžních úloh

1. Dokažte, že reálná čísla p, q, r splňují podmínku

$$p^4(q-r)^2 + 2p^2(q+r) + 1 = p^4,$$

právě když kvadratické rovnice

$$x^2 + px + q = 0, \qquad y^2 - py + r = 0$$

mají reálné kořeny (ne nutně různé), které lze označit $x_{1,2}$ resp. $y_{1,2}$ v takovém pořadí, že platí rovnost $x_1y_1 - x_2y_2 = 1$. (*J. Šimša*)

2. Dokažte, že pro každé přirozené číslo k existuje nejvýše konečně mnoho takových trojic navzájem různých prvočísel p, q, r, pro něž je číslo $qr - k$ násobkem p, číslo $pr - k$ násobkem q a současně číslo $pq - k$ násobkem r.

3. Uvnitř tětivového čtyřúhelníku $ABCD$ je dán bod P tak, že platí

$$|\sphericalangle BPC| = |\sphericalangle BAP| + |\sphericalangle PDC|.$$

Označme E, F, G paty kolmic z bodu P po řadě na přímky AB, AD a DC. Dokažte, že trojúhelník FEG je podobný trojúhelníku PBC. (*Toshio Seimiya*)

4. V oboru reálných čísel řešte soustavu rovnic

$$\frac{1}{xy} = \frac{x}{z} + 1, \quad \frac{1}{yz} = \frac{y}{x} + 1, \quad \frac{1}{zx} = \frac{z}{y} + 1.$$

(*J. Földes*)

5. Uvnitř stran AB, BC, CA daného trojúhelníku ABC jsou zvoleny po řadě body K, L, M tak, že platí

$$\frac{|AK|}{|KB|} = \frac{|BL|}{|LC|} = \frac{|CM|}{|MA|}.$$

Dokažte, že trojúhelníky ABC a KLM mají společný průsečík výšek, právě když je trojúhelník ABC rovnostranný. (*P. Černek*)

6. Na stole leží k hromádek o $1, 2, \ldots, k$ kamenech, kde $k \geqq 3$. V prvním kroku vybereme tři libovolné hromádky na stole, sloučíme je do jedné a z této nové hromádky odstraníme jeden kámen (pryč ze stolu). Ve druhém kroku opět sloučíme některé tři hromádky do jedné a pak z ní odebereme dva kameny. Obecně v i-tém kroku sloučíme libovolné tři hromádky, ve kterých je dohromady více než i kamenů, do jedné hromádky a pak z ní i kamenů odstraníme. Předpokládejme, že po několika krocích zůstane na stole jediná hromádka, v níž je p kamenů. Dokažte, že číslo p je úplný kvadrát, právě když obě čísla $2k+2$ a $3k+1$ jsou úplné kvadráty. Dále pak najděte nejmenší k, pro které je číslo p úplný kvadrát.

(*R. Kučera*)

Řešení úloh

1. Předpokládejme, že uvažované kořeny obou daných rovnic splňují rovnost $x_1y_1 - x_2y_2 = 1$. Podle známého vzorce je

$$x_{1,2} = \frac{-p \pm K}{2} \qquad \text{a} \qquad y_{1,2} = \frac{p \pm L}{2}, \tag{1}$$

kde reálná čísla K, L splňují rovnosti $K^2 = p^2 - 4q$ a $L^2 = p^2 - 4r$ (číslům K, L přiřadíme znaménka podle očíslování kořenů). Potom

$$1 = x_1y_1 - x_2y_2 = \frac{(-p+K)(p+L) - (-p-K)(p-L)}{4} = \frac{p(K-L)}{2},$$

odkud $p \neq 0$ a $K - L = 2/p$. Dosadíme-li to do rovnosti

$$(K+L)(K-L) = K^2 - L^2 = (p^2 - 4q) - (p^2 - 4r) = 4(r-q),$$

vyjde nám $K + L = 2p(r-q)$. Ze získaných vyjádření čísel $K+L$ a $K-L$ dostaneme $K = 1/p - p(q-r)$, po umocnění $K^2 = 1/p^2 - 2(q-r) + p^2(q-r)^2$. Porovnáme-li to s rovností $K^2 = p^2 - 4q$, obdržíme po snadné úpravě požadovanou podmínku

$$p^4(q-r)^2 + 2p^2(q+r) + 1 = p^4, \tag{2}$$

Předpokládejme naopak, že platí rovnice (2). Pak zřejmě $p \neq 0$. Rovnici upravíme dvěma obdobnými způsoby do tvarů

$$p^4(r-q)^2 + 2p^2(r-q) + 1 = p^4 - 4p^2q$$

a

$$p^4(q-r)^2 + 2p^2(q-r) + 1 = p^4 - 4p^2r;$$

odtud po vydělení číslem p^2 zjišťujeme, že diskriminanty uvažovaných kvadratických rovnic mají vyjádření

$$p^2 - 4q = \left(\frac{p^2(r-q)+1}{p}\right)^2 \quad \text{a} \quad p^2 - 4r = \left(\frac{p^2(q-r)+1}{p}\right)^2,$$

takže to jsou nezáporná čísla a příslušné (reálné) kořeny mají tvar (1), kde

$$K = \frac{p^2(r-q)+1}{p} \quad \text{a} \quad L = -\frac{p^2(q-r)+1}{p}.$$

Znaménka čísel K a L jsme zvolili tak, aby vyšlo (viz důkaz 1. implikace)

$$x_1y_1 - x_2y_2 = \frac{p(K-L)}{2} = \frac{p}{2} \cdot \left(\frac{p^2(r-q)+1}{p} + \frac{p^2(q-r)+1}{p}\right) = 1.$$

2. Navzájem různá prvočísla p, q, r vyhovují podmínkám úlohy, právě když číslo $pq + qr + rp - k$ je dělitelné každým z čísel p, q, r, neboli jejich součinem pqr. Rovnost $pq + qr + rp - k = n \cdot pqr$ pro vhodné celé n přepišme do tvaru $k = pq + qr + rp - n \cdot pqr$. Je-li $n \leqq 0$, plyne z poslední rovnosti, že $\max\{pq, pr, qr\} \leqq k$; pak ovšem každé z prvočísel p, q, r je nejvýše $\frac{1}{2}k$ a takových trojic je konečný počet. Je-li $n \geqq 1$, dostáváme odhad $k \leqq pq + qr + rp - pqr$. Zřejmě můžeme předpokládat, že $2 \leqq p < q < r$.

Kdyby bylo $r \geqq 7$, dostali bychom

$$\frac{pq + qr + rp - npqr}{pqr} = \frac{1}{p} + \frac{1}{q} + \frac{1}{r} - n \leqq \frac{1}{2} + \frac{1}{3} + \frac{1}{7} - 1 < 0,$$

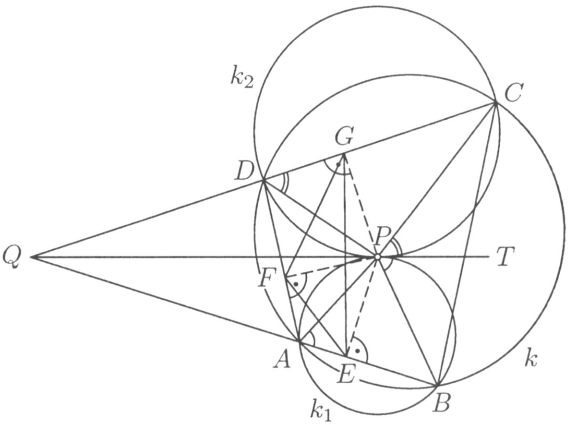
což nejde. Může tedy být jedině $r = 5$, $q = 3$ a $p = 2$.

3. Označme k kružnici opsanou čtyřúhelníku $ABCD$ a k_1, k_2 kružnice opsané trojúhelníkům PAB, PCD. Uvnitř úhlu BPC uvažujme takovou polopřímku PT, pro niž platí $|\sphericalangle BPT| = |\sphericalangle BAP|$. Podle zadání pak platí (obr. 42)

$$|\sphericalangle CPT| = |\sphericalangle BPC| - |\sphericalangle BPT| = |\sphericalangle BPC| - |\sphericalangle BAP| = |\sphericalangle PDC|.$$

Přímka PT je tudíž společnou vnitřní tečnou obou kružnic k_1 a k_2.

Uvažujme nejprve případ, kdy strany AB a CD uvažovaného tětivového čtyřúhelníku nejsou rovnoběžné. Vzhledem k tomu, že úsečky AB a CD jsou společnými tětivami odpovídajících dvojic kružnic k_1,



Obr. 42

k a k_2, k, existuje jediný bod Q, který má stejnou mocnost ke všem třem kružnicím k, k_1 a k_2. Tímto bodem Q je společný bod všech tří přímek (chordál) AB, CD a PT. Bez újmy na obecnosti předpokládejme, že bod Q leží na polopřímce BA za bodem A (obr. 42).

Podle Thaletovy věty jsou zřejmě čtyřúhelníky $AEPF$, $FPGD$ a $QEPG$ tětivové. Z rovností příslušných obvodových úhlů tak plyne

$$\begin{aligned}|\sphericalangle FEG| &= |\sphericalangle FEP| - |\sphericalangle GEP| = |\sphericalangle FAP| - |\sphericalangle GQP| = \\ &= |\sphericalangle DAP| - |\sphericalangle DQP| = |\sphericalangle QDA| - |\sphericalangle QPA|,\end{aligned}$$

neboť úhly při vrcholech A, P a úhly při vrcholech D, Q v nekonvexním čtyřúhelníku $APQD$ dávají stejný součet: $|\sphericalangle DAP| + |\sphericalangle QPA| = |\sphericalangle QDA| + |\sphericalangle DQP|$. Pro úsekový úhel QPA navíc platí $|\sphericalangle QPA| = |\sphericalangle PBA|$, takže

$$\begin{aligned}|\sphericalangle FEG| &= |\sphericalangle QDA| - |\sphericalangle QPA| = |\sphericalangle QDA| - |\sphericalangle PBA| = \\ &= |\sphericalangle QBC| - |\sphericalangle PBA| = |\sphericalangle PBC|.\end{aligned}$$

Analogicky dokážeme, že je $|\sphericalangle FGE| = |\sphericalangle PCB|$. Trojúhelníky FEG a PBC se tedy shodují ve dvou vnitřních úhlech a jsou podobné (uu).

Jsou-li přímky AB a CD rovnoběžné, je $ABCD$ rovnoramenný lichoběžník se základnami AB a CD. Odtud plyne, že body E, P, G leží na ose souměrnosti lichoběžníku $ABCD$ a trojúhelníky APD a BPC jsou shodné. Jak snadno plyne z vlastností obvodových úhlů tětivových

čtyřúhelníků $AEPF$ a $FPGD$, jsou trojúhelníky EFG a APD podobné (je $|\sphericalangle FEG| = |\sphericalangle PAD|$ a $|\sphericalangle EGF| = |\sphericalangle ADP|$). Odtud již plyne podobnost trojúhelníků FEG a PBC. Tím je důkaz ukončen.

Poznámka. V předvedeném řešení jsme potřebovali ukázat, že se trojúhelníky FEG a PBC shodují ve dvou úhlech. Rovnost úhlů $|\sphericalangle EFG| = |\sphericalangle BPC|$ plyne téměř okamžitě z rovností příslušných obvodových úhlů v tětivových čtyřúhelnících $AEPF$, $FPGD$.

4. Z tvaru rovnic plyne podmínka $xyz \neq 0$. Dvě z čísel x, y, z musejí mít stejné znaménko, pak je kladná pravá strana rovnice, ve které jsou tato dvě čísla v podílu, proto je kladná i příslušná levá strana, takže zbývající z čísel x, y, z má totéž znaménko jako prvá dvě. Platí tedy buď $x, y, z > 0$, nebo $x, y, z < 0$.

Zabývejme se pouze prvním případem, druhý se totiž převede na první změnou řešení (x, y, z) na řešení $(-x, -y, -z)$. První dvě rovnice soustavy vynásobme výrazem xyz a odečtěme je, po úpravě dostaneme $z - x = y(x^2 - yz)$. Je-li trojice (x, y, z) řešením, jsou řešeními i trojice (y, z, x) a (z, x, y), které dostaneme cyklickou záměnou. Proto můžeme předpokládat, že $x = \max\{x, y, z\}$. Potom $z - x \leqq 0$ a $x^2 - yz \geqq 0$ (nezapomeňme, že $x, y, z > 0$), takže z rovnosti $z - x = y(x^2 - yz)$ a podmínky $y > 0$ plyne $z - x = x^2 - yz = 0$, což znamená $x = y = z$. Máme tedy jedinou rovnici $1/x^2 = 1 + 1$, která má (jediný) kladný kořen $x = \frac{1}{2}\sqrt{2}$.

Odpověď. Soustava má právě dvě řešení $x = y = z = \pm\frac{1}{2}\sqrt{2}$.

5. Bod V roviny trojúhelníku ABC je průsečíkem jeho výšek, právě když platí zároveň $AV \perp BC$ a $BV \perp AC$, neboli $\boldsymbol{AV} \cdot \boldsymbol{BC} = 0$ a $\boldsymbol{BV} \cdot \boldsymbol{AC} = 0$. Po dosazení $\boldsymbol{BC} = \boldsymbol{BV} - \boldsymbol{CV}$, $\boldsymbol{AC} = \boldsymbol{AV} - \boldsymbol{CV}$ a snadné úpravě dostaneme ekvivalentní podmínku ve formě rovnosti skalárních součinů

$$\boldsymbol{AV} \cdot \boldsymbol{BV} = \boldsymbol{BV} \cdot \boldsymbol{CV} = \boldsymbol{CV} \cdot \boldsymbol{AV}. \tag{1}$$

Naším úkolem je tedy zjistit, kdy platí soustava (1) zároveň s obdobnou soustavou

$$\boldsymbol{KV} \cdot \boldsymbol{LV} = \boldsymbol{LV} \cdot \boldsymbol{MV} = \boldsymbol{MV} \cdot \boldsymbol{KV}, \tag{2}$$

která vyjadřuje, že bod V je průsečíkem výšek trojúhelníku KLM. Vyjádříme vektory z (2) jako lineární kombinace vektorů z (1). Podle zadání existuje číslo λ, $0 < \lambda < 1$, pro které platí

$$\boldsymbol{AK} = \lambda\,\boldsymbol{AB}, \quad \boldsymbol{BL} = \lambda\,\boldsymbol{BC}, \quad \boldsymbol{CM} = \lambda\,\boldsymbol{CA}.$$

Dosadíme-li do první rovnosti $\boldsymbol{AK} = \boldsymbol{AV} - \boldsymbol{KV}$ a $\boldsymbol{AB} = \boldsymbol{AV} - \boldsymbol{BV}$, dostaneme po úpravě první z rovností

$$\boldsymbol{KV} = (1-\lambda)\boldsymbol{AV} + \lambda\boldsymbol{BV}, \quad \boldsymbol{LV} = (1-\lambda)\boldsymbol{BV} + \lambda\boldsymbol{CV},$$
$$\boldsymbol{MV} = (1-\lambda)\boldsymbol{CV} + \lambda\boldsymbol{AV};$$

druhé dvě rovnosti odvodíme analogicky. Odtud vynásobením dostaneme

$$\boldsymbol{KV}\cdot\boldsymbol{LV} = (1-\lambda)^2\boldsymbol{AV}\cdot\boldsymbol{BV} + \lambda(1-\lambda)\boldsymbol{AV}\cdot\boldsymbol{CV} + \lambda(1-\lambda)\boldsymbol{BV}^2 =$$
$$= (1-\lambda)s + \lambda(1-\lambda)\boldsymbol{BV}^2,$$

kde písmeno s označuje společnou hodnotu součinů z (1). Analogicky platí

$$\boldsymbol{KV}\cdot\boldsymbol{MV} = (1-\lambda)s+\lambda(1-\lambda)\boldsymbol{AV}^2 \quad \text{a} \quad \boldsymbol{LV}\cdot\boldsymbol{MV} = (1-\lambda)s+\lambda(1-\lambda)\boldsymbol{BV}^2.$$

Vidíme, že soustava (2) je ekvivalentní se soustavou rovností

$$\lambda(1-\lambda)\boldsymbol{AV}^2 = \lambda(1-\lambda)\boldsymbol{BV}^2 = \lambda(1-\lambda)\boldsymbol{CV}^2,$$

která je s ohledem na $\lambda(1-\lambda) \neq 0$ splněna, právě když $|AV| = |BV| = |CV|$. Tato podmínka znamená, že průsečík výšek V trojúhelníku ABC splývá se středem kružnice opsané. To nastane, právě když je trojúhelník ABC rovnostranný.

Jiné řešení. Je-li trojúhelník ABC rovnostranný, označme O střed jeho kružnice opsané. Při jedné z rotací o $120°$ kolem bodu O platí $A \mapsto B \mapsto C \mapsto A$ a rovněž $K \mapsto L \mapsto M \mapsto K$, neboť například body K, resp. L dělí ve stejném poměru úsečku AB, resp. úsečku BC, jež je obrazem první úsečky ve zmíněné rotaci. To znamená, že i trojúhelník KLM je rovnostranný a bod O je středem (a tedy i průsečíkem výšek) obou trojúhelníků ABC a KLM.

Jestliže ABC není rovnostranný trojúhelník, je střed O jeho opsané kružnice různý od těžiště T. Snadno ukážeme, že bod $T = \frac{1}{3}(A+B+C)$ je těžištěm i trojúhelníku KLM: podle zadání totiž existuje číslo $\lambda \in (0,1)$ tak, že

$$K = \lambda A + (1-\lambda)B, \quad L = \lambda B + (1-\lambda)C, \quad M = \lambda C + (1-\lambda)A,$$

odkud okamžitě plyne rovnost $\frac{1}{3}(K+L+M) = \frac{1}{3}(A+B+C)$.

Připusťme, že trojúhelníky ABC a KLM mají kromě těžiště T společné i ortocentrum, které označíme H. Podle známé věty leží body H, T, O v uvedeném pořadí na jedné přímce (zvané Eulerova přímka trojúhelníku ABC), přičemž platí $|HT| : |TO| = 2 : 1$. Střed kružnice opsané je tedy těžištěm a ortocentrem jednoznačně určen; úvahou o Eulerově přímce trojúhelníku KLM tak zjišťujeme, že bod O je nejen středem kružnice opsané trojúhelníku ABC, ale i středem kružnice opsané trojúhelníku KLM. Body K, L, M mají proto stejnou vzdálenost od bodu O, takže mají i stejnou mocnost ke kružnici opsané trojúhelníku ABC. Tyto mocnosti se rovnají veličinám

$$\begin{aligned} -|AK| \cdot |BK| &= -p(1-p)|AB|^2, \\ -|BL| \cdot |CL| &= -p(1-p)|BC|^2, \\ -|CM| \cdot |AM| &= -p(1-p)|AC|^2, \end{aligned}$$

jejichž porovnáním dostaneme rovnosti $|AB| = |BC| = |CA|$ (neboť $\lambda \notin \{0, 1\}$). To je ve sporu s předpokladem, že trojúhelník ABC není rovnostranný.

6. Po i provedených krocích bude na stole $k - 2i$ hromádek; zůstane-li proto nakonec na stole jediná hromádka, bylo číslo k liché a celkový počet kroků byl $\frac{1}{2}(k-1)$. Rozlišíme, zda číslo k dává při dělení čtyřmi zbytek 1, nebo zbytek 3.

Případ $k = 4c+1$. Na začátku leží na stole $1 + \ldots + k = \frac{1}{2}k(k+1) = (4c+1)(2c+1)$ kamenů, ve všech $2c$ krocích odstraníme celkem $1 + \ldots + 2c = c(2c+1)$ kamenů, takže počet kamenů v poslední hromádce bude

$$p = (4c+1)(2c+1) - c(2c+1) = (2c+1)(3c+1).$$

Čísla $2c+1$ a $3c+1$ jsou ovšem nesoudělná, takže p je úplný kvadrát, právě když jsou úplné kvadráty obě čísla $2c+1$ a $3c+1$, tedy právě když jsou úplné kvadráty jejich čtyřnásobky $4(2c+1) = 2k+2$ a $4(3c+1) = 3k+1$.

Případ $k = 4c+3$. Na začátku leží na stole $1 + \ldots + k = \frac{1}{2}k(k+1) = 2(c+1)(4c+3)$ kamenů, ve všech $2c+1$ krocích odstraníme celkem $1 + \ldots + (2c+1) = (c+1)(2c+1)$ kamenů, takže počet kamenů v poslední hromádce bude

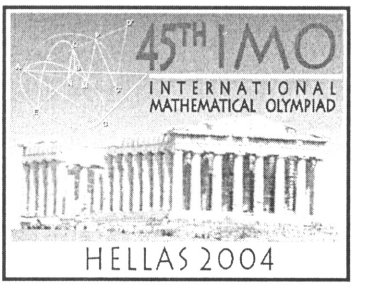
$$p = 2(c+1)(4c+3) - (c+1)(2c+1) = (c+1)(6c+5).$$

Kdyby bylo číslo p úplný kvadrát, musela by být úplnými kvadráty obě nesoudělná čísla $c+1$ a $6c+5$. Ukažme, že to není možné: připusťme

existenci přirozených čísel x, y takových, že $c+1 = x^2$ a $6c+5 = y^2$. Z rovnosti $6x^2 - y^2 = 1$ plyne, že číslo y je liché, tudíž číslo y^2 dává při dělení osmi zbytek 1. Číslo $6x^2$ pak při dělení osmi dává zbytek 2, odkud plyne, že číslo $3x^2$ při dělení čtyřmi dává zbytek 1, což není možné. V případě $k = 4c+3$ tedy p nikdy není úplný kvadrát, stejně jako není úplný kvadrát ani číslo $3k+1 = 12c+10$ (sudé číslo, jež není dělitelné čtyřmi).

Najdeme nyní nejmenší číslo $k = 4c+1$, $c \geqq 1$, pro které jsou obě čísla $2c+1$ a $3c+1$ úplné kvadráty. Z rovností $2c+1 = x^2$ a $3c+1 = y^2$ pro vhodná celá $x, y > 1$ plyne $3x^2 - 2y^2 = 1$, takže x je liché, číslo $2y^2$ pak při dělení čtyřmi dává zbytek 2, takže i y je liché. Položme $x = 2a+1$, $y = 2b+1$ (a, b celá kladná) a dosaďme do rovnosti $3x^2 - 2y^2 = 1$. Po úpravě dostaneme vztah $3a(a+1) = 2b(b+1)$, kam postupně dosazujeme přirozená čísla $a = 1, 2, \ldots$. Najdeme tak rychle nejmenší vyhovující $a = 4$ a $b = 5$, kterým odpovídají $x = 9$, $y = 11$, $c = 40$ a $k = 161$.

45. mezinárodní matematická olympiáda



Zhruba měsíc před zahájením letních olympijských her se v době od 4. do 18. července 2004 uskutečnil v hlavním městě Řecka Athénách i 45. ročník mezinárodní matematické olympiády. Olympiády se tentokrát zúčastnilo 486 studentů z 85 zemí (každou zemi reprezentuje vždy nejvýše šest soutěžících).

Výběr soutěžících za Českou republiku byl proveden v Kostelci nad Černými lesy na závěrečném soutěžním soustředění devíti nejúspěšnějších účastníků ústředního kola kategorie A. Vybraní reprezentanti se pak ještě zúčastnili trojutkání v severomoravském Bílovci mezi Českou republikou, Polskem a Slovenskem, kde soutěžili reprezentanti zúčastněných zemí za podmínek podobných jako při soutěži na MMO. Po této přípravě odjela do Athén tato šestice soutěžících: *Vítězslav Kala* a *Jaromír Kuben* z Gymnázia na tř. Kpt. Jaroše v Brně, *František Konopecký* z Gymnázia Holešov, *Jan Moláček* z Gymnázia J. K. Tyla v Hradci Králové a *Marek Pechal* z Gymnázia ve Zlíně, Lesní čtvrť. Vedoucím české delegace byl RNDr. *Karel Horák*, CSc., z Matematického ústavu Akademie věd v Praze, jeho zástupcem a pedagogickým vedoucím byl RNDr. *Jaroslav Švrček*, CSc., z Univerzity Palackého v Olomouci.

Mezinárodní jury složená z vedoucích jednotlivých zúčastněných zemí strávila prvních šest dnů výběrem úloh a přípravou jejich textů v národních jazycích v Delfách. Den po příletu soutěžících se konalo slavnostní zahájení v athénském Paláci kultury (Megaro Mousikis). Vlastní soutěž pak proběhla v pondělí a v úterý 12. a 13. července ve dvou velkých sálech Matematického ústavu Athénské univerzity. Každý z těchto dnů řešili soutěžící trojici úloh po dobu 4,5 hodiny. Za každou úlohu mohli získat maximálně 7 bodů.

Kromě vlastní soutěže byl pro studenty připraven další zajímavý program: mimo prohlídku starověkých Athén s Akropolí po soutěži všichni účastníci během celodenního výletu navštívili slavné Mykény, přímořské městečko Nauplios a antický amfiteátr v Epidauru.

Výsledky našich jsou uvedeny v následující tabulce:

	Body za úlohu						Body	Cena
Umístění	1	2	3	4	5	6		
143.–158. Vítězslav Kala	6	6	0	6	3	1	22	III.
351.–366. Alexandr Kazda	0	5	0	3	0	0	8	
93.–100. František Konopecký	7	2	1	7	7	2	26	II.
124.–142. Jaromír Kuben	6	3	0	7	7	0	23	III.
113.–123. Jan Moláček	7	3	0	7	6	1	24	II.
382.–395. Marek Pechal	2	2	0	0	2	0	6	
Celkem	28	21	1	30	25	4	109	

Jak je z tabulky vidět, z našich si nejlépe vedli *František Konopecký* z Gymnázia Holešov a *Jan Moláček* z Gymnázia v Hradci Králové. Ten zopakoval svůj úspěch z předchozí MMO a opět přivezl stříbrnou medaili. Nezklamali ani maturant *Vítězslav Kala*, který oproti předešlé MMO získal dvojnásobný počet bodů, a student teprve 2. ročníku *Jaromír Kuben*, oba z Gymnázia na tř. Kpt. Jaroše v Brně. Stříbro jim uteklo jen o chloupek.

O náročnosti soutěžních úloh zpravidla svědčí hranice pro zisk medailí, případně počet absolutních vítězů: na bronzovou medaili tentokrát stačilo 16 bodů, stříbro se udělovalo za 24–31 bodů a zlato za alespoň 32 z možného počtu 42 bodů. Plný počet bodů získali čtyři soutěžící: Kanaďan *Jacob Tsimerman*, Maďar *Bela Andras Racz* a dva soutěžící Ruska *Andrej Badzjan* a *Michail Dubašinskij*.

Jak dopadli naši slovenští kolegové, je nejlépe vidět z následující tabulky. Za pozornost stojí zejména bezchybný výkon slovenského družstva v první úloze, která byla opravdu nejlehčí, a nás může jen mrzet, že jsme za ni získali o 14 bodů méně.

	Body za úlohu						Body	Cena
Umístění	1	2	3	4	5	6		
244.–263. Jozef Bodnár	7	2	0	1	4	1	15	HM
113.–123. Ondrej Budáč	7	2	2	7	6	0	24	II.
244.–263. Hana Budáčová	7	1	3	1	3	0	15	HM
244.–263. Peter Černo	7	0	0	7	1	0	15	HM
101.–112. František Simančík	7	4	1	7	3	3	25	II.
101.–112. Tomáš Váňa	7	5	0	7	6	0	25	II.
Celkem	42	14	6	30	23	4	119	

	I	II	III	body
ČLR	6	0	0	220
USA	5	1	0	212
Rusko	4	1	1	205
Vietnam	4	2	0	196
Bulharsko	3	3	0	194
Tchaj-wan	3	3	0	190
Maďarsko	2	3	1	187
Japonsko	2	4	0	182
Írán	1	5	0	178
Rumunsko	1	4	1	176
Ukrajina	1	5	0	174
Korea	2	2	2	166
Bělorusko	0	4	2	154
Indie	0	4	2	151
Izrael	1	1	4	147
Polsko	2	1	1	142
Moldavsko	2	0	4	140
Singapur	0	3	3	139
Mongolsko	0	3	2	135
Velká Británie	1	1	4	134
Brazílie	0	2	4	132
Kanada	1	0	3	132
Kazachstán	2	0	2	132
Srbsko a Černá Hora	0	2	3	132
Německo	0	3	1	130
Řecko	0	2	3	126
Austrálie	1	1	2	125
Gruzie	0	0	5	123
Kolumbie	0	2	2	122
Hongkong	0	2	2	120
Slovensko	0	3	0	119
Turecko	0	2	3	118
Jihoafrická republika	0	3	1	110
Česká republika	0	2	2	109
Thajsko	0	0	4	99
Arménie	0	0	4	98
Mexiko	0	0	3	96
Francie	0	0	4	94
Argentina	1	0	2	92
Chorvatsko	0	0	3	89
Maroko	0	0	3	88
Belgie	0	1	2	86
Macao	0	0	2	86
Estonsko	0	0	2	85
Uzbekistán	0	0	3	79
Švédsko	0	0	3	75
Ázerbájdžán	0	1	0	72
Makedonie	0	0	1	71
Itálie	0	0	2	69
Slovinsko	0	0	2	69
Litva	0	0	0	65
Kirgizie	0	0	1	63
Lotyšsko	0	0	1	63
Indonézie	0	0	1	61
Albánie	0	0	1	57
Španělsko	0	0	1	57
Švýcarsko	0	0	2	57
Nový Zéland	0	0	2	56
Norsko	0	0	0	55
Rakousko	0	0	1	55
Nizozemsko	0	0	0	53
Turkmenistán	0	0	2	52
Finsko	0	0	1	49
Kypr	0	0	1	49
Peru (3)	0	0	2	49
Irsko	0	0	1	48
Uruguay	0	0	0	47
Dánsko	0	0	1	46
Portoriko (5)	0	1	0	43
Bosna a Hercegovina	0	0	0	40
Lucembursko (3)	0	1	0	36
Island	0	0	0	35
Malajsie	0	0	1	34
Srí Lanka	0	0	0	33
Tunisko	0	0	0	31
Trinidad a Tobago (5)	0	0	0	29
Portugalsko	0	0	0	26
Kuba (1)	0	0	1	17
Filipíny (5)	0	0	0	16
Venezuela (2)	0	0	0	15
Ekvádor	0	0	0	14
Mozambik (3)	0	0	0	13
Paraguay (3)	0	0	0	13
Kuvajt	0	0	0	5
Saudská Arábie	0	0	0	4

Jak je patrno z tabulky zúčastněných států, na čelných místech se žádné překvapení nekonalo. Naše i slovenské družstvo se opět nedostalo ani do třetí desítky, o moc lépe se zato vedlo Polákům. (Případná čísla v závorce upozorňují na nižší počet reprezentantů.)

Texty soutěžních úloh

(v závorce je uvedena země, která úlohu navrhla)

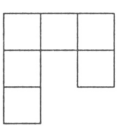
1. Nechť ABC je ostroúhlý trojúhelník, v němž $|AB| \neq |AC|$. Kružnice nad průměrem BC protíná strany AB a AC po řadě v bodech M a N. Označme O střed strany BC. Osy úhlů BAC a MON se protínají v bodě R. Dokažte, že kružnice opsané trojúhelníkům BMR a CNR procházejí společným bodem ležícím na straně BC. (*Rumunsko*)

2. Najděte všechny mnohočleny $P(x)$ s reálnými koeficienty, jež splňují rovnost

$$P(a-b)+P(b-c)+P(c-a)=2\,P(a+b+c)$$

pro všechna reálná čísla a, b, c taková, že $ab+bc+ca=0$. (*Korea*)

3. Nazvěme *dlaždicí* obrazec vytvořený ze šesti jednotkových čtverců jako na obrázku



anebo libovolný obrazec vzniklý jeho otočením či souměrností. Určete všechny pravoúhelníky $m \times n$, které lze dlaždicemi pokrýt tak, že

- ▷ pravoúhelník je pokryt bez mezer a překrytí;
- ▷ žádná část dlaždice nepokrývá plochu vně pravoúhelníku.

(*Estonsko*)

4. Nechť $n \geqq 3$ je celé číslo. Nechť $t_1, t_2, \ldots, t_n$ jsou kladná reálná čísla taková, že

$$n^2+1 > (t_1+t_2+\ldots+t_n)\Big(\frac{1}{t_1}+\frac{1}{t_2}+\ldots+\frac{1}{t_n}\Big).$$

Ukažte, že t_i, t_j, t_k jsou délky stran trojúhelníku pro všechna i, j, k, kde $1 \leqq i < j < k \leqq n$. (*Korea*)

5. V konvexním čtyřúhelníku $ABCD$ úhlopříčka BD nepůlí ani jeden z úhlů ABC, CDA. Bod P leží uvnitř $ABCD$ a splňuje rovnosti

$$|\sphericalangle PBC| = |\sphericalangle DBA| \quad \text{a} \quad |\sphericalangle PDC| = |\sphericalangle BDA|.$$

Dokažte, že $ABCD$ je tětivový, právě když $|AP| = |CP|$. (*Polsko*)

6. Přirozené číslo nazveme *pruhované*, jestliže každé dvě sousední číslice v jeho desítkovém zápise mají různou paritu. Najděte všechna přirozená čísla n taková, že n má pruhovaný násobek. (*Írán*)

Řešení úloh

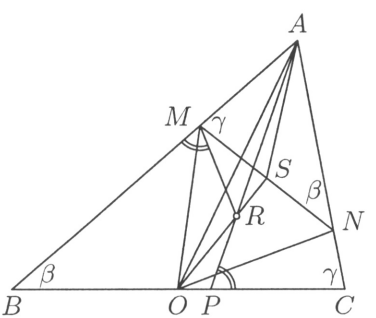
1. Označme S střed úsečky MN a P průsečík osy úhlu BAC se stranou BC. Protože trojúhelníky AMN a ACB jsou podobné (to plyne z vlastností tětivového čtyřúhelníku $BCNM$), přičemž těžnici AS odpovídá těžnice AO, je $|\sphericalangle BAO| = |\sphericalangle CAS|$ (obr. 43), takže osa úhlu BAC je zároveň i osou úhlu OAS. Proto

$$\frac{|RS|}{|RO|} = \frac{|AS|}{|AO|}.$$

Z uvedené podobnosti dále plyne

$$\frac{|AS|}{|AO|} = \frac{|MN|}{|BC|} = \frac{|MS|}{|BO|} = \frac{|MS|}{|MO|},$$

což spolu s předchozí rovností znamená, že MR je osou úhlu OMS.



Obr. 43

Označme vnitřní úhly trojúhelníku ABC obvyklým způsobem. Protože $|OM| = |OB|$, tedy $|\sphericalangle BMO| = \beta$, a protože $|\sphericalangle AMN| = |\sphericalangle BCA| = \gamma$, vychází velikost úhlu OMN jako α. Je tudíž $|\sphericalangle BMR| = \beta + \frac{1}{2}\alpha = |\sphericalangle CPA|$. Dostali jsme, že čtyřúhelník $BPRM$ je tětivový. Analogicky je tětivový i čtyřúhelník $CPRN$. Je tedy bod $P \in BC$ společným bodem obou kružnic opsaných trojúhelníkům BMR a CNR, což jsme měli dokázat.

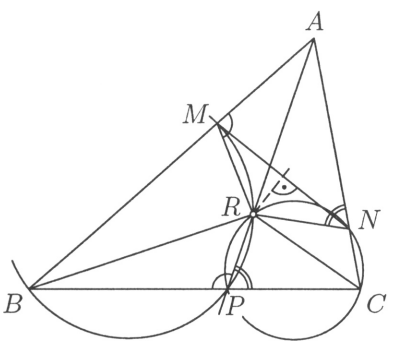
Jiné řešení. (Podle *Františka Konopeckého.*) Protože body M a N leží na kružnici se středem O, je $|OM| = |ON|$. Trojúhelník MNO je tedy rovnoramenný a osa jeho úhlu MON je zároveň osou úsečky MN. Bod R, který je průsečíkem osy úhlu MAN s osou protější strany MN trojúhelníku AMN, leží proto na kružnici trojúhelníku AMN opsané. Přitom

obě osy splynou, jen když $|AM| = |AN|$, což vzhledem k předpokladu $|AB| \neq |AC|$ nelze, neboť z vlastností tětivového čtyřúhelníku $BCNM$ snadno plyne, že trojúhelníky AMN a ACB jsou podobné (shodují se ve dvou úhlech).

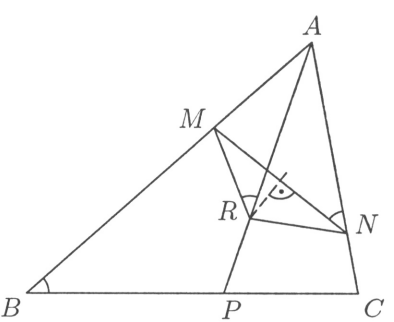
Z mocnosti bodu A ke kružnici s průměrem BC plyne, že bod A má stejnou mocnost i k oběma kružnicím opsaným trojúhelníkům BMR a CNR (obr. 44). Označíme-li P druhý společný bod těchto dvou kružnic (jedním je bod R), musí bod A ležet na jejich společné sečně PR (to je právě množina všech bodů, jež mají k oběma kružnicím stejnou mocnost). Abychom ukázali, že bod P leží na straně BC, spočteme velikost úhlu BPC z tětivových čtyřúhelníků $BPRM$ a $CPRN$:

$$|\sphericalangle BPC| = |\sphericalangle BPR| + |\sphericalangle CPR| = |\sphericalangle AMR| + |\sphericalangle ANR| = 180°,$$

neboť AMR a ANR jsou protější úhly tětivového čtyřúhelníku $AMRN$. Tím je tvrzení úlohy dokázáno.



Obr. 44



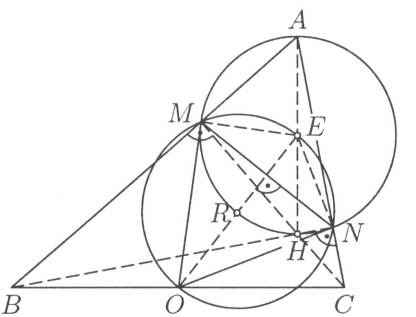
Obr. 45

Jiné řešení. Stejně jako v předchozím řešení ukážeme nejprve, že bod R leží na kružnici opsané trojúhelníku AMN. Označme dále P ten bod strany BC, v němž ji protne osa AR úhlu BAC (obr. 45). Protože čtyřúhelník $AMRN$ je tětivový, je $|\sphericalangle ARM| = |\sphericalangle ANM| = |\sphericalangle ABC|$, což znamená, že i čtyřúhelník $BPRM$ je tětivový. Analogicky ukážeme, že i čtyřúhelník $PCNR$ je tětivový. Bod P na úsečce BC je pak ovšem průsečíkem kružnic opsaných trojúhelníkům BMR a CNR, jak jsme měli dokázat.

Poznámky. V předchozích dvou řešeních jsme nijak nevyužili, že střed kružnice opsané čtyřúhelníku $BCNM$ je zároveň středem strany BC.

Tvrzení ve skutečnosti platí pro libovolnou kružnici s tětivou BC, pokud není opsána trojúhelníku ABC (pak by bylo $M = N = A = R$). Uvedená situace je vlastně speciálním případem tzv. *Miquelovy věty*, když za jeden z bodů zvolíme průsečík osy úhlu s protější stranou: Je-li na každé straně daného trojúhelníku zvolen bod, tři kružnice určené dvojicemi bodů na sousedních stranách a jejich společným vrcholem mají společný bod.

Pokud ovšem předpokládáme, že střed O kružnice opsané čtyřúhelníku $BCNM$ je středem strany BC, vyplývá z prvního řešení, že uvažovaný bod R je středem kružnice vepsané trojúhelníku MNO. V tom případě jsou body M a N patami výšek trojúhelníku ABC. Kružnice nad průměrem AH, kde H je průsečík obou výšek, je tedy kružnicí opsanou trojúhelníku AMN (obr. 46). Paty výšek, středy stran a středy spojnic



Obr. 46

průsečíku výšek s vrcholy leží na tzv. *kružnici devíti bodů* daného trojúhelníku. Pro trojúhelník ABC jsou čtyřmi z těchto devíti bodů body M, N, O a střed E úsečky AH, a protože $|EM| = |EN|$, je to zároveň střed příslušného oblouku MN kružnice opsané trojúhelníku MNO. Ten má však tu vlastnost, že leží jednak na ose úhlu MON, jednak má od středu kružnice vepsané trojúhelníku MNO stejnou vzdálenost jako od obou vrcholů M, N (tuto vlastnost hezky využívá např. řešení 2. úlohy na 43. MMO, viz ročenku 51. ročníku MO), tj. leží na kružnici opsané trojúhelníku AMN. Jak už víme, tímto bodem je bod R.

2. (Podle *Alexandra Kazdy.*) Ukážeme, že řešením jsou jen mnohočleny $P(x) = \alpha_1 x^2 + \alpha_2 x^4$ pro libovolná reálná α_1 a α_2.

Nechť mnohočlen P splňuje podmínky úlohy. Je-li $a = b = 0$, je $ab + bc + ca = 0$ pro každé reálné číslo c. Dostáváme proto

$$P(0-0) + P(0-c) + P(c-0) = 2P(0+0+c),$$

neboli

$$P(0) + P(-c) = P(c)$$

pro libovolné reálné c. Dosazením $c = 0$ dostaneme $P(0) = 0$, takže $P(c) = P(-c)$ pro všechna reálná c. Mnohočlen P je tudíž sudá funkce a musí být tvaru

$$P(x) = a_n x^{2n} + a_{n-1} x^{2n-2} + \ldots + a_1 x^2, \quad a_1, \ldots, a_n \in \mathbb{R}, \ a_n \neq 0.$$

Ukážeme nyní, že stupeň mnohočlenu P je nejvýše 4.

Rovnost $ab+bc+ca = 0$ je homogenní, proto ji s trojicí (a, b, c) splňuje i každá trojice (ta, tb, tc) pro libovolné reálné t. Protože $ab + bc + ca = ab+(a+b)c$, vidíme, že pro $a+b = 1$, $c = -ab$ je $ab+bc+ca = 0$, takže pro libovolná reálná čísla s a t uvedenou rovnost splňuje i trojice $a = st$, $b = (1 - s)t$, $c = -ab = (s^2 - s)t$. Dosazením této trojice do rovnosti ze zadání dostaneme pro všechna reálná s a t rovnost

$$P\left((2s-1)t\right) + P\left((1-s^2)t\right) + P\left((s^2-2s)t\right) = 2P\left((s^2-s+1)t\right).$$

Pro pevné s ji můžeme považovat za rovnost mnohočlenů v proměnné t. Porovnáním vedoucích koeficientů (u mocnin t^{2n}) na obou stranách dostáváme pro všechna reálná s rovnost

$$(2s-1)^{2n} + (1-s^2)^{2n} + (s^2-2s)^{2n} = 2(s^2-s+1)^{2n}. \qquad (1)$$

Porovnejme nyní koeficienty u mocnin s^{4n-2}. Na pravé straně je dle polynomické věty mnohočlen

$$2 \sum_{i+j \leqq 2n} \frac{(2n)!}{i!\, j!\, (2n-i-j)!} (-1)^j s^{2i+j},$$

přičemž $2i + j = 4n - 2$ jedině pro $j = 0$, $i = 2n - 1$ a pro $j = 2$, $i = 2n - 2$ (j musí být sudé a z podmínky $i + j \leqq 2n$ díky rovnosti $i = (2i + j) - (i + j)$ plyne $i \geqq 2n - 2$, tudíž $j \leqq 2$). Zmíněná mocnina má tedy na pravé straně koeficient

$$2\left[\binom{2n}{1} + \binom{2n}{2}\right] = 4n^2 + 2n,$$

zatímco na levé straně rovnosti (1) dostaneme podle binomické věty koeficient

$$\binom{2n}{2-2n} 2^{4n-2} - \binom{2n}{1} + \binom{2n}{2} 2^2 = \begin{cases} 6 & \text{pro } n = 1, \\ 8n^2 - 6n & \text{pro } n \geqq 2. \end{cases}$$

Vidíme, že rovnosti obou koeficientů vyhovuje $n = 1$, a pro $n \geqq 2$ dostáváme rovnici $4n(n-2) = 0$, které vyhovuje $n = 2$. Podmínkám úlohy tak mohou vyhovět jedině mnohočleny tvaru $P(x) = \alpha_1 x^2 + \alpha_2 x^4$ pro reálná α_1 a α_2.

Nyní ukážeme, že každý mnohočlen uvedeného tvaru splňuje podmínky úlohy. Abychom to ověřili, uvědomme si nejdříve, že libovolná lineární kombinace dvou mnohočlenů, jež splňují podmínky úlohy, je rovněž splňuje. Stačí to tedy ověřit pro mnohočleny x^2 a x^4.

To, že vyhovuje x^2, vyplývá z rovnosti

$$(a-b)^2 + (b-c)^2 + (c-a)^2 - 2(a+b+c)^2 = -6(ab+bc+ca).$$

Ověřme požadovanou rovnost i pro jednočlen x^4. Nechť $ab + bc + ca = 0$ a položme $p = a - b$, $q = b - c$ a $r = c - a$. Při ověřování x^2 jsme vlastně ukázali, že

$$p^2 + q^2 + r^2 = 2(a+b+c)^2.$$

Protože $p + q + r = 0$, postupně dostaneme:

$$pq + qr + rp = -\tfrac{1}{2}(p^2 + q^2 + r^2) = -(a+b+c)^2,$$
$$(pq)^2 + (qr)^2 + (rp)^2 = (pq + qr + rp)^2 - 2pqr(p+q+r) = (a+b+c)^4,$$

takže

$$p^4 + q^4 + r^4 = (p^2 + q^2 + r^2)^2 - 2\left((pq)^2 + (qr)^2 + (rp)^2\right) = 2(a+b+c)^4,$$

což je požadovaná rovnost.

Jiné řešení. Vraťme se k rovnosti (1) předchozího řešení. Volbou $s = -2$ vyjde $5^{2n} + 3^{2n} + 8^{2n} = 2 \cdot 7^{2n}$, takže $8^{2n} < 2 \cdot 7^{2n}$. Ale už pro $n = 3$ platí $8^{2n} > 2 \cdot 7^{2n}$ ($8^{2 \cdot 3} = 262\,144 > 235\,298 = 2 \cdot 7^{2 \cdot 3}$), tím spíš to platí pro $n > 3$. Takže $n \leqq 2$, což znamená, že $P(x) = \alpha_1 x^2 + \alpha_2 x^4$ pro reálná α_1 a α_2. Zbývá jen ověřit, že všechny mnohočleny tohoto tvaru úloze vyhovují, což učiníme stejně jako v předchozím řešení.

Jiné řešení. Pro každé reálné t splňuje trojice $(a, b, c) = (6t, 3t, -2t)$ podmínku $ab + bc + ca = 0$. Dosazením do dané rovnosti dostáváme

$$P(3t) + P(5t) + P(-8t) = 2P(7t).$$

Pokud tedy $P(x) = a_n x^n + \ldots + a_1 x + a_0$, platí nutně pro každé $i = 0, 1, 2, \ldots$ rovnost

$$\left(3^i + 5^i + (-8)^i - 2 \cdot 7^i\right) a_i = 0.$$

Výraz v závorkách je záporný pro lichá i a kladný pro $i = 0$ a pro všechna sudá $i \geqq 6$. Jen pro $i = 2$ a $i = 4$ je výraz nulový. Proto musí být $P(x) = \alpha_1 x^2 + \alpha_2 x^4$ pro reálná α_1 a α_2. Zbývá jen ověřit, že všechny mnohočleny tohoto tvaru úloze vyhovují, což učiníme stejně jako v prvním řešení.

Jiné řešení. (Podle Tony Zhanga.) Jak jsme už zjistili v úvodu prvního řešení, je hledaný mnohočlen P sudá funkce s nulovým absolutním členem, je tedy tvaru $P(x) = x^2 f(x^2)$ pro vhodný mnohočlen f. Ukážeme, že jeho stupeň je nejvýše 1.

Daná podmínka má pro mnohočlen f tvar

$$\begin{aligned}&(a-b)^2 f\big((a-b)^2\big) + (b-c)^2 f\big((b-c)^2\big) + \\ &+ (c-a)^2 f\big((c-a)^2\big) = 2(a+b+c)^2 f\big((a+b+c)^2\big). \qquad (2)\end{aligned}$$

Mezi čísly a, b, c, jež splňují rovnost $ab+bc+ca = 0$, najdeme taková, pro něž bude $a-b = b-c$, tj. $a+c = 2b$, takže $0 = (a+c)b+ca = 2b^2+(2b-a)a$. Z této kvadratické rovnice vyjde $a = b \pm b\sqrt{3}$, takže do (2) můžeme dosadit $a = (1-\sqrt{3})b$ a $c = (1+\sqrt{3})b$ a dostaneme

$$6b^2 f(3b^2) + 12b^2 f(12b^2) = 18b^2 f(9b^2),$$

což můžeme přepsat jako

$$12b^2\big(f(12b^2) - f(9b^2)\big) = 6b^2\big(f(9b^2) - f(3b^2)\big),$$

anebo pro $b \neq 0$ jako

$$\frac{f(12b^2) - f(9b^2)}{3b^2} = \frac{f(9b^2) - f(3b^2)}{6b^2}. \qquad (3)$$

Protože $3b^2$ je libovolné kladné číslo, platí podle (3) pro každé reálné $x > 0$ rovnost

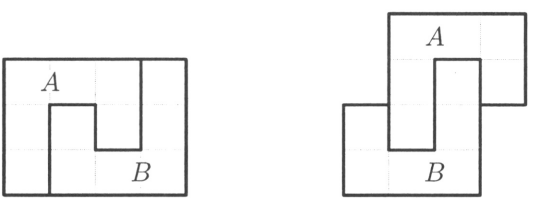
$$\frac{f(4x) - f(3x)}{x} = \frac{f(3x) - f(x)}{2x}. \qquad (4)$$

Obě strany (4) jsou (po zkrácení) mnohočleny proměnné x, takže musí jít o stejné mnohočleny. Je-li mnohočlen f stupně $k \geqq 1$ s vedoucím členem αx^k, jsou obě strany (4) mnohočleny stupně $k-1$ s vedoucími členy

$$\alpha(4^k - 3^k)x^{k-1}, \quad \text{resp.} \quad \frac{\alpha(3^k-1)}{2}x^{k-1}.$$

Porovnáním těchto členů s ohledem na $\alpha \neq 0$ dostaneme po úpravě rovnici $2 \cdot 4^k = 3^{k+1} - 1$, která je splněna pouze pro $k = 1$, neboť pro $k \geqq 2$ máme $2 \cdot 4^k > 3^{k+1}$, jak snadno ověříme indukcí. Mnohočlen f je tudíž nejvýše lineární, $f(x) = \alpha_1 + \alpha_2 x$ a $P(x) = x^2 f(x^2) = \alpha_1 x^2 + \alpha_2 x^4$. Zbývá jen ověřit, že všechny mnohočleny tohoto tvaru úloze vyhovují, což učiníme stejně jako v prvním řešení.

3. Předpokládejme, že pravoúhelník $m \times n$ je pokryt dlaždicemi dle zadání. Ke každé dlaždici v pokrytí přísluší jedno čtvercové pole pravoúhelníku, které dlaždice nepokrývá, které je však zároveň ze tří stran obklopeno jejími čtverci. Takové pole označíme jako „vnitřní pole" dlaždice. Zřejmě tedy ke každé dlaždici A v pokrytí můžeme přiřadit dlaždici B, která pokrývá „vnitřní pole" dlaždice A. Jsou jen dvě možnosti (až na otočení a souměrnost), jak takto dlaždici B k dlaždici A přiložit (obr. 47); v obou případech dlaždice A „recipročně" pokrývá vnitřní pole dlaždice B. To znamená, že v pokrytí jsou všechny dlaždice jednoznačně rozděleny do dvojic, z nichž každá vytváří buď obdélník 3×4, nebo nekonvexní osmiúhelník znázorněný na obr. 47 vpravo. Daný pravoúhelník lze tedy dlaždicemi pokrýt, právě když ho lze pokrýt *dvojútvary* složenými z 12 čtverců na obr. 47.



Obr. 47

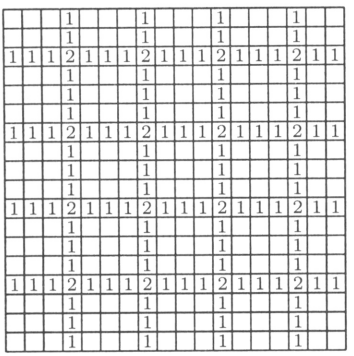
Dále je vidět, že žádná ze stran daného pravoúhelníku nemůže mít délku 1, 2 či 5 čtverců (řádek či sloupec podél takové strany pravoúhelníku nedokážeme žádným způsobem pokrýt).

Naopak je zřejmé, že jen pomocí obdélníků 3×4 dokážeme pokrýt každý pravoúhelník $3a \times 4b$, speciálně tedy i pravoúhelník $12c \times 3$ a $12c \times$ $\times 4$. A protože každé číslo $d \geqq 6$ lze napsat jako součet několika trojek a několika čtyřek, lze pokrýt i každý pravoúhelník $12c \times d$, pokud $d \notin$ $\notin \{1, 2, 5\}$. Ukážeme, že tím jsou všechny možné pravoúhelníky, jež lze danými dlaždicemi pokrýt, vyčerpány.

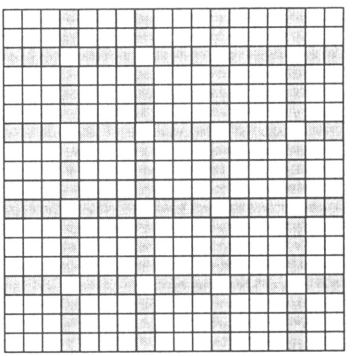
Jestliže pravoúhelník $m \times n$ je pokryt dvojútvary složenými z 12 čtverců, je jeho obsah mn dělitelný dvanácti. Naším jediným úkolem je dokázat, že aspoň jedno z čísel m, n musí být dělitelné čtyřmi. Předpokládejme, že tomu tak není. Protože mn je dělitelné 4, jsou obě čísla m, n sudá. Ukážeme-li, že počet dvojútvarů v pokrytí musí být sudý, bude součin mn dělitelný dvaceti čtyřmi, což odporuje předpokladu, že ani jedno z čísel m, n není čtyřmi dělitelné.

První způsob. Označme v pravoúhelníku čtvercová pole každého čtvrtého sloupce a každého čtvrtého řádku jednotkami, přičemž na pole v prů-

sečíku označených řádků a sloupců místo dvou jednotek napíšeme dvojku (obr. 48 pro $m = n = 18$, řádky počítáme odspodu jako u šachovnice). Protože počet řádků i sloupců pravoúhelníku je sudý, je součet všech čísel v něm sudý. Na druhé straně obdélník 3×4 pokryje čísla se součtem 3 nebo 7, zatímco osmiúhelníkový útvar z obr. 47 pokryje čísla se součtem 5 nebo 7. To znamená, že počet všech dvojútvarů v pokrytí je sudý.

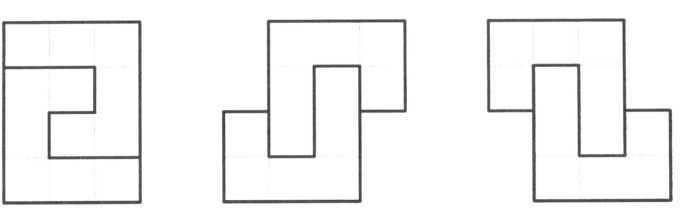


Obr. 48



Obr. 49

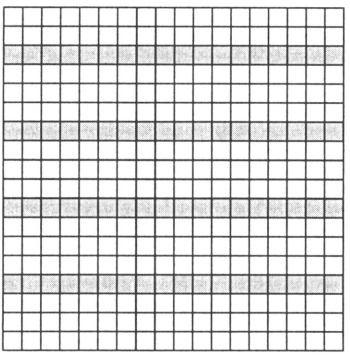
Druhý způsob. Místo čísel obarvíme čtvercová pole v každém čtvrtém sloupci a v každém čtvrtém řádku, přičemž společná pole v jejich průsečíku ponecháme neobarvená (obr. 49). Jestliže $m = 4i + 2$, $n = 4j + 2$, bude celkový počet tmavých polí $i(3j + 2) + j(3i + 2) = 2(3ij + i + j)$, což je sudé číslo. Zároveň není těžké se přesvědčit, že každý dvojútvar pokryje 3 nebo 5 obarvených čtverců. Jejich počet proto musí být sudý.



Obr. 50

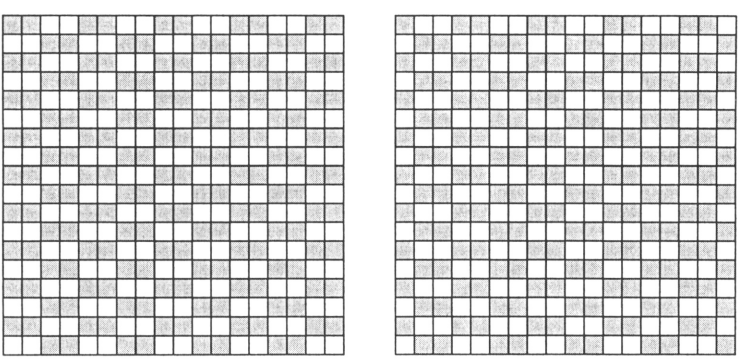
Třetí způsob. Dvojútvary v poloze na obr. 50 můžeme charakterizovat následujícím způsobem: V každém ze čtyř řádků jsou tři čtverce a v každém sloupci je sudý počet čtverců. Jestliže v daném pravoúhelníku obarvíme každý čtvrtý řádek (obr. 51), bude počet obarvených čtverců sudý.

Přitom dvojútvary z obr. 50 (v této poloze) pokryjí každý právě tři obarvené čtverce, zatímco dvojútvary, jež z nich vzniknou otočením o 90°, jich díky uvedené charakterizaci pokryjí sudý počet. Počet dvojútvarů z obr. 50 v daném pokrytí je tedy sudý. Podobně obarvíme-li v daném pravoúhelníku každý čtvrtý sloupec, zjistíme, že i počet dvojútvarů, jež vzniknou z těch na obr. 50 otočením o 90°, je sudý.



Obr. 51

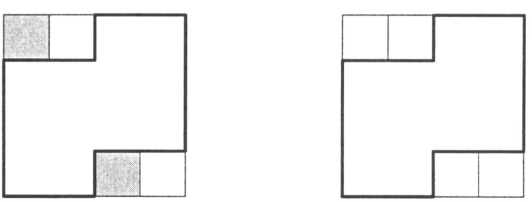
Jiné řešení. Předpokládejme, že existuje pokrytí pravoúhelníku $m \times n$, kde $m \equiv n \equiv 2 \pmod 4$. Pomocí několika různých obarvení ukážeme, že počet všech osmiúhelníkových dvojútvarů musí být sudý i lichý zároveň, což samozřejmě nelze.



Obr. 52

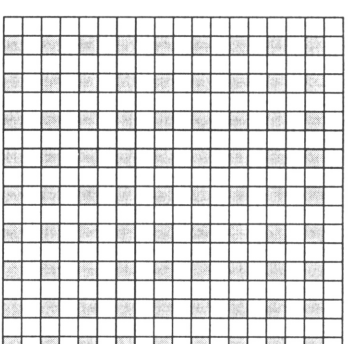
Uvažujme dvě různá obarvení daného pravoúhelníku (obr. 52), v nichž jsou střídavě obarveny obdélníky 2×1, přičemž druhé obarvení se od

prvého liší jen „posunutím“ o jeden sloupec doprava. V každém z nich je stejný počet tmavých i světlých polí (počet řádků je sudý). Zatímco každý z obdélníků 3×4 pokryje čtyři tmavá a čtyři světlá pole, pro osmiúhelníkové dvojútvary to vždy neplatí. Záleží totiž na jejich poloze vůči zvolenému obarvení. Podívejme se na ně jako na čtverce 4×4, z nichž jsme odřízli dva protější rohy 2×1 (orientované stejně jako obdélníky zmíněného obarvení). Oba odříznuté obdélníky zřejmě odkrývají stejně obarvená pole (obr. 53), proto v případě, že mají všechna čtyři pole stejnou barvu, bude se počet tmavých a světlých polí pokrytých takovým dvojútvarem lišit o čtyři. Počet útvarů, které vykazují tuto asymetrii v jednom obarvení, musí tedy být sudý. Totéž platí pro dvojútvary, které vykáží stejnou asymetrii v „posunutém“ obarvení (obě množiny jsou zjevně disjunktní!). Podobně i počet dvojútvarů orientovaných „svisle“, tj. s odříznutými obdélníky 1×2, je sudý (použijeme analogické obarvení, v němž jsou střídavě obarveny „svislé“ obdélníky 1×2).



Obr. 53

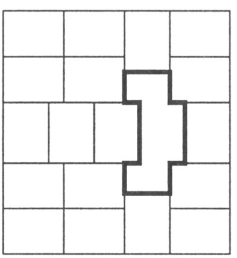
Obarvěme nyní daný pravoúhelník podle obr. 54. Počet tmavých polí je v něm lichý, přitom každý obdélník 3×4 pokryje dvě nebo čtyři tmavá pole, tj. sudý počet, zatímco osmiúhelníkové dvojútvary pokryjí vždy tři



Obr. 54

tmavá pole. Odtud ovšem plyne, že jejich celkový počet musí být lichý. Došli jsme ke sporu.

Poznámka. Tato úloha byla z šestice úloh nejobtížnější. Jedním z důvodů bylo patrně i to, že obvyklý argument s vhodným obarvením polí pravoúhelníku sice funguje, ale jak ukazují předvedená řešení, jeden způsob obarvení často nestačí. Přestože hledaná množina je totožná s množinou všech pravoúhelníků, jež lze pokrýt pouze obdélníky 3×4, není pravda, že by neexistovala pokrytí využívající osmiúhelníkový dvojútvar (obr. 55).



Obr. 55

4. Vzhledem k symetrii stačí ukázat, že za daných předpokladů platí $t_1 < t_2 + t_3$. Podle nerovnosti mezi aritmetickým a geometrickým průměrem je

$$\frac{1}{t_2} + \frac{1}{t_3} \geqq \frac{2}{\sqrt{t_2 t_3}}, \quad t_2 + t_3 \geqq 2\sqrt{t_2 t_3}, \tag{1}$$

$$\frac{t_i}{t_j} + \frac{t_j}{t_i} \geqq 2 \quad \text{pro všechna } i, j.$$

Úpravou pravé strany dané nerovnosti tak dostáváme

$$\begin{aligned} n^2 + 1 &> n + \sum_{1 \leqq i < j \leqq n} \left(\frac{t_i}{t_j} + \frac{t_j}{t_i} \right) = \\ &= n + t_1 \left(\frac{1}{t_2} + \frac{1}{t_3} \right) + \frac{1}{t_1}(t_2 + t_3) + \sum_{\substack{1 \leqq i < j \leqq n \\ (i,j) \neq (1,2),(1,3)}} \left(\frac{t_i}{t_j} + \frac{t_j}{t_i} \right) \geqq \\ &\geqq n + 2\frac{t_1}{\sqrt{t_2 t_3}} + 2\frac{\sqrt{t_2 t_3}}{t_1} + 2\left[\binom{n}{2} - 2 \right] = \\ &= 2a + \frac{2}{a} + n^2 - 4, \end{aligned}$$

kde jsme označili $a = t_1/\sqrt{t_2 t_3}$. Z poslední úpravy vychází nerovnost

$$0 > 2a + \frac{2}{a} - 5 = \frac{1}{a}(2a^2 - 5a + 2) = \frac{1}{a}(2a - 1)(a - 2),$$

odkud plyne $a < 2$. Je tedy $t_1 = a\sqrt{t_2 t_3} < 2\sqrt{t_2 t_3} \leqq t_2 + t_3$ (ještě jednou jsme využili druhou nerovnost v (1)), což jsme chtěli dokázat.

Jiné řešení. (Podle *Arne Smeetse* z Belgie, který získal stříbrnou medaili.) Dokážeme tvrzení sporem, a to nejprve pro $n = 3$. Podle známé nerovnosti mezi aritmetickým a harmonickým průměrem

$$\frac{a+b}{2} \geqq \frac{2}{\frac{1}{a} + \frac{1}{b}} \quad \text{neboli} \quad \frac{1}{a} + \frac{1}{b} \geqq \frac{4}{a+b}$$

pro tři kladná čísla $a = t_1$, $b = t_2$, $c = t_3$ platí

$$(a+b+c)\Big(\frac{1}{a} + \frac{1}{b} + \frac{1}{c}\Big) \geqq (a+b+c)\Big(\frac{4}{a+b} + \frac{1}{c}\Big) = 5 + \frac{4c}{a+b} + \frac{a+b}{c},$$

takže pokud předpokládáme, že $c \geqq a + b$, je podle nerovnosti mezi aritmetickým a geometrickým průměrem

$$(a+b+c)\Big(\frac{1}{a} + \frac{1}{b} + \frac{1}{c}\Big) \geqq 5 + 5\sqrt[5]{\frac{c^4}{(a+b)^4} \cdot \frac{a+b}{c}} \geqq 5 + 5 = 10. \quad (2)$$

To odporuje dané nerovnosti (pro $n = 3$ je $n^2 + 1 = 10$), takže kladná čísla a, b a c splňují trojúhelníkovou nerovnost $c < a + b$.

Pro $n \geqq 4$ označme $a = t_1$, $b = t_2$, $c = t_3$, $S = \sum_{i \geqq 4} t_i$, $T = \sum_{i \geqq 4} 1/t_i$ a předpokládejme opět, že $c \geqq a + b$. Součin na pravé straně dané nerovnosti můžeme pak vyjádřit jako

$$P = (a+b+c)\Big(\frac{1}{a} + \frac{1}{b} + \frac{1}{c}\Big) + S\Big(\frac{1}{a} + \frac{1}{b} + \frac{1}{c}\Big) + T(a+b+c) + ST.$$

Podle Cauchyovy nerovnosti je $ST \geqq (n-3)^2$, podle nerovnosti mezi aritmetickým a geometrickým průměrem a podle (2) pak je

$$S\Big(\frac{1}{a} + \frac{1}{b} + \frac{1}{c}\Big) + T(a+b+c) \geqq 2\sqrt{ST(a+b+c)\Big(\frac{1}{a} + \frac{1}{b} + \frac{1}{c}\Big)} \geqq$$
$$\geqq 2(n-3)\sqrt{10} > 6(n-3).$$

To dohromady dává

$$P \geqq 10 + 2(n-3)\sqrt{10} + (n-3)^2 = n^2 + 1,$$

což opět odporuje dané nerovnosti.

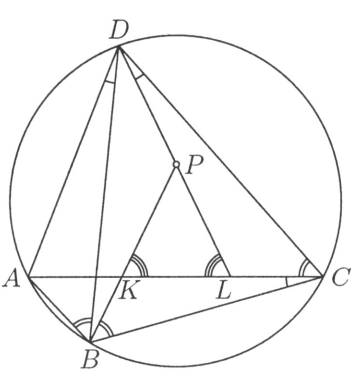
Poznámka. Z předchozího výpočtu je vidět, že místo $n^2 + 1$ mohl být na levé straně předpokládané nerovnosti i výraz

$$10 + 2\sqrt{10}(n-3) + (n-3)^2 = \left(n - 3 + \sqrt{10}\right)^2,$$

který je větší než $n^2 + 1$ pro každé $n \geqq 4$ (pro $n = 3$ se oba výrazy rovnají).

5. Protože vymezení bodu P je symetrické vůči vrcholům B a D, můžeme bez újmy na obecnosti předpokládat, že bod P leží v trojúhelníku ACD. Podobně i podmínka rovnosti příslušných úhlů charakterizující bod P je symetrická vůči vrcholům A a C ($|\sphericalangle PBC| = |\sphericalangle DBA|$, právě když $|\sphericalangle PBA| = |\sphericalangle DBC|$). Můžeme tedy předpokládat, že bod P leží v trojúhelníku BCD.

Předpokládejme nejprve, že čtyřúhelník $ABCD$ je tětivový a označme K, L průsečíky úhlopříčky AC s polopřímkami PB a DP (obr. 56). Z rovnosti příslušných obvodových úhlů nad tětivami AB a AD plyne, že trojúhelníky DAB, DLC a CKB jsou podobné. Z rovnosti vnějších úhlů při vrcholech L a K posledních dvou uvedených trojúhelníků dostáváme, že trojúhelník PKL je rovnoramenný, takže $|PK| = |PL|$.



Obr. 56

Trojúhelníky ADL, BDC jsou také podobné: zřejmě se shodují v úhlu při společném vrcholu D a v obvodových úhlech nad tětivou CD. Využitím podobností $ADL \sim BDC$ a $DAB \sim CKB$ pak můžeme psát

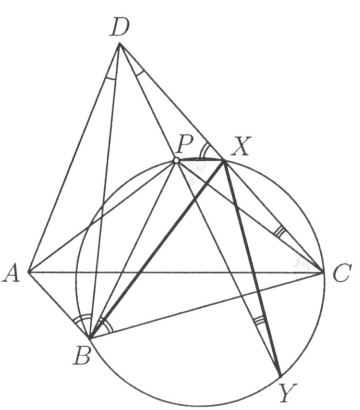
$$\frac{|AL|}{|BC|} = \frac{|AD|}{|BD|} = \frac{|CK|}{|BC|},$$

takže $|AL| = |CK|$. To spolu s rovností $|PK| = |PL|$ dává $|AP| = |CP|$.

Obráceně nechť $|AP| = |CP|$. Sestrojme kružnici opsanou trojúhelníku BCP a označme X, Y další průsečíky této kružnice s polopřímkami DC a DP (obr. 57). Protože $BCXP$ je tětivový, je $|\sphericalangle PXD| = |\sphericalangle PBC|$, takže trojúhelníky PDX a ADB jsou podobné. Z této podobnosti navíc plyne spirální podobnost trojúhelníků $ADP \sim BDX$. Nakonec z rovnosti obvodových úhlů nad tětivou PX dostáváme podobnost $DPC \sim DXY$. Využitím posledních dvou podobností pak máme

$$\frac{|AP|}{|BX|} = \frac{|DP|}{|DX|} = \frac{|PC|}{|XY|},$$

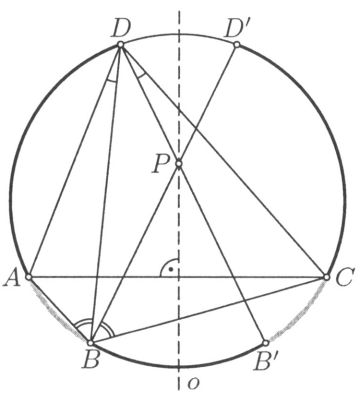
takže $|BX| = |XY|$. V uvažované kružnici jsme našli shodné tětivy se společným krajním bodem X). Tětivě BX přísluší obvodový úhel BCD, a protože ve zvolené konfiguraci je $|\sphericalangle BCD| < |\sphericalangle YCD|$, shoduje se s obvodovým úhlem XPY nad tětivou YX, který je vnějším úhlem při vrcholu P trojúhelníku PDX shodným s vnějším úhlem při vrcholu A trojúhelníku ADB. Platí tedy $|\sphericalangle BCD| = 180° - |\sphericalangle BAD|$ a čtyřúhelník $ABCD$ je tětivový, jak jsme chtěli dokázat.



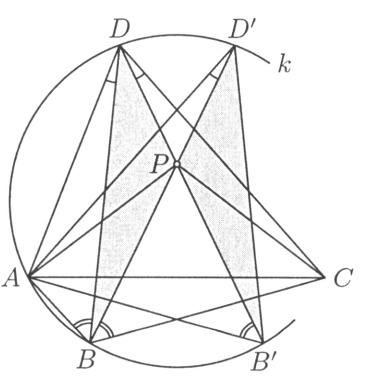
Obr. 57

Jiné řešení. (Podle *Františka Konopeckého.*) Vzhledem k tomu, že úhlopříčka BD není osou ani jednoho z vnitřních úhlů ABC, CDA, shodnosti úhlů určující polohu bodu P znamenají, že bod P nemůže ležet na úhlopříčce BD.

Předpokládejme nejprve, že $ABCD$ je tětivový. Označme B' a D' průsečíky opsané mu kružnice s polopřímkami opačnými k PD, resp. k PB. Shodnost úhlů ABD a PBC (obr. 58) tak znamená shodnost příslušných oblouků AD a CD'. Podobně se shodují i oblouky AB a CB'. To ale znamená, že bod B' je obrazem bodu B a bod D' obrazem bodu D v osové souměrnosti podle osy o úsečky AC. V této osové souměrnosti je tak úsečka BD' obrazem úsečky $B'D$, a jejich průsečík P proto leží na ose úhlopříčky AC. Tudíž $|AP| = |CP|$.



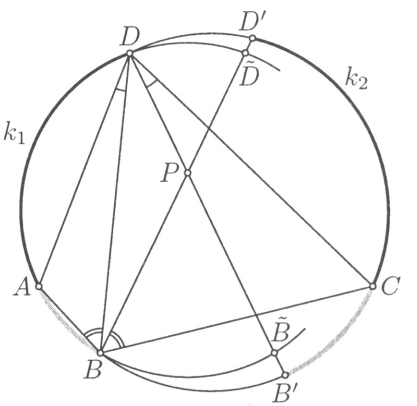
Obr. 58



Obr. 59

Obráceně nechť $|AP| = |CP|$. Uvažujme kružnici k opsanou trojúhelníku ABD a označme B' a D' její průsečíky s polopřímkami opačnými k PD, resp. k PB (obr. 59). Ze shodnosti obvodových úhlů nad tětivou BB' resp. DD' vyplývá, že trojúhelníky BPD a $B'PD'$ jsou podobné podle věty *uu*. Zmíněné shodnosti navíc znamenají, že i trojuhelniky BDC a $B'D'A$ se shodují ve dvojicích úhlů při stranách BD a $B'D'$, takže jsou podobné se stejným poměrem podobnosti jako trojúhelníky BPD a $B'PD'$. A protože v této podobnosti si odpovídají dvě shodné úsečky CP a AP, jedná se o shodnost (snadno nahlédneme, že se jedná o osovou souměrnost). Tato shodnost převádí kružnici k opsanou trojúhelníku ABD na sebe, proto bod C, který je obrazem bodu A, leží rovněž na této kružnici, a čtyřúhelník $ABCD$ je tedy tětivový.

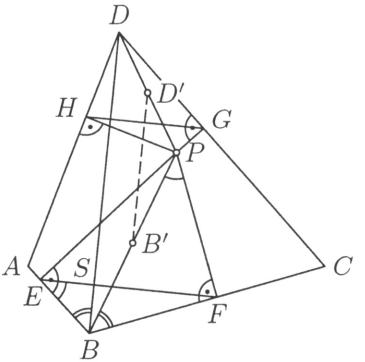
Jiné řešení. Dokážeme obrácenou implikaci trochu jinak než v předchozím řešení. Předpokládejme, že $|AP| = |CP|$ a označme k_1 kružnici opsanou trojúhelníku ABD a k_2 kružnici opsanou trojúhelníku BCD. Označme dále $\tilde{B}$, $\tilde{D}$ průsečíky polopřímek DP a BP s kružnicí k_1 a B', D' s kružnicí k_2 (obr. 60). Z rovnosti úhlů $|\sphericalangle PBC| = |\sphericalangle DBA|$ plyne, že



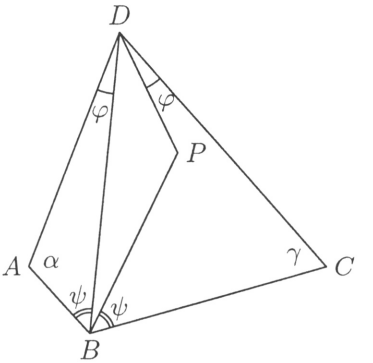
Obr. 60

příslušné oblouky DA kružnice k_1 a $D'C$ kružnice k_2 mají stejnou délku v obloukové míře, a podobně díky rovnosti $|\sphericalangle PDC| = |\sphericalangle BDA|$ mají stejnou délku i oblouky AB kružnice k_1 a $B'C$ kružnice k_2. Existuje tedy nepřímá podobnost f, jež zobrazuje kružnici k_1 na k_2, přičemž bodům D, A, B v této podobnosti postupně odpovídají body D', C a B' a dále bodům $\tilde{B}$, $\tilde{D}$ pak body B a D (i odpovídající oblouky $B\tilde{B}$, $B'B$ a $D\tilde{D}$, DD' mají očividně stejnou obloukovou míru). Protože bod P je průsečíkem přímek $B\tilde{D}$ a $\tilde{B}D$, zobrazí se podobností f na průsečík přímek $B'D$ a BD', což je opět bod P. A protože $|AP| = |CP| = |f(A)f(P)|$, je poměr podobnosti f roven 1, takže je také $|PB'| = |f(P)f(B)| = |PB|$ a podobně i $|PB| = |P\tilde{B}|$. Body B' a $\tilde{B}$ tak nutně splývají a kružnice k_1, k_2 jsou totožné.

Jiné řešení. Označme po řadě E, F, G, H kolmé průměty bodu P na jednotlivé strany AB, BC, CD a DA daného čtyřúhelníku. Čtyřúhelník $EBFP$ je zřejmě tětivový, takže $|\sphericalangle BEF| = |\sphericalangle BPF|$. Označíme-li S průsečík příčky EF s úhlopříčkou BD (obr. 61), vidíme, že trojúhelníky EBS, PBF jsou podobné, neboť dle předpokladu $|\sphericalangle EBS| = |\sphericalangle FBP|$. To znamená, že $|\sphericalangle BSE| = |\sphericalangle BFP| = 90°$, neboli příčka EF je kolmá na úhlopříčku BD. Analogicky zjistíme, že i příčka GH je kolmá na BD.



Obr. 61



Obr. 62

Označíme-li B' střed úsečky PB a D' střed úsečky PD, bude $B'D'$ střední příčkou trojúhelníku BPD, takže úsečka $B'D'$ je stejně jako úsečka BD kolmá na obě příčky EF a GH. Body B' a D' jsou ovšem zároveň středy příslušných Thaletových kružnic nad průměry PB a PD, takže $B'D'$ je osou obou příček EF i GH, $EFGH$ je tudíž rovnoramenný lichoběžník se základnami EF, GH. Z dalších dvou tětivových čtyřúhelníků $EAHP$ a $FCGP$ tak dostáváme následující rovnosti:

$$|AP| \cdot \sin|\sphericalangle BAD| = |EH| = |FG| = |CP| \cdot \sin|\sphericalangle BCD|.$$

Rovnost $|AP| = |CP|$ je tudíž ekvivalentní rovnosti $\sin|\sphericalangle BAD| = \sin|\sphericalangle BCD|$. Vzhledem k podmínkám úlohy nemůže bod P ležet na úhlopříčce BD. Označíme-li pro jednoduchost α a γ úhly při vrcholech A, C a φ, ψ oba úhly vystupující v definici bodu P (obr. 62), dostáváme pro velikost úhlu BPD ve čtyřúhelníku $BCDP$

$$\begin{aligned}|\sphericalangle BPD| &= |\sphericalangle CPB| + |\sphericalangle CPD| = 360^\circ - \gamma - (\varphi + \psi) = \\ &= 180^\circ - \gamma + \alpha.\end{aligned}$$

Vidíme tedy, že za daných předpokladů nemůže být $\alpha = \gamma$, takže rovnost $|AP| = |CP|$ je ekvivalentní rovnosti $\alpha + \gamma = 180^\circ$, což je ekvivalentní tomu, že čtyřúhelník $ABCD$ je tětivový.

6. Pruhovaných čísel je mnoho, obecně je však těžko dokážeme popsat tak, aby bylo vidět, čím jsou dělitelná. Zaměřme se proto na poměrně úzkou skupinu pruhovaných čísel obsahujících jen nuly a jedničky, o nichž dokážeme zjistit víc. Přesněji řečeno, budeme se snažit pro dané číslo n

vytvořit pruhované číslo tvaru $s_k = 1010\ldots101$, kde k je počet jedniček v jeho desítkovém zápisu, které bude násobkem n. Mezi pruhovanými čísly $s_1, s_2, \ldots, s_{n+1}$ najdeme díky Dirichletovu principu určitě dvě různá, která dávají při dělení číslem n stejný zbytek, takže jejich rozdíl je násobkem n. Přitom pro $k > l$ je

$$s_k - s_l = \underbrace{1010\ldots101}_{k \text{ jedniček}} - \underbrace{1010\ldots101}_{l \text{ jedniček}} = \underbrace{1010\ldots10100\ldots00}_{k-l \text{ jedniček} \quad 2l \text{ nul}} = s_{k-l} \cdot 10^{2l}.$$

Pro každé n umíme tedy najít k a l tak, že $n \mid s_{k-l} \cdot 10^{2l}$. Pokud je číslo n s číslem 10 nesoudělné, tak dokonce $n \mid s_{k-l}$, tudíž n má pruhovaný násobek.

Vidíme, že problém je s čísly n, která jsou sudá nebo dělitelná pěti. Pro ně musíme pruhované násobky hledat v jiném tvaru.

Pokusme se je nejprve najít pro čísla n, jež jsou mocninou čísla 5, tj. pro čísla $n = 5^\alpha$. Pro malé hodnoty α snadno nacházíme, že čísla

$$5,\ \ 25 = 5^2,\ \ 125 = 5^3,\ \ 8\,125 = 13 \cdot 5^4,\ \ 78\,125 = 25 \cdot 5^5 \qquad (1)$$

jsou pruhované násobky mocnin 5^1, 5^2, 5^3, 5^4 a 5^5. Pro větší α vytvoříme pruhovaný násobek čísla 5^α indukcí. Předpokládejme, že máme pruhovaný násobek A_k čísla 5^k, který má k číslic, přičemž první zleva může být i nula, což nijak nevadí, naopak je to pro náš postup výhodné, když nemusíme tuto „zbytečnou“ číslici brát do úvahy při rozhodování o pruhovanosti daného čísla, a vytvořme pruhovaný násobek čísla 5^{k+1} tak, že na začátek A_k připíšeme nějakou vhodnou číslici. Je-li tedy

$$A_k = \overline{a_k a_{k-1} \ldots a_1} = 5^k \cdot d, \quad d \in \mathbb{N},\ a_1, a_2, \ldots, a_k \in \{0, 1, 2, \ldots, 9\},$$

takový pruhovaný násobek, připojením číslice a_{k+1} na jeho začátek dostaneme

$$A_{k+1} = \overline{a_{k+1} A_k} = a_{k+1} \cdot 10^k + A_k = 10^k a_{k+1} + 5^k d = 5^k (2^k a_{k+1} + d).$$

Aby A_{k+1} bylo pruhované a zároveň násobek 5^{k+1}, stačí a_{k+1} zvolit tak, aby mělo opačnou paritu než a_k a aby $2^k a_{k+1} + d$ bylo dělitelné pěti. To zřejmě jde vždy, protože pro volbu a_{k+1} máme 5 různých možností (podle parity první číslice čísla A_k buď číslice 1, 3, 5, 7, 9, anebo 0, 2, 4, 6, 8) a pro každou z nich dává číslice a_{k+1}, a tedy i číslo $2^k a_{k+1} + d$ jiný zbytek při dělení pěti (čísla 2^k a 5 jsou nesoudělná). Jeden z těch

zbytků tedy musí být nulový a v tom případě je $2^k a_{k+1} + d$ pěti dělitelné. Ukázali jsme, že všechny mocniny pěti mají pruhované násobky. Přitom z uvedeného postupu vyplývá, že pro dané $n = 5^\alpha$ umíme pruhovaný násobek vytvořit tak, aby měl sudý počet číslic (včetně případné nuly na začátku) a končil číslicí 5.

Věnujme se teď mocninám čísla dvě. Opět ukážeme, že pro každé $n = 2^\beta$ existuje jeho pruhovaný násobek. Postup bude obdobný jako při mocninách pěti, budeme však přidávat až dvě číslice a na vytvářená čísla budeme klást přísnější požadavky. Přesněji, dokážeme, že pro každé přirozené k existuje $(2k-1)$-ciferné pruhované číslo B_k, které je dělitelné číslem 2^{2k-1}, ale není dělitelné číslem 2^{2k} a jehož všechny sudé číslice jsou dvojky. Pro první krok indukce máme pruhovaná čísla

$$B_1 = 2, \quad B_2 = 232 = 2^3 \cdot 29, \quad B_3 = 27\,232 = 2^5 \cdot 851,$$
$$B_4 = 2\,127\,232 = 2^7 \cdot 16\,619.$$

Předpokládejme tedy, že máme pruhované číslo

$$B_k = \overline{b_{2k-1} b_{2k-2} \dots b_1} = 2^{2k-1} \cdot d,$$
$$d \in \mathbb{N} \text{ liché }, b_{2i} \in \{1,3,5,7\} \text{ a } b_{2i-1} = 2 \text{ pro } 1 \leqq i \leqq k$$

(protože B_k je pruhované a sudé, je i číslice b_{2k-1} sudá). Chceme najít (dvojmístné) číslo $b = \overline{2b_{2k}}$ ($b_{2k} \in \{1,3,5,7\}$) tak, aby

$$B_{k+1} = \overline{bB_k} = b \cdot 10^{2k-1} + B_k = 10^{2k-1} b + 2^{2k-1} d = 2^{2k-1}(5^{2k-1} b + d)$$

bylo dělitelné číslem 2^{2k+1}, ale nebylo dělitelné číslem 2^{2k+2}. Potřebujeme tedy, aby $5^{2k-1}b + d$ bylo dělitelné čtyřmi, ale nebylo dělitelné osmi. Podle předpokladu je d liché, dává tedy při dělení osmi jeden ze zbytků 1, 3, 5 nebo 7. Za b proto stačí zvolit jedno z čísel 21, 23, 25 anebo 27 tak, aby $5^{2k-1}b$ dávalo při dělení osmi takový zbytek, který po přičtení d dá zbytek 4 (tj. dává-li d zbytek 1, volíme b tak, aby $5^{2k-1}b$ dávalo zbytek 3, dává-li 3, chceme zbytek 1, pro zbytek 5 zbytek 7 a pro zbytek 7 zbytek 5). Protože 5^{2k-1} je nesoudělné s 8 a čísla 21, 23, 25 a 27 dávají při dělení osmi různé liché zbytky, dostaneme pro jednu z hodnot $b \in \{21, 23, 25, 27\}$ při dělení čísla $5^{2k-1}b$ osmi potřebný zbytek. Našli jsme tedy pruhované násobky i pro mocniny dvou. Přitom když před B_k připíšeme libovolnou lichou číslici, dostaneme pruhované číslo, které je rovněž násobkem 2^{2k-1}, protože B_k má $2k-1$ číslic (přičítáme $b \cdot 10^{2k-1}$). Umíme tedy ke každému $n = 2^\beta$ najít pruhované číslo, které má sudý počet číslic.

Takto vyzbrojeni můžeme přejít k obecnému případu, kdy $n = 5^\alpha \cdot 2^\beta \cdot m$, přičemž m není dělitelné dvěma ani pěti. Pro $\alpha = \beta = 0$ jsme už úlohu vyřešili (číslo n má pruhovaný násobek z nul a jedniček).

Uvažujme případ, kdy $\beta = 0$. Číslo 5^α má pruhovaný násobek se sudým počtem číslic, označme ho M. Zřejmě i číslo

$$S_k = \underbrace{\overline{MM\ldots M}}_{k \text{ čísel } M}$$

je pruhovaným násobkem 5^α. Stejnou úvahou jako na začátku najdeme k a l tak, že $m \mid S_{k-l}$. Pak je S_{k-l} pruhovaným násobkem n.

Úplně stejně najdeme pruhovaný násobek n i v případě, kdy $\alpha = 0$.

Je-li $\beta = 1$ a $\alpha \geqq 1$, stačí k číslu S_{k-l}, které jsme našli pro $n = 5^\alpha \cdot m$, připsat zprava nulu. Dostaneme pruhované číslo (M, a tedy i S_{k-l} končilo číslicí 5), které bude násobkem dvou i čísla $5^\alpha \cdot m$, tj. bude násobkem $5^\alpha \cdot 2^1 \cdot m$.

Zůstal případ, kdy $\beta \geqq 2$ a $\alpha \geqq 1$. V takovém případě je ale $n = 5^\alpha \cdot 2^\beta \cdot m$ násobkem čísla 20, jehož každý násobek končí některým z dvojčíslí 00, 20, 40, 60, 80, takže není nikdy pruhovaný.

Odpověď. Hledanými čísly jsou všechna čísla, která nejsou násobkem čísla 20.

Jiné řešení. (Podle *Tiankai Liu* (USA), který získal zlatou medaili.) Protože každý násobek 20 končí dvěma sudými číslicemi, nemohou mezi hledaná čísla patřit násobky 20. Ukážeme, že každé jiné přirozené číslo má pruhovaný násobek. Protože dělitel takového čísla má stejnou vlastnost, můžeme dále předpokládat, že n je sudé.

Nejprve ukážeme, že pro čísla n tvaru $n = 2^\alpha$ nebo $n = 2 \cdot 5^\alpha$ ($\alpha \geqq 1$) existuje pruhovaný násobek $X(n)$, který má n číslic. Položme (pro n sudé)

$$M = \frac{10^{n+1} - 10}{99} = \underbrace{\overline{101010\ldots10}}_{n \text{ číslic}}.$$

To je zřejmě pruhované číslo a pruhované bude i číslo

$$M + \sum_{i=0}^{k} e_i \cdot 10^i,$$

kde e_i jsou sudé číslice a $k \leqq n - 1$. Navíc poměrně snadno ověříme, že podle toho, zda bylo $n = 2^\alpha$ nebo $2 \cdot 5^\alpha$, dokážeme vybrat posloupnost

číslic $e_0, e_1, \ldots, e_{n-1} \in \{0, 2, 4, 6, 8\}$ tak, že pro každé $k \leqq n-1$ je výše uvedené číslo dělitelné číslem 2^{k+2}, resp. $2 \cdot 5^{k+1}$.

Krátce naznačíme, jak lze při výběru vhodných číslic e_i postupovat např. pro $n = 2 \cdot 5^\alpha$ (zajímá nás samozřejmě jen dělitelnost číslem 5^{k+1}, protože se jedná vesměs o čísla sudá). Protože čísla 0, 2, 4, 6, 8 tvoří úplnou soustavu zbytků modulo 5, snadno určíme e_0 tak, aby bylo $M + e_0 \equiv 0 \pmod 5$; to znamená, že modulo 5^2 dává $M + e_0$ některý ze zbytků 0, 5, 10, 15, 20. Z čísel 0, 2, 4, 6, 8, jež dávají modulo 5 všechny možné zbytky 0, 1, 2, 3 a 4, tedy dokážeme vybrat e_1 tak, že číslo $10e_1$ bude mít modulo 5^2 zbytek opačný, tj. $M + e_0 + 10e_1 \equiv 0 \pmod{5^2}$. A tak pokračujeme dále. Podobně postupujeme i pro $n = 2^\alpha$, kde samozřejmě vystačíme s volbou $e_i \in \{0, 2\}$.

Speciálně tedy umíme pro každé uvažované n najít číslice e_0, $e_1, \ldots, e_{n-1} \in \{0, 2, 4, 6, 8\}$ tak, že číslo

$$X(n) = M + \sum_{i=0}^{n-1} e_i \cdot 10^i$$

bude pruhované a n-místné.

Je-li n obecně sudé číslo, které není dělitelné dvaceti, můžeme je zapsat ve tvaru $n'm$, kde $n' = 2^\alpha$ nebo $n' = 2 \cdot 5^\alpha$, přičemž m je nesoudělné s 10. Vezměme $N \geqq n'$ takové, že $10^N \equiv 1 \pmod m$ (takové N existuje, protože $10^{\varphi(m)} \equiv 1 \pmod m$, kde φ je tzv. Eulerova funkce, takže pro dané m za N stačí vzít dostatečně velký násobek čísla $\varphi(m)$). Nechť

$$M = \frac{10^{2mN+1} - 10}{99} \cdot 10^{n'} + X(n') = \overline{\underbrace{101010\ldots10}_{2mN \text{ číslic}} X(n')}.$$

Protože existuje $k \in \{0, 1, 2, \ldots, m-1\}$, pro něž $M \equiv -2k \pmod m$, je číslo[6]

$$X(n) = M + \sum_{i=1}^{k} 2 \cdot 10^{Ni}$$

nejen pruhované, ale i dělitelné m a samozřejmě i číslem n', neboť jak $n' = 2^\alpha$, tak i $n' = 2 \cdot 5^\alpha$ dělí 10^α, takže dělí i $10^{n'}$ a 10^N. Takto zvolené $X(n)$ je tedy hledaným pruhovaným násobkem čísla n.

[6] Méně zkušeného čtenáře upozorňujeme, že při prázdné množině sčítacích indexů považujeme příslušnou sumu za nulu; taková situace v následující formuli nastane pro $k = 0$, kdy skutečně nepotřebujeme nic přidávat.

Jedenáctý ročník Středoevropské olympiády v informatice

Hostitelem jedenácté Středoevropské olympiády v informatice CEOI 2004 bylo Polsko. Soutěž se uskutečnila ve dnech 12.–18. 7. 2004 na Univerzitě informačních technologií a managementu v Rzeszowě. Olympiáda probíhala v době prázdnin, takže na soutěž mohly být využity univerzitní počítačové laboratoře a na zahajovací a závěrečný ceremoniál i na průběžná jednání volné posluchárny. Ubytování a stravování bylo zajištěno v hotelu nedaleko univerzity.

Olympiáda byla skvěle připravena po stránce organizační, po stránce počítačového vybavení i z hlediska kvality přípravy soutěžních úloh. V každém ze dvou soutěžních dní studenti řešili u počítačů tři náročné příklady. Se svými počítači i soutěžním prostředím se přitom všichni mohli seznámit den před vlastní soutěží, kdy probíhalo tréninkové předkolo. Při soutěži bylo možné programovat v některém z programovacích jazyků Pascal, C nebo C++, každý si mohl zvolit podle svých zkušeností pracovní prostředí operačních systémů Windows nebo Linux. O oba uvedené systémy byl mezi účastníky přibližně stejný zájem. Současně s klasickou soutěží probíhala i soutěž po Internetu pro veřejnost.

K testování a hodnocení vytvořených programů se na CEOI již několik let používá automatický vyhodnocovací systém testující programy na připravené sadě vstupních dat. Všechny prováděné testy mají dobu výpočtu omezenu předem známým časovým limitem a jednotlivá testovací vstupní data mají různou velikost a různou složitost, což dohromady umožňuje bodově rozlišit programy podle kvality použitého algoritmu. Za každou úlohu šlo získat maximálně 100 bodů, nejčastěji bylo zadáno 10 sad testovacích dat po 10 bodech. Některé sady se skládaly z několika vstupů, což umožňovalo vyloučit jednoduchá řešení, která vždy vypíší stejnou odpověď.

Jedenácté středoevropské olympiády v informatice se zúčastnilo 40 soutěžících studentů z 8 zemí střední Evropy. Z Polska se účastnily tři týmy, z nichž se ale do celkových výsledků započítával pouze jeden, z ostatních států se účastnil vždy jeden tým. Českou republiku reprezentovalo čtyřčlenné družstvo ve složení *Ondřej Bílka* (student gymnázia

Lesní čtvrť ve Zlíně), *Jan Bulánek* (student gymnázia J. Vrchlického v Klatovech), *Martin Dobroucký* (student gymnázia v Moravské Třebové) a *Daniel Marek* (student gymnázia Ch. Dopplera v Praze). Naši soutěžící byli vybráni na základě výsledků dosažených v celostátním kole 53. ročníku Matematické olympiády – kategorie P (programování). Soutěže se měl původně místo Jana Bulánka účastnit Oto Petřík, který ovšem před soutěží onemocněl. Jan Bulánek byl proto pozván jako první náhradník. Vedením družstva byli pověřeni Mgr. *Jan Kára* a *Milan Straka*, oba z Matematicko-fyzikální fakulty Univerzity Karlovy v Praze.

Na středoevropské olympiádě v informatice se udělují ocenění podle podobného klíče, jaký se používá například na mezinárodní olympiádě v informatice. Nejvýše polovina soutěžících obdrží některou z medailí, přičemž zlaté, stříbrné a bronzové medaile se dělí přibližně v poměru 1 : 2 : 3. Na letošní CEOI 2004 bylo rozděleno celkem 19 medailí, z toho 4 zlaté, 5 stříbrných a 10 bronzových. O čtyři zlaté medaile se letos netradičně podělily pouze dva státy – Chorvatsko (1. a 3. místo) a Polsko (2. a 4. místo).

Naši studenti letos získali pouze jednu bronzovou medaili, což je částečně způsobeno tím, že na středoevropskou olympiádu jsou vysíláni mladí studenti, aby získali zkušenosti, které v následujících letech mohou zúročit na olympiádě mezinárodní. Také slovenská reprezentace získala jen dvě bronzové medaile. Následující tabulka shrnuje výsledky všech českých studentů v soutěži:

19.	Daniel Marek	107 bodů	bronzová
29.–33.	Ondřej Bílka	60 bodů	–
29.–33.	Jan Bulánek	60 bodů	–
29.–33.	Martin Dobroucký	60 bodů	–

Příští, v pořadí dvanáctá, středoevropská olympiáda v informatice CEOI 2005 se uskuteční ve městě Sarospatak v Maďarsku v první polovině července 2005.

16. mezinárodní olympiáda v informatice

Ve dnech 11.–18. 9. 2004 se konala v Athénách v Řecku 16. mezinárodní olympiáda v informatice IOI 2004 (International Olympiad in Informatics). Soutěže se zúčastnilo 300 studentů ze 77 zemí celého světa. Česká národní delegace byla sestavena na základě výsledků celostátního kola 53. ročníku Matematické olympiády v kategorii P (programování). Naše družstvo mělo následující složení:

Tomáš Gavenčiak, absolvent gymnázia M. Koperníka v Bílovci,
Daniel Marek, student gymnázia Ch. Dopplera v Praze 5,
Petr Škoda, absolvent gymnázia v Praze 8, Ústavní,
Martin Vejnár, student gymnázia na tř. Kpt. Jaroše v Brně.

Vedoucími české delegace byli doc. RNDr. *Pavel Töpfer*, CSc. a Mgr. *Martin Mareš*, oba z Univerzity Karlovy v Praze, Matematicko-fyzikální fakulty.

Zvláštností letošního ročníku Mezinárodní olympiády v informatice byla její přímá návaznost na sportovní letní olympijské hry. Soutěž se konala jen asi dva týdny po skončení olympijských her v objektu nově vybudovaném jako dočasné sídlo sportovních novinářů v těsném sousedství hlavního olympijského stadionu. Ihned po skončení IOI v Athénách pro změnu začínala paralympiáda tělesně postižených sportovců.

Mezinárodní olympiáda v informatice byla organizátory výborně zajištěna. Po stránce odborné připravili pořadatelé soutěže kvalitní úlohy přiměřené obtížnosti, z hlediska celkového pobytu pak vhodným způsobem doplnili odborný program dvěma zajímavými výlety pro všechny účastníky — mezi oběma soutěžními dny se konal půldenní výlet na Akropolis, po skončení soutěže pak celodenní plavba lodí mezi řeckými ostrovy.

Vlastní soutěž probíhala jako vždy ve dvou soutěžních dnech. V každém z nich dostali studenti zadány vždy tři soutěžní úlohy a na jejich vyřešení 5 hodin času. Po tuto dobu mohli pracovat na přiděleném osobním počítači vybaveném základním systémovým prostředím a překladači

programovacích jazyků Pascal, C a C++. Cílem každé úlohy je zvládnout zadaný problém algoritmicky a navržený algoritmus pak také naprogramovat. Vytvořené programy se testují pomocí předem připravené sady vstupních dat, takže práce musí být skutečně dovedena až do podoby spolehlivě odladěného programu. Všechny testy jsou navíc vázány na předem známé časové limity. Tím je zajištěno, že program založený na méně efektivním algoritmu stihne včas doběhnout jen pro malá vstupní data, zatímco při výpočtu s velkými daty je běh programu předčasně přerušen, aniž by se program dobral k výsledku a jeho autor obdržel za příslušný test body. Jednotlivá testovací data (zpravidla se používá 20 sad dat hodnocených po 5 bodech) se liší ve své velikosti i složitosti, takže teoreticky správný, ale pomalý program získá ve výsledném hodnocení jen část z celkového dosažitelného množství bodů.

Za každou z šesti soutěžních úloh bylo možné získat nejvýše 100 bodů, celkově tedy až 600 bodů. Skutečné bodové zisky většiny účastníků byly ale samozřejmě výrazně nižší. Podle počtu dosažených bodů bylo stanoveno výsledné pořadí. Lepší polovina účastníků olympiády byla oceněna medailemi, přičemž zlaté, stříbrné a bronzové medaile se rozdělují přibližně v poměru 1 : 2 : 3. Celkem bylo letos uděleno 26 zlatých, 49 stříbrných a 71 bronzových medailí.

Naši studenti se podobně jako v minulých letech umístili mezi úspěšnějšími týmy, i když dosažené výsledky jsou letos o něco horší než loni:

67.–72.	Petr Škoda	370 bodů	stříbrná medaile
73.–75.	Daniel Marek	365 bodů	stříbrná medaile
	Martin Vejnár	220 bodů	–
	Tomáš Gavenčiak	145 bodů	–

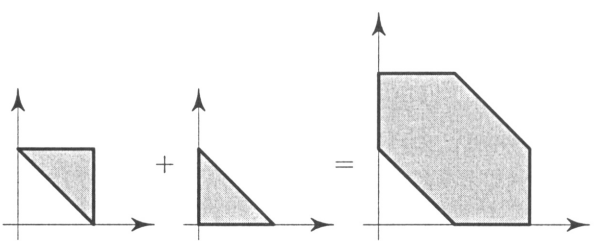
Mezinárodní olympiáda v informatice je soutěží jednotlivců a žádné oficiální pořadí zúčastněných zemí se v ní nevyhlašuje. Podle dosažených výsledků bychom se však řadili přibližně na 25.–30. místo v celkovém pořadí zemí. Nejúspěšnějšími zeměmi byly letos Čína a Rusko se čtyřmi zlatými medailemi, další v pořadí jsou USA, Polsko, Írán, Rumunsko, Bulharsko, Slovensko, Lotyšsko a Tchaj-wan.

Příští, v pořadí již 17. mezinárodní olympiáda v informatice IOI 2005 se uskuteční v polském městě Nowy Sacz nedaleko Krakowa ve dnech 18.–25. 8. 2005.

1. Polygon

Polygon, čili též mnohoúhelník jistě všichni znáte. Ujasněme si jen, že za jeho body považujeme jak body ležící na hranici, tak body uvnitř. Polygon je konvexní právě tehdy, když pro libovolné dva jeho body X a Y platí, že všechny body úsečky XY jsou součástí polygonu. Každý polygon v této úloze bude konvexní, bude mít alespoň dva vrcholy, přičemž všechny jeho vrcholy budou navzájem různé a budou mít celočíselné souřadnice. Žádné tři vrcholy polygonu neleží na jedné přímce.

Minkowského součet polygonů A a B se skládá ze všech bodů ve tvaru $(x_1 + x_2, y_1 + y_2)$, kde (x_1, y_1) je bod polygonu A a (x_2, y_2) je bod polygonu B. Laskavý, ale trpělivý čtenář si může snadno dokázat, že Minkowského součet dvou polygonů je opět polygon. Následující obrázek ukazuje příklad dvou trojúhelníků a jejich součtu:



My se zaměříme na inverzní operaci k Minkowského součtu. Pro zadaný polygon P budeme hledat dva polygony A a B takové, že:

- P je Minkowského součtem polygonů A a B,
- A má 2 až 4 vrcholy, čili je to buďto úsečka (2 vrcholy), trojúhelník (3 vrcholy), nebo čtyřúhelník (4 vrcholy),
- A má největší možný počet vrcholů, čili:
  - A musí být čtyřúhelník, pokud to je možné;
  - jinak to musí být trojúhelník;
  - když není možné ani to, musí to být úsečka.

Je zřejmé, že ani A, ani B nemůže být roven P, neboť druhý sčítanec by musel být bod, což není polygon.

Dostanete několik vstupních souborů, z nichž každý obsahuje popis jednoho polygonu P. Pro každý vstupní soubor nalezněte polygony A a B podle pravidel uvedených výše a vytvořte výstupní soubor s popisem

polygonů A a B. Pro každý vstupní soubor bude řešení existovat. Pokud existuje více řešení, popište libovolné jedno z nich.

Neodevzdávejte žádný program, pouze výstupní soubory.

Vstup: Dostanete 10 vstupních souborů pojmenovaných postupně `polygon1.in` až `polygon10.in`, přičemž číslo na konci jména souboru je číslo vstupu. Každý vstupní soubor vypadá následovně. První řádek obsahuje jediné celé číslo N: počet vrcholů zadaného polygonu. Následujících N řádků popisuje jednotlivé vrcholy polygonu v pořadí proti směru hodinových ručiček. $(I+1)$-ní řádek (pro $1 \leqq I \leqq N$) se skládá ze dvou nezáporných celých čísel X_I a Y_I oddělených jednou mezerou, která udávají x-ovou a y-ovou souřadnici I-tého vrcholu polygonu.

Výstup: Odevzdejte 10 výstupních souborů odpovídajících jednotlivým vstupním souborům. Každý z těchto výstupních souborů bude obsahovat popis polygonů A a B pro příslušný vstupní polygon. Tento popis musí začínat řádkem ve tvaru:

`#FILE polygon` I

kde celé číslo I $(1 \leqq I \leqq 10)$ je pořadové číslo vstupního souboru.

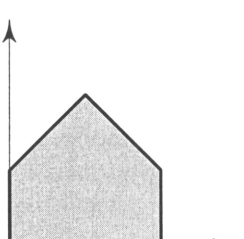
Zbytek výstupního souboru je v podobném tvaru, jako soubor vstupní. Druhý řádek obsahuje jedno celé číslo N_A: počet vrcholů polygonu A $(2 \leqq N_A \leqq 4)$. Následuje N_A řádků popisujících jednotlivé vrcholy polygonu A v pořadí proti směru hodinových ručiček. $(I+2)$-hý řádek (pro $1 \leqq I \leqq N_A$) obsahuje dvě celá čísla X a Y oddělená jednou mezerou, což jsou souřadnice I-tého vrcholu polygonu A.

(N_A+3)-tí řádek pak obsahuje jediné celé číslo N_B počet vrcholů polygonu B $(2 \leqq N_B)$. Následujících N_B řádků popisuje vrcholy polygonu B v pořadí proti směru hodinových ručiček. (N_A+J+3)-tí řádek (pro $1 \leqq J \leqq N_B$) obsahuje dvě celá čísla X a Y oddělená jednou mezerou, což jsou souřadnice J-tého vrcholu polygonu B.

Příklad vstupu a výstupu:

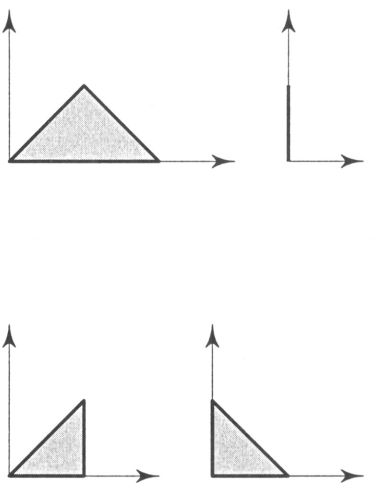
`polygon0.in`

```
5
0 1
0 0
2 0
2 1
1 2
```



Pro tento vstupní soubor jsou správně oba následující výstupní soubory (viz též obrázky). V obou případech je A trojúhelník a nemůže to být čtyřúhelník.

```
#FILE polygon 0
3
0 0
2 0
1 1
2
0 1
0 0
#FILE polygon 0
3
0 0
2 0
1 1
3
0 1
0 0
1 0
```



2. Artemis

Zeus obdaroval bohyni lovu Artemis obdélníkovým územím, aby tam mohla vysadit les. Jelikož Zeus byl tak trochu pedant, levý okraj území byl rovnoběžný s y-ovou souřadnou osou a dolní okraj s x-ovou osou, přičemž levý dolní roh byl v bodě $(0, 0)$. Navíc Artemidě dovolil stromy sázet pouze do bodů s celočíselnými souřadnicemi. Artemis se snažila, aby její les vypadal alespoň trochu přirozeně, a proto stromy sázela tak, aby žádné dva neměly stejnou x-ovou ani y-ovou souřadnici.

Jednoho dne si Zeus usmyslil, že pro něj Artemis musí porazit několik stromů, a to následovně:

1. musí jich být poraženo alespoň T,
2. musí být poraženy právě všechny stromy v nějakém obdélníku,
3. strany obdélníka musí být rovnoběžné se souřadnými osami,
4. v protilehlých vrcholech obdélníka musí růst stromy a ty budou také poraženy.

Artemis má ale stromy moc ráda, takže chce tyto podmínky splnit a přitom porazit co možná nejméně stromů. Vás požádala, abyste napsali

program, který pro zadané rozmístění stromů a minimální počet stromů k poražení T nalezne příslušný obdélník, kde se má porážet.

Vstup: Vstupní soubor se jmenuje `artemis.in`. Jeho první řádek obsahuje jediné celé číslo N: počet stromů v lese. Druhý řádek obsahuje rovněž jediné celé číslo T: minimální počet stromů, které mají být poraženy. Následujících N řádků popisuje polohy jednotlivých stromů: na každém z nich jsou dvě celá čísla X a Y, která představují x-ovou a y-ovou souřadnici příslušného stromu.

Výstup: Výstupní soubor se jmenuje `artemis.out`. Je tvořen jediným řádkem, na němž jsou dvě celá čísla I a J oddělená jednou mezerou. Říkají, že Artemis má kácet stromy v obdélníku, v jehož protilehlých vrcholech jsou stromy s čísly I a J (popsané na řádcích $I+2$ a $J+2$ ve vstupním souboru). Na pořadí stromů I, J ve výstupu nezáleží. Pro každá vstupní data existuje alespoň jedno řešení, pokud jich je více, vypište jedno libovolné z nich.

Příklad vstupu a výstupu:

`artemis.in`	`artemis.out`
`3`	`1 2`
`2`	
`1 1`	
`2 3`	
`5 6`	

Omezení: Pro všechny vstupy platí: $1 < N \leqq 20\,000$, $0 \leqq X, Y \leqq 64\,000$ a $1 < T \leqq N$. Navíc v 50 % vstupů je $1 < N < 5\,000$.

3. Hermes

I řečtí bohové jdou s dobou. Nedávno se z vrcholu Olympu přestěhovali do velkoměsta s pravoúhlou sítí ulic rovnoběžných se souřadnými osami. Ulice mají celočíselné souřadnice a pro každé celé číslo existuje svislá i vodorovná ulice s tímto číslem. Dvojice celých čísel pak určují křižovatky ulic. Horké letní dny bohové tráví v kavárnách umístěných právě na těchto křižovatkách. Posel bohů Hermes dostal za úkol doručit odpočívajícím bohům světelné signály. Může se při tom pohybovat pouze ulicemi města. Každý signál je určen jedinému bohovi, ale nevadí, když ho spatří i někteří z ostatních bohů.

Hermes dostane souřadnice jednotlivých kaváren v pořadí, v jakém do nich mají být signály doručeny. Svou cestu začne v bodě $(0, 0)$. Bohové

signály vidí na libovolnou vzdálenost, ale pouze podél ulic. Má-li tedy Hermes doručit signál bohovi do kavárny na křižovatce (X_i, Y_i), stačí mu dostat se do libovolného bodu na téže vodorovné ulici (s y-ovou souřadnicí Y_i) nebo na téže svislé ulici (s x-ovou souřadnicí X_i). Když odešle poslední signál, jeho mise končí.

Napište program, který dostane zadanou posloupnost souřadnic křižovatek a nalezne délku nejkratší cesty, při níž Hermes doručí všechny signály.

Vstup: Vstupní soubor se jmenuje `hermes.in`. Jeho první řádek obsahuje jediné celé číslo N: počet signálů k doručení. Následujících N řádků obsahuje souřadnice N křižovatek, na které mají být jednotlivé signály doručeny. Pořadí řádků odpovídá pořadí doručování. Na každém z těchto N řádků se nacházejí dvě celá čísla: nejprve x-ová a pak y-ová souřadnice příslušné křižovatky.

Výstup: Výstupní soubor se jmenuje `hermes.out`. Obsahuje jediný řádek s jedním celým číslem: minimální vzdáleností, kterou musí Hermes urazit, aby doručil všechny signály.

Příklad vstupu a výstupu:

```
hermes.in                    hermes.out
5                            11
8 3
7 -7
8 1
-2 1
6 -5
```

Omezení: Ve všech vstupech je $1 \leqq N \leqq 20\,000$, $-1\,000 \leqq X_i, Y_i \leqq$ $\leqq 1\,000$. Mimo to je v 50 % vstupů $1 \leqq N \leqq 80$.

4. Empodia

Starověký matematik a filosof Pythagoras věřil, že pravá podstata tohoto světa je matematická. Současní biologové také občas kráčejí v jeho stopách. Mimo jiné studují vlastnosti *biosekvencí*. To jsou posloupnosti M celých čísel, které:

- ▷ obsahují každé z čísel $0, 1, \ldots, M-1$,
- ▷ začínají nulou a končí číslem $M-1$,
- ▷ neobsahují žádný prvek E bezprostředně následovaný prvkem $E+1$.

Souvislým podposloupnostem biosekvencí se říká *segmenty*. *Ohraničený interval* (dále již jen *interval*) je takový segment, který obsahuje

všechna celá čísla z rozmezí od hodnoty prvního prvku tohoto segmentu do hodnoty posledního prvku segmentu. Navíc první prvek musí být nejmenším prvkem segmentu a poslední prvek největším. První a poslední prvek musí být různé. Interval, který neobsahuje žádný kratší interval, se nazývá *empodio*.

Kupříkladu biosekvence $(0, 3, 5, 4, 6, 2, 1, 7)$ je sama o sobě interval. Jelikož ale obsahuje segment $(3, 5, 4, 6)$, který je také intervalem, není celá tato biosekvence empodio. Interval $(3, 5, 4, 6)$ již empodio je, jelikož žádný kratší interval neobsahuje. Žádná jiná empodia v uvažované biosekvenci nejsou.

Napište program, který pro danou biosekvenci vypíše všechna empodia v ní obsažená.

Vstup: Vstupní soubor se jmenuje `empodia.in`. Jeho první řádek obsahuje jediné celé číslo M: počet čísel ve vstupní biosekvenci. Následujících M řádků obsahuje jednotlivé prvky biosekvence, což jsou celá čísla v pořadí, jak po sobě v biosekvenci následují.

Výstup: Výstupní soubor se jmenuje `empodia.out`. Na jeho prvním řádku je jediné celé číslo H: počet empodií vyskytujících se ve vstupní biosekvenci. Následuje H řádků popisujících jednotlivá empodia v pořadí určeném tím, kde v biosekvenci začínají. Každý z těchto řádků obsahuje dvě celá čísla A a B (v tomto pořadí) oddělená jednou mezerou, kde A je pořadové číslo prvního prvku empodia v biosekvenci a B pořadové číslo posledního prvku empodia.

Příklad vstupu a výstupu:

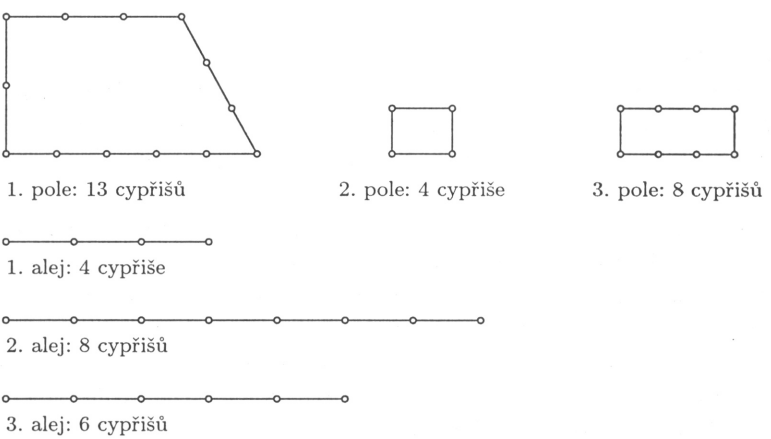
`empodia.in`	`empodia.out`
8	1
0	2 5
3	
5	
4	
6	
2	
1	
7	

Omezení: Ve všech vstupech mimo jednoho je $1 \leqq M \leqq 60\,000$. V jednom vstupu je $1\,000\,000 \leqq M \leqq 1\,100\,000$. Mimo to, v 50 % vstupů je $M \leqq 2\,600$.

5. Sedlák

Byl jednou jeden sedlák. Měl několikero polí lemovaných cypřišovými stromy (inu, řecký sedlák). Na každém ze svých dalších pozemků vysázel ještě jednu samostatnou alej, tvořenou jednou řadou cypřišů. Jak kolem polí, tak v alejích ovšem mezi každými dvěma sousedními cypřiši stojí ještě jeden olivovník. Jiné cypřiše ani olivovníky už naštěstí nemá.

Roky plynuly a život starého sedláka se zvolna chýlil ke konci. Jednoho dne si zavolal svého nejstaršího syna a pravil: „Dám ti libovolných Q cypřišů, které si vybereš, a s nimi dostaneš i každý olivovník, který sousedí se dvěma tebou vybranými cypřiši.“ Z každého pole a každé aleje si syn může vybrat libovolnou kombinaci cypřišů. Miluje ovšem olivy, a proto si chce vybrat takových Q cypřišů, aby s nimi získal co nejvíce olivovníků.



Obr. 63. Příklad rozmístění cypřišů. Olivovníky nejsou zobrazeny.

V příkladě vyobrazeném na obr. 63 si má syn vybrat $Q = 17$ cypřišů. Aby získal co nejvíce olivovníků, vybere si všechny cypřiše na 1. a 2. poli a dostane s nimi 17 olivovníků.

Napište program, který na základě čísla Q a údajů o polích a alejích spočte maximální možný počet olivovníků, které může syn získat.

Vstup: Vstupní soubor se jmenuje `farmer.in`. Jeho první řádek obsahuje tři celá čísla Q, M a K. Číslo Q udává počet cypřišů, které si syn má vybrat, M určuje počet sedlákových polí a K počet alejí. Na druhém řádku je M celých čísel $N_1, N_2, \ldots, N_M$, kde N_I určuje počet cypřišů na I-tém poli. Třetí řádek obsahuje K celých čísel $R_1, R_2, \ldots, R_K$, přičemž R_J udává počet cypřišů v J-té aleji.

Výstup: Výstupní soubor se jmenuje `farmer.out`. Obsahuje jediný řádek s jediným celým číslem, které udává maximální počet olivovníků, jež může syn získat.

Příklad vstupu a výstupu:

farmer.in	farmer.out
17 3 3	17
13 4 8	
4 8 6	

Omezení: Ve všech vstupech je $0 \leqq Q \leqq 150\,000$, $0 \leqq M \leqq 2\,000$, $0 \leqq K \leqq 2\,000$, $3 \leqq N_1 \leqq 150$, $3 \leqq N_2 \leqq 150$, ..., $3 \leqq N_M \leqq 150$, $2 \leqq R_1 \leqq 150$, $2 \leqq R_2 \leqq 150$, ..., $2 \leqq R_K \leqq 150$. Celkový počet všech cypřišů je alespoň Q. Mimo to, v 50 % vstupů je $Q \leqq 1\,500$.

6. Phidias

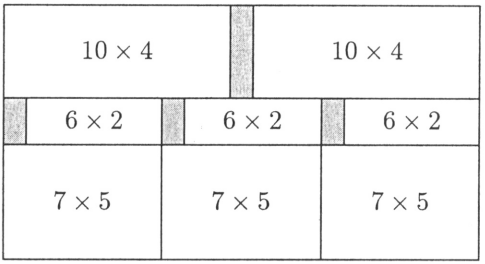
Slavný řecký sochař Phidias (u nás od nepaměti známý spíše pod jménem Feidias) se chystá stvořit další z divů světa. Proto potřebuje obdélníkové desky z mramoru o velikostech $W_1 \times H_1$, $W_2 \times H_2$, ..., $W_N \times$ $\times H_N$.

Před pár dny Phidias připadl na velký obdélníkový blok mramoru. Rozhodl se ho rozřezat, aby získal desky požadovaných velikostí. Jakýkoli kus mramoru (ať už to je původní blok nebo části z něj nařezané) může přeříznout svisle nebo vodorovně na dvě obdélníkové části s celočíselnou šířkou i výškou, přičemž řez musí vždy být rovný a vést od jednoho okraje k druhému. Nijak jinak řezat nelze a ani není možné nařezané kusy slepovat k sobě. A jelikož povrch mramoru má kresbu, nelze kusy ani otáčet: pokud Phidias uřízne část o rozměrech $A \times B$, nemůže ji použít jako desku velikosti $B \times A$, leda že by bylo $A = B$. Od každé požadované velikosti může vyrobit libovolný, třeba i nulový počet desek.

Odpadem jsou ty desky, které po provedení všech řezů nemají žádnou z požadovaných velikostí. Phidias by rád věděl, jak má svůj blok rozřezat, aby se do odpadu dostalo co nejméně mramoru.

Například na níže uvedeném obrázku máme blok o šířce 21 a výšce 11. Potřebujeme desky o velikostech 10×4, 6×2, 7×5 a 15×10. Tehdy je nejmenší možná plocha odpadu 10 a obr. 64 ukazuje jedno možné rozřezání s tímto množstvím odpadu.

Vaším úkolem je napsat program, který pro zadanou velikost původního bloku a požadované velikosti desek spočte nejmenší možnou plochu odpadu.



Obr. 64

Vstup: Vstupní soubor se jmenuje `phidias.in`. Jeho první řádek obsahuje dvě celá čísla: nejprve W, šířku původního bloku, a následně H, jeho výšku. Druhý řádek obsahuje jediné celé číslo N: počet požadovaných velikostí desek. Na následujících N řádcích jsou popsány jednotlivé požadované velikosti desek. Na každém z těchto řádků jsou dvě celá čísla udávající příslušnou velikost: šířka W_i následovaná výškou H_i $(1 \leqq i \leqq N)$.

Výstup: Výstupní soubor se jmenuje `phidias.out`. Obsahuje jediné celé číslo: minimální možnou plochu odpadu.

Příklad vstupu a výstupu:

`phidias.in`	`phidias.out`

`21 11`
`4`
`10 4`
`6 2`
`7 5`
`15 10`
	`10`

Omezení: Ve všech vstupech je $1 \leqq W \leqq 600$, $1 \leqq H \leqq 600$, $0 < N \leqq 200$, $1 \leqq W_i \leqq W$ a $1 \leqq H_i \leqq H$. Navíc 50 % vstupů má $W \leqq 20$, $H \leqq 20$ a $N \leqq 5$.

RNDr. Karel Horák, CSc., Mgr. Jan Kára,
doc. RNDr. Jaromír Šimša, CSc., RNDr. Jaroslav Švrček, CSc.,
doc. RNDr. Pavel Töpfer, CSc., RNDr. Jaroslav Zhouf, PhD.

PADESÁTÝ TŘETÍ ROČNÍK MATEMATICKÉ OLYMPIÁDY NA STŘEDNÍCH ŠKOLÁCH

Vydala Jednota českých matematiků a fyziků,
Žitná 25, 117 10 Praha 1
jako svou publikaci č. 57-551-06
Sazbu programem TEX připravil Karel Horák
Vytisklo Polygrafické středisko
Univerzity Palackého v Olomouci
1. vydání
Praha 2006

ISBN 80-7015-025-4

MO

matematická olympiáda

ISBN 80-7015-025-4