

49. ročník matematické olympiády na středních školách

Leo Boček (editor); Karel Horák (editor); Jaromír Šimša (editor); Jaroslav Švrček (editor); Pavel Töpfer (editor): 49. ročník matematické olympiády na středních školách. Zpráva o řešení úloh ze soutěže konané ve školním roce 1999/2000. 41. mezinárodní matematická olympiáda. 12. mezinárodní olympiáda v informatice. (Czech). Praha: Jednota českých matematiků a fyziků, 2005.

Persistent URL: <http://dml.cz/dmlcz/405008>

Terms of use:

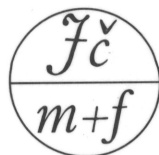
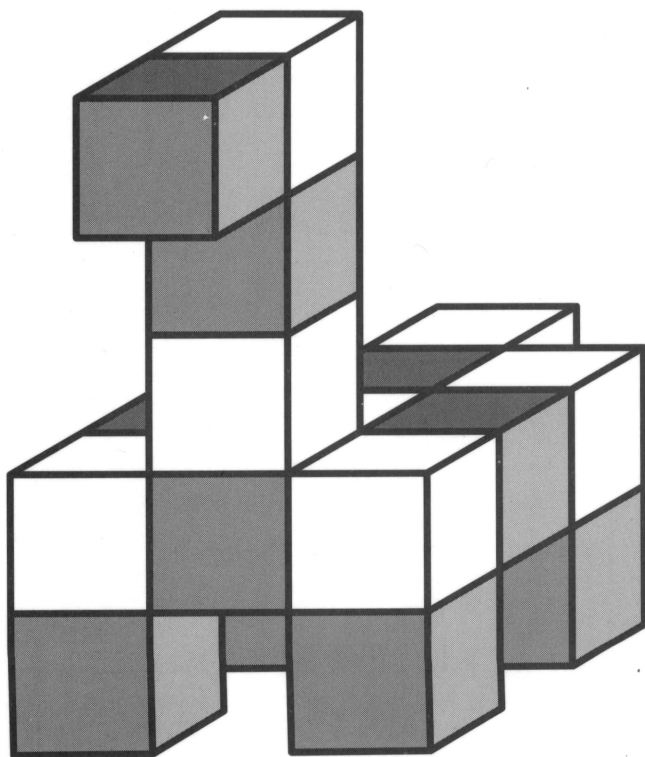
Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

MO

**49. ročník matematické olympiády
na středních školách**



ČTYŘICÁTÝ DEVÁTÝ ROČNÍK MATEMATICKÉ OLYMPIÁDY NA STŘEDNÍCH ŠKOLÁCH

Zpráva o řešení úloh ze soutěže konané ve školním roce 1999/2000

41. MEZINÁRODNÍ MATEMATICKÁ OLYMPIÁDA
DVANÁCTÁ MEZINÁRODNÍ OLYMPIÁDA V INFORMATICE

Za přispění spolupracovníků zpracovali
doc. RNDr. Leo Boček, CSc., RNDr. Karel Horák, CSc.,
doc. RNDr. Jaromír Šimša, CSc., RNDr. Jaroslav Švrček, CSc.,
doc. RNDr. Pavel Töpfer, CSc.

© Karel Horák, editor, 2005

ISBN 80-7015-984-7

Obsah

O průběhu 49. ročníku matematické olympiády	5
Nejúspěšnější řešitelé II. kola MO v kategoriích A, B, C a P	9
Výsledky celostátního kola 49. ročníku MO kategorie A	22
Výsledky celostátního kola 49. ročníku MO kategorie P	23
Kategorie C	24
Texty úloh	24
Řešení úloh	28
Kategorie B	47
Texty úloh	47
Řešení úloh	50
Kategorie A	70
Texty úloh	70
Řešení úloh	74
Kategorie P	101
Texty úloh	101
Řešení úloh	118
Přípravné soustředění před 41. MMO	147
Úlohy zadáné na přípravném soustředění	148
6. česko-slovenské střetnutí	151
Řešení úloh	152
Řešení úloh	152
41. mezinárodní matematická olympiáda	159
Texty soutěžních úloh	162
Řešení úloh	163
Sedmý ročník Středoevropské olympiády v informatice	177
12. mezinárodní olympiáda v informatice	179
Texty soutěžních úloh	181

O průběhu 49. ročníku matematické olympiády

Soutěž Matematická olympiáda ve školním roce 1999/2000 pořádaly pro žáky středních a základních škol Ministerstvo školství, mládeže a tělovýchovy České republiky ve spolupráci s Jednotou českých matematiků a fyziků a Matematickým ústavem Akademie věd ČR. Soutěž řídil ústřední výbor matematické olympiády (ÚV MO) prostřednictvím oblastních a okresních výborů matematické olympiády.

Cílem soutěže je vyhledávání žáků talentovaných v matematice, probouzení jejich hlubšího zájmu o matematiku a rozvíjení jejich matematických schopností. Ve školním roce 1999/2000 se uskutečnil její 49. ročník.

V průběhu 49. ročníku MO se konala dvě zasedání ÚV MO, první dne 14. prosince 1999 v Praze, druhé 10. dubna 2000 na gymnáziu v Bílovci při celostátním kole kategorie A. Bylo projednáváno hodnocení průběhu soutěže, zabezpečení celostátních soustředění úspěšných řešitelů MO, příprava na MMO, organizace dalších kol soutěže a v neposlední řadě i další spolupráce mezi českým a slovenským výborem olympiády.

V organizaci vlastní soutěže nedošlo k žádným podstatným změnám. Pro žáky středních škol byla soutěž organizována ve čtyřech kategoriích A, B, C a P. Kategorie A byla určena žákům 3. a 4. ročníků středních škol, kategorie B byla pro žáky 2. ročníků a v kategorii C soutěžili žáci 1. ročníků. Pro žáky všech tříd středních škol byla určena ještě kategorie P, zaměřená na úlohy z programování a matematické informatiky.

V kategoriích A, B a C má I. kolo dvě části. V první části řeší soutěžící 6 úloh doma nebo v matematických kroužcích a mohou se přitom radit se svými učiteli, vedoucími kroužků apod. Druhá část má formu klauzurní práce, v níž řeší žáci tři úlohy v omezeném čase 4 hodin. Řešitelé, kteří úspěšně projdou prvním kolem, jsou pozváni do druhého (oblastního) kola soutěže, kde řeší čtyři úlohy opět v limitu čtyř hodin.

V kategoriích A a P se koná ještě třetí, celostátní kolo. Celostátní kolo 49. ročníku se uskutečnilo ve dnech 9.–12. dubna 2000 (kategorie A) a 13.–15. dubna 2000 (kategorie P) na Gymnáziu M. Koperníka v Bílovci. Obě kola byla velmi pečlivě připravena, za což je nutno poděkovat celému

kolektivu učitelů bíloveckého gymnázia a především pak řediteli školy Mgr. Václavu Vaňkovi.

Na základě jednotné koordinace úloh II. kola kategorie A bylo k účasti ve III. kole pozváno 51 nejlepších řešitelů z celé České republiky. Soutěžními dny byly 10. a 11. duben 2000. Žákům byly tradičně předloženy v každém z obou dnů vždy 3 úlohy. Na řešení každé trojice úloh měli žáci rezervovány 4,5 hodiny čistého času a každá úloha byla přitom hodnocena maximálně 7 body.

V rámci doprovodného programu dostali soutěžící možnost navštívit ostravské planetárium a po soutěži pak absolvovali výlet do Kopřivnice, kde si prohlédli nově vybudované muzeum TATRY, a do Štramberku, kde navštívili dominantu města *Trúbu*.

Vybraná družstva se zúčastnila mezinárodní matematické olympiády i mezinárodní olympiády v informatice. Těmto soutěžím je věnována samostatná kapitola v závěru této ročenky. Přípravě českého družstva na mezinárodní matematickou olympiádu bylo věnováno soustředění 17.–21. dubna 2000 v Jevíčku.

K matematické olympiádě vedle vlastní soutěže patří i řada doprovodných akcí pro talentované žáky. Z akcí pořádaných oblastními výbory MO k nim zejména patří semináře pro řešitele MO a instruktáže pro učitele. Pro nejúspěšnější řešitele oblastních kol MO a korespondenčních seminářů byla pořádána (většinou týdenní) soustředění.

Ústřední výbor MO zajišťoval již tradiční soustředění úspěšných řešitelů úloh MO a FO pro žáky nematurujících ročníků. Proběhlo ve dnech 28. května až 9. června 2000 v Jevíčku.

Soutěžní úlohy I. (domácího) kola všech kategorií matematické olympiády jsou publikovány v tzv. soutěžních letácích. Úlohy jsou dále zveřejňovány v časopisech *Matematika*, *fyzika*, *informatika* a *Rozhledy matematicko-fyzikální*. Na pomoc učitelům jsou pak rozesílány na školy komentáře k úlohám.

Tímto svazkem, který držíte v ruce, zaplňujeme mezeru v chybějících ročenkách, jež přestaly v 90. letech v SPN vycházet. Postupně by se tak měly objevit i ostatní chybějící ročníky (43, 44, 46–48), neboť jejich těžištěm je zejména velký počet zajímavých a originálních úloh, které by jinak zůstaly ukryty v soukromých archivech. Těšíme se na váš ohlas a připomínky. Zároveň vychází i ročenka minulého 53. ročníku MO.

Autoři ročenky jménem Ústředního výboru MO děkují touto cestou všem organizátorům soutěže, především pak učitelům za jejich oběta-

vou spolupráci a za péči, kterou věnují svým žákům. Zároveň vyzývají všechny zájemce o spolupráci při tvorbě zajímavých úloh. Zkuste zažít pocit radosti z toho, že objevíte svou úlohu i se svým jménem v soutěžním letáku.

Návrhy na soutěžní úlohy pro kategorie A, B a C laskavě zasílejte na adresu předsedy úlohové komise MO doc. RNDr. Jaromíra Šimši, CSc., MÚ AV ČR, Žižkova 22, 616 00 Brno. Úlohová komise se schází zpravidla dvakrát ročně za účasti českých i slovenských kolegů (v tomto ročníku MO se schůzky konaly 24.–26. listopadu 1999 v Jevíčku a 10.–12. května 2000 v Modre na Slovensku).

Návrhy úloh vhodných pro kategorii P zasílejte na adresu doc. RNDr. Pavla Töpfera, CSc., MFF UK Praha, Malostranské nám. 25, 118 00 Praha 1. Návrhy úloh vhodných pro kategorie Z zasílejte na adresu doc. Pavla Tlustého, CSc., Pedagogická fakulta Jihočeské univerzity, Jeronýmova 10, České Budějovice.

Tabulka 1

Počty žáků středních škol soutěžících v I. kole 49. ročníku MO

Oblast	Kategorie								Celkem	
	A		B		C		P			
	S	U	S	U	S	U	S	U	S	U
Praha	46	31	90	52	89	34	19	19	244	136
Střední Čechy	118	57	104	26	141	47	27	24	390	154
Západní Čechy	39	29	23	11	52	36	11	6	125	82
Liberec	21	19	17	8	46	24	7	7	91	58
Ústí n. Labem	24	16	26	13	56	26	2	2	108	57
Východní Čechy	42	26	40	23	59	37			141	86
Jihlava	50	33	43	27	39	24	11	9	143	93
Brno	108	62	65	32	125	75	21	21	319	190
Zlín	61	26	49	11	52	15	10	10	172	62
Severní Morava	137	38	139	44	286	56	14	14	576	152
ČR	646	337	596	247	945	374	122	112	2 309	1 070

Tabulka 2

Počty žáků středních škol soutěžících v II. kole 49. ročníku MO

Oblast	Kategorie								Celkem	
	A		B		C		P			
	S	U	S	U	S	U	S	U	S	U
Praha	23	14	50	29	32	24	18	7	123	74
Střední Čechy	51	9	26	10	43	16	22	6	142	41
Západní Čechy	29	11	11	5	36	30	6	1	82	47
Liberec	19	1	6	2	23	12	7	0	55	15
Ústí n. Labem	13	1	13	3	22	9	2	1	50	14
Východní Čechy	26	3	22	9	35	25			83	37
Jihlava	30	2	23	5	21	11	7	1	81	19
Brno	58	16	32	12	74	49	21	6	185	83
Zlín	23	1	12	1	13	8	9	0	57	10
Severní Morava	38	20	44	21	56	42	14	4	152	87
ČR	310	78	239	97	355	226	106	26	1 010	427

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 je uvedeno nejvýše prvních deset úspěšných řešitelů. Označení G znamená gymnázium, M, resp. MF zaměření studijního oboru 01 Matematika, resp. 02 Matematika a fyzika.

• • • • • **Praha** • • • • •

Kategorie A

1. *Pavel Kůs*, GChD, Zborovská, Praha 5
2. *David Chodounský*, GChD, Zborovská, Praha 5
3. *Jan Pipek*, GJK, Parlérova, Praha 6
4. *Jiří Koula*, G U Libeň. zámku, Praha 6
5. *Filip Jaroš*, GChD, Zborovská, Praha 5
6. *David Šálek*, G Na Vítězné pláni, Praha 4
7. *Jan Kulveit*, G Ústavní, Praha 8
8. *Martin Tancer*, GChD, Zborovská, Praha 5
- 9.–11. *Lenka Havrdová*, G Písnická, Praha 4
Pavel Milar, G Bernarda Bolzana, V Holešovičkách, Praha 8
František Němec, GChD, Zborovská, Praha 5

Kategorie B

- 1.–2. *Martin Klimeš*, G Botičská, Praha 2
Jan Verfl, G Postupická, Praha 4
3. *Ondřej Šimek*, G U Libeňského zámku, Praha 8
- 4.–10. *Martin Blažek*, G Českolipská Praha 9
Josef Cibulka, G Štěpánská Praha 1
Petra Kocábová, G Hellichova Praha 1
Ondřej Kurka, G Zborovská Praha 5
Alice Mašková, GJK, Parlérova, Praha 6

Martin Suda, G Na Vítězné pláni, Praha 4
Václav Vlasák, G Korunní Praha 2

Kategorie C

1. *Pavel Bažant*, GChD, Zborovská, Praha 5
2. *Jan Kadlec*, GChD, Zborovská, Praha 5
- 3.–4. *Martin Káldy*, GChD, Zborovská, Praha 5
Vojtěch Matyska, GChD, Zborovská, Praha 5
- 5.–7. *Jiří Hron*, GChD, Zborovská, Praha 5
Martin Popel, G Nad Alejí, Praha 6
Petr Škoda, G Ústavní, Praha 8
8. *Miroslav Rada*, G Hellichova, Praha 1
- 9.–11. *Jaroslav Kodým*, G U Libeňského zámku, Praha 8
Jan Popelka, G Mezi školami, Praha 5
Jakub Šolc, SPŠST Panská 3, Praha 1

Kategorie P

1. *Roman Krejčík*, GChD, Zborovská, Praha 5
- 2.–3. *Martin Beránek*, G Ohradní, Praha 4
Pavel Charvát, G Ohradní, Praha 4
4. *Jiří Svoboda*, GChD, Zborovská, Praha 5
- 5.–6. *Jan Kadlec*, GChD, Zborovská, Praha 5
Matyáš Novák, G Ústavní, Praha 8
7. *Václav Jůza*, G Špitálská, Praha 9

• • • • • Střední Čechy • • • • •

Kategorie A

1. *Tomáš Hanzák*, G Kladno
2. *Ondřej Šerý*, G a sport. škola Kladno
- 3.–4. *Pavel Čížek*, G Kralupy
Pavel Vališ, G Kralupy
- 5.–6. *Jakub Jeřábek*, G Hořovice
Miroslav Pištěk, G Sedlčany

Kategorie B

1. *Tomáš Hanzák*, G Kladno
- 2.-3. *Martin Doubek*, G Kladno
Ondřej Chochola, G Kladno
- 4.-6. *Petr Kavánek*, G Čáslav
Marek Pokorný, G Kladno
Jan Skoupý, G Příbram
- 7.-8. *Zdeněk Poledník*, G Beroun
Martin Werner, G Čáslav
- 9.-10. *Lucie Koltzsiterová*, G Benešov
Oldřich Svoboda, G Kolín VI

Kategorie C

1. *Jan Lamač*, G Mnichovo Hradiště
2. *Pavel Čížek*, G Kralupy
- 3.-4. *Martin Kruliš*, G Kolín V
Marek Šmíd, G Brandýs nad Labem
5. *Thomas Prentis*, G Kladno
6. *Jiří Hyldebrant*, G Vlašim
- 7.-8. *Jiří Paleček*, G Kladno
Radka Picková, GJP, Mladá Boleslav
- 9.-12. *Pavel Brom*, GJP, Mladá Boleslav
Ondřej Herout, G Nové Strašecí
Miroslav Nulíček, G Čáslav
Petr Procházka, G Čáslav

Kategorie P

1. *Pavel Čížek*, Dvořákovo GOA, Kralupy
- 2.-3. *Miroslav Pištěk*, GOA Sedlčany
Tomáš Vala, Dvořákovo GOA, Kralupy
- 4.-6. *Ondřej Šerý*, SpG Kladno-Sítná
Zdeněk Bulan, G Benešov
Jakub Maršík, G Benešov

Kategorie A

1. *Jan Houštěk*, G Pelhřimov
- 2.–3. *Tomáš Dolejšek*, G České Budějovice, Jírovcova
Tomáš Lusk, G České Budějovice, Jírovcova

Kategorie B

- 1.–2. *Ondřej Břížek*, G České Budějovice, Jírovcova
Jan Heyda, G České Budějovice, Jírovcova
- 3.–5. *Inga Domračeva*, G České Budějovice, Jírovcova
Richard Chudoba, G České Budějovice, Jírovcova
Jan Pospíchal, G Pelhřimov
6. *David Pospíšil*, G České Budějovice, Jírovcova
- 7.–10. *David Pražák*, G České Budějovice, Jírovcova
Martin Sigmund, G České Budějovice, Jírovcova
Věra Šídllová, G České Budějovice, Jírovcova
Jan Švec, SPŠ Písek

Kategorie C

- 1.–2. *Milan Straka*, G Strakonice
O. Šedivý, G České Budějovice, Jírovcova
3. *Jiří Danihelka*, SPŠ Písek
- 4.–6. *Tereza Cvejnová*, G Písek
Pavel Kubas, G Jindřichův Hradec
Josef Machač, G Tábor
Jan Šedivý, G Tábor
8. *Daniel Kratochvíl*, G České Budějovice, Jírovcova
- 9.–11. *Kotrba*, G Jindřichův Hradec
Radek Mlada, G Pelhřimov
Václav Profant, G Strakonice

Kategorie P

- 1.–2. *Milan Straka*, G Strakonice
Tomáš Lusk, G České Budějovice, Jírovcova

Kategorie A

- 1.–3. *Jaromír Dobrý*, G Plzeň, Mikulášské nám.
Václav Flaška, Svob. chebská škola, Cheb
Ondřej Suchý, G Plzeň, Mikulášské nám.
4. *Josef Křišťan*, G Plzeň, Mikulášské nám.
5. *Tomáš Matoušek*, G Karlovy Vary
- 6.–8. *Jaromír Chalupský*, G Sušice
Ondřej Rucký, G Plzeň, Mikulášské nám.
Martin Setvín, G Plzeň, Mikulášské nám.
9. *Petr Matas*, G Klatovy
- 10.–11. *Martin Kozák*, G Klatovy
Jiří Skála, G Plzeň, Mikulášské nám.

Kategorie B

1. *Josef Mládek*, 1. G Plzeň
- 2.–4. *Lenka Beranová*, G Klatovy
Hana Medová, G Sokolov
Jindřich Poskočil, 3. G Plzeň
5. *Zdeňka Kapešová*, G Sokolov

Kategorie C

- 1.–2. *Michal Bareš*, 1. G Plzeň
Michal Kvíz, 1. G Plzeň
- 3.–4. *Ladislav Lenc*, 1. G Plzeň
Jaroslav Svoboda, 1. G Plzeň
5. *Martin Jílek*, G Domažlice
- 6.–7. *Martina Smitková*, 1. G Plzeň
Barbora Vostracká, 1. G Plzeň
8. *Pavel Klesa*, 1. G Plzeň
- 9.–10. *Hana Kutáková*, 1. G Plzeň
David Mareček, 1. G Plzeň

Kategorie P

1. *Ondřej Rucký*, 1. G Plzeň

..... **Ústí nad Labem**

Kategorie A

1. *Nelly Vostrá*, G Děčín, Komenského nám.

Kategorie B

- 1.–3. *Lukáš Janda*, G Teplice, Čs. Dobrovolců
Pavel Šašek, G Ústí nad Labem, Stavbařů
Miroslav Šulc, G Ústí nad Labem, Stavbařů

Kategorie C

1. *David Török*, G Ústí nad Labem, Stavbařů
- 2.–3. *Pavel Daniel*, ZŠ Teplice III, Buzulucká
Tomáš Janata, G Most
- 4.–7. *Ladislav Bartoš*, G J. Jungmana, Litoměřice
Ondřej Honzl, G Podbořany
Tomáš Kozelek, G Kadaň
Jana Šindelářová, G Kadaň
8. *Vít Šípál*, G Ústí nad Labem, Jateční
9. *Jan Kolařík*, G Děčín, Komenského nám.

Kategorie P

1. *Ladislav Prošek*, G V. Hlavatého, Louny
2. *Jan Hyka*, G Most

..... **Liberec**

Kategorie A

1. *Jan Kynčl*, GaSG Jilemnice

Kategorie B

1. *Ondřej Melich*, G Jablonec, U Balvanu
2. *Jiří Sláma*, G Liberec, Jeronýmova

Kategorie C

1. *Lukáš Bajer*, GFXŠ, Liberec
- 2.–3. *Terezie Párysová*, GFXŠ, Liberec
Karel Pupík, G dr. Randy, Jablonec
- 4.–5. *Jana Hornová*, G Jablonec, U Balvanu
Soňa Charvátová, G dr. Randy, Jablonec
6. *Roman Bém*, GFXŠ, Liberec
- 7.–8. *Eva Böhmová*, GFXŠ, Liberec
Ondřej Nešpor, GFXŠ, Liberec
- 9.–12. *Pavel Grmela*, GFXŠ, Liberec
Zuzana Klápšťová, G Jablonec, U Balvanu
Marek Sobota, G Liberec, Jeronýmova
Vratislav Žabka, G Frýdlant

Kategorie P

1. *Martin Nečaský*, G I. Olbrachta, Semily

• • • • • Východní Čechy • • • • •

Kategorie A

1. *Martin Hroch*, SPŠE Dobruška
- 2.–3. *Michal Hroch*, GJKT, Hradec Králové
Václav Jára, GJKT, Hradec Králové

Kategorie B

- 1.–3. *Jan Hošek*, GJKT, Hradec Králové
Jitka Kühnová, GJKT, Hradec Králové
Ondřej Pokorný, G Chrudim
4. *Martina Kubátová*, GJKT, Hradec Králové

5. *Václav Šafář*, G Broumov
6. *Monika Šmídová*, G Havlíčkův Brod
7. *Pavel Mühl*, G Nová Paka
- 8.–9. *Vojtěch Hladík*, GJKT, Hradec Králové
Jiří Sedlák, SPŠE Pardubice

Kategorie C

1. *Petr Rezek*, G Litomyšl
- 2.–3. *Petr Pošta*, G Pardubice
Jan Prachař, G Rychnov n. K.
- 4.–5. *Miroslav Hejna*, G Rychnov n. K.
Lenka Matoulková, G Náchod
6. *Tomáš Gubanec*, GJKT, Hradec Králové
- 7.–10. *Dana Chromíková*, G Pardubice, Dašická
Štěpán Krtička, G Náchod
Hana Matoušová, G Litomyšl
Martin Zelenka, GJKT, Hradec Králové

Kategorie P

1. *Tomáš Vyskočil*, G Lanškroun
2. *Roman Hocke*, G Pardubice, Dašická

• • • • • Jihlava • • • • •

Kategorie A

1. *Zdeněk Černý*, G Žďár nad Sázavou
2. *Michaela Šedová*, G Jihlava

Kategorie B

1. *Vít Urbánek*, G Jihlava
2. *Jaromír Šír*, GVM, Nové Město n. M.
3. *Pavel Trnka*, G Třebíč
4. *Ondřej Kozdas*, G Jihlava
5. *Michal Musil*, G Jihlava

Kategorie C

- 1.-3. *Josef Janák*, G Velké Meziříčí
Jiří Lipovský, G Bystřice nad Pernštejnem
Jaroslav Pazdera, G Jihlava
- 4.-5. *Jana Fabriková*, G Velké Meziříčí
Ivo Kubita, GKP, Znojmo
6. *Jindřiška Vábková*, G Žďár nad Sázavou
- 7.-9. *Miroslav Kaňka*, G Třebíč
Vojtěch Mates, G Třebíč
Lukáš Vašek, Biskupské G Žďár nad Sázavou
10. *Roman Matoušek*, G Žďár nad Sázavou

Kategorie P

1. *Petr Adam*, G Třebíč

..... **Brno**

Kategorie A

1. *Jan Herman*, G Brno, tř. Kpt. Jaroše
- 2.-3. *Jiří Chaloupka*, G Židlochovice
Tomáš Protivínský, G Brno, tř. Kpt. Jaroše
- 4.-5. *Alexandr Jevsejenko*, G Brno, tř. Kpt. Jaroše
Rudolf Stolař, G Brno, tř. Kpt. Jaroše
- 6.-8. *Ondřej Bystrý*, G Brno, tř. Kpt. Jaroše
Jiří Cvachovec, G Brno, tř. Kpt. Jaroše
Tomáš Híkl, G Brno, Tábořská
9. *Jan Trojek*, G Břeclav
- 10.-12. *Pavel Dvořák*, G Brno, tř. Kpt. Jaroše
Martin Ležatka, G Brno, tř. Kpt. Jaroše
Antonín Pavelka, G Brno, tř. Kpt. Jaroše

Kategorie B

1. *Tomáš Protivínský*, G Brno, tř. Kpt. Jaroše
- 2.-3. *Petr Dadák*, G Brno, Vídeňská

- Miroslav Frost*, G Brno, Elgartova
4.-7. *Jan Beneš*, Biskupské G Brno
Karel Martišek, G Brno, Elgartova
Ema Tomášková, G Brno, tř. Kpt. Jaroše
Martina Vaníková, G Brno, Žižkova
8.-11. *Šárka Buršová*, G Šlapanice
Vladimír Hloušek, G Boskovice
Pavel Krupa, G Šlapanice
Hana Martinásková, G Brno, tř. Kpt. Jaroše

Kategorie C

- 1.-2. *Sven Dražan*, G Brno, tř. Kpt. Jaroše
Marek Krčál, G Brno, tř. Kpt. Jaroše
3. *Robert Ganián*, G Brno, Pastviny
4.-6. *Jan Hladký*, G Brno, tř. Kpt. Jaroše
Jan Kuchař, G Brno, tř. Kpt. Jaroše
Veronika Trnková, G Brno, tř. Kpt. Jaroše
7.-10. *Lukáš Chváta*, G Brno, Vejrostova
Petra Kohoutková, G Brno, Žižkova
Antonín Pavelka, G Brno, tř. Kpt. Jaroše
Jaroslav Zůda, G Brno, tř. Kpt. Jaroše

Kategorie P

1. *Pavel Šimeček*, G Brno, tř. Kpt. Jaroše
2. *Miloslav Trmač*, Biskupské G Brno
3.-4. *Jiří Cvachovec*, G Brno, tř. Kpt. Jaroše
Marek Sulovský, G Brno, tř. Kpt. Jaroše
5.-6. *Václav Fiala*, G Brno, tř. Kpt. Jaroše
Martin Vejnár, G Brno, tř. Kpt. Jaroše

..... **Zlín**

Kategorie A

1. *Zuzana Špiritová*, G Uherský Brod
2. *Tomáš Potrusil*, G Uherské Hradiště

- 3.-4. *Pavel Kočica*, G Uherský Brod
Jan Kodovský, G Zlín, Lesní čtvrť
5. *Libor Křížka*, G Hodonín
6. *Václav Blaha*, G Uherský Brod

Kategorie B

1. *Josef Houser*, G Zlín, Lesní čtvrť

Kategorie C

- 1.-3. *Radek Krejčířík*, G Uherské Hradiště
Ludmila Obdržálková, G Uherské Hradiště
Jan Vavrys, GJAK, Uherský Brod
4.-6. *Tomáš Kadlec*, G Holešov
Jaroslav Kudlička, G Hodonín
Jiří Zámečník, GJAK, Uherský Brod
7.-8. *Martin Cetkovský*, G Zlín, Lesní čtvrť
Tomáš Dlabaja, G Holešov
9. *Petr Fišer*, G Kroměříž
10. *Ondřej Kunčar*, G Kroměříž

Kategorie P

1. *Martin Zlomek*, G Strážnice
2. *Jiří Kazík*, G Kunovice

• • • • • Severní Morava • • • • •

Kategorie A

1. *Jaroslav Hájek*, GMK, Bílovec
2. *Martin Holík*, GMK, Bílovec
3. *Jakub Bystroň*, G Karviná
4.-6. *Pavel Augustinský*, G Havířov, Studentská
Jakub Krchák, GTGM, Vsetín
Jiří Šimša, GJW, Prostějov
7.-11. *Lucia Jarešová*, GJŠ, Přerov

Martin Kocurek, GTGM, Vsetín
Ondřej Kreml, GMK, Bílovec
Michal Malohlava, GMK, Bílovec
Jan Trčka, GMK, Bílovec

Kategorie B

1. *Jan Klesnil, GJŠ, Přerov*
- 2.–3. *Jaroslav Hájek, GMK, Bílovec*
Libor Olšák, GMK, Bílovec
- 4.–7. *Jakub Galgonek, GPB, Frýdek-Místek*
Jan Kaluza, GPB, Frýdek-Místek
Lenka Škovroňová, GMK, Bílovec
Jan Zacios, G Ostrava 8-Poruba, Čs. exilu
8. *Lucia Jarešová, GJŠ, Přerov*
- 9.–11. *Jan Dostál, MGO Opava*
Miroslav Moser, GMK, Bílovec
Jiří Peinlich, GMK, Bílovec

Kategorie C

1. *Václav Cviček, GPB, Frýdek-Místek*
2. *Martin Kyslinger, G Šternberk*
3. *Ondřej Májek, ZŠ Opava, Otická*
- 4.–6. *Jan Chmelař, G Hranice na Moravě*
Veronika Chromčíková, GJŠ, Přerov
Pavel Ludvík, GMK, Bílovec
- 7.–9. *Jitka Halašová, GJW, Prostějov*
Eliška Krainová, GMK, Bílovec
Tibor Vansa, Matiční G, Ostrava
- 10.–12. *Pavel Jež, GPB, Frýdek-Místek*
Lukáš Perůtka, G Hranice na Moravě
Eva Riedingerová, GJW, Prostějov

Kategorie P

1. *Jakub Bystroň*, G Karviná
2. *Martin Děcký*, GMK, Bílovec
3. *Jan Chrastina*, G Třinec, Komenského
4. *Pavel Petřek*, GMK, Bílovec

Výsledky celostátního kola 49. ročníku MO

kategorie A

Vítězové

1.	<i>Jan Houštěk</i> , 7/7, G Pelhřimov	40 b.
2.	<i>Jan Herman</i> , 3/4 (M), G Brno, tř. Kpt. Jaroše	37 b.
3.	<i>Jan Kynčl</i> , 5/6, G Jilemnice	36 b.
4.	<i>Jan Pipek</i> , 7/8, G Praha 6, Parlérova	32 b.
5.–6.	<i>Jaroslav Hájek</i> , 2/4, G M. Koperníka, Bílovec	31 b.
	<i>Ondřej Šerý</i> , 7/8, G a sport. škola Kladno	31 b.
7.	<i>Ondřej Suchý</i> , 6/7, G Plzeň, Mikulášské nám.	27 b.
8.–10.	<i>Josef Křišťan</i> , 6/7, G Plzeň, Mikulášské nám.	26 b.
	<i>Ondřej Rucký</i> , 7/7, G Plzeň, Mikulášské nám.	26 b.
	<i>Rudolf Stolař</i> , 3/4 (M), G Brno, tř. Kpt. Jaroše	26 b.
11.–12.	<i>Jan Kulveit</i> , 7/7, G Praha 8, Ústavní	25 b.
	<i>Tomáš Protivínský</i> , 2/4, (M), G Brno, tř. Kpt. Jaroše	25 b.

Další úspěšní řešitelé

13.	<i>Martin Tancer</i> , 2/4 (M), G Ch. Dopplera, Praha 5	23 b.
14.–16.	<i>David Chodounský</i> , 3/4 (M), G Ch. Dopplera, Praha 5	22 b.
	<i>Filip Jaroš</i> , 3/4 (M), G Ch. Dopplera, Praha 5	22 b.
	<i>Jiří Koula</i> , 3/4 (P), G Praha 8, U Libeňského zámku	22 b.
17.–18.	<i>Tomáš Matoušek</i> , 7/7, G Karlovy Vary	21 b.
	<i>Pavel Vališ</i> , 3/4, Dvořákovo G, Kralupy nad Vltavou	21 b.
19.–20.	<i>Ondřej Bystrý</i> , 4/4 (M), G Brno, tř. Kpt. Jaroše	20 b.
	<i>Václav Flaška</i> , 7/8, Svobodná chebská škola Cheb	20 b.
21.	<i>Jaromír Dobrý</i> , 7/7, G Plzeň, Mikulášské nám.	19 b.
22.	<i>Ondřej Kreml</i> , 3/4, G M. Koperníka, Bílovec	18 b.
23.	<i>Pavel Augustinský</i> , 7/7, G Havířov	17 b.
24.	<i>František Němec</i> , 3/4 (M), G Ch. Dopplera, Praha 5	16 b.

Výsledky celostátního kola 49. ročníku MO
kategorie P

Vítězové

- | | |
|---|-------|
| 1. Ondřej Rucký, 7/7, G Plzeň, Mikulášské nám. | 41 b. |
| 2. Pavel Charvát, 7/7, G Praha 4, Ohradní | 39 b. |
| 3. Jakub Bystroň, 7/7, G Karviná | 34 b. |
| 4. Jiří Svoboda, 6/8, G Ch. Dopplera, Praha 5 | 33 b. |
| 5. Matyáš Novák, 7/7, G Praha 8, Ústavní | 30 b. |
| 6. Pavel Čížek, 5/8, Dvořákovo G, Kralupy nad Vltavou | 29 b. |
| 7. Václav Jůza, 7/7, G Praha 9, Špitálská | 26 b. |

Další úspěšní řešitelé

- | | |
|---|-------|
| 8. Miroslav Trmač, 7/8, Biskupské G, Brno | 25 b. |
| 9. Roman Krejčík, 3/4, G Ch. Dopplera, Praha 5 | 24 b. |
| 10. Miroslav Pištěk, 7/7, GOA Sedlčany | 23 b. |
| 11. Ondřej Šerý, 7/8, SpG Plzeňská, Kladno | 22 b. |
| 12. Martin Beránek, 6/7, G Praha 4, Ohradní | 21 b. |
| 13. Ladislav Prošek, 7/7, G V. Hlavatého, Louny | 19 b. |
| 14. Zdeněk Bulan, 6/7, G Benešov | 18 b. |
| 15. Martin Vejnár, 3/8, G Brno, tř. Kpt. Jaroše | 17 b. |

Kategorie C

Texty úloh

C – I – 1

Při dělení jistého přirozeného čísla čísly 19 a 99 vyjdou jako zbytky dvě prvočísla. Součet obou neúplných podílů se rovná 1999. Určete dělené číslo.
(J. Šimša)

C – I – 2

Najděte všechny pravouhlé trojúhelníky, ve kterých spojnice středů vepsané a opsané kružnice svírá s přeponou úhel 45 stupňů.
(M. Krállová)

C – I – 3

Zjistěte nejmenší přirozená čísla k , pro něž platí jednotlivá tvrzení a), b) a c): Obsadíme-li figurkami libovolných k polí šachovnice 8×8 , pak budou obsazena některá

- a) tři sousední pole některého řádku,
- b) tři sousední pole některé šikmé řady,
- c) čtyři sousední pole některého řádku nebo sloupce.

Šikmou řadou rozumíme takovou skupinu polí, jejichž úhlopříčky jednoho z obou směrů leží na jedné a téže přímce.
(J. Šimša)

C – I – 4

Jirka zhotovil papírový model pravidelného čtyřbokého jehlanu $ABCDV$ s podstavou $ABCD$. Když pak model podél čtyř hran rozřízl, bylo ho možno rozvinout (bez překrytí) do roviny. Kolik různých sítí daného jehlanu tak mohl Jirka dostat? Ukázalo se, že síť, kterou Jirka dostal, měla tvar (nekonvexního) sedmiúhelníku. Vypočtěte úhel AVB v boční stěně jehlanu.
(P. Leischner)

C – I – 5

V číselném výrazu

$$+1 + 2 + 3 - 4 - 5 - 6 + 7 + 8 + 9 - 10 - 11 - 12 + \dots + \\ + 595 + 596 + 597 - 598 - 599 - 600),$$

ve kterém chybí levá závorka, jsou postupně vypsána všechna přirozená čísla od 1 do 600; před nimi se pravidelně opakují tři znaménka $+$ a tři znaménka $-$. Doplňte levou závorku do výrazu tak, aby vyšel výsledek 378. (P. Černek)

C – I – 6

Je dán pravidelný šestiúhelník $KLMNOP$. Sestrojte pravoúhlý trojúhelník ABC s přeponou AB tak, aby jeho vrchol C ležel na úsečce NP , body M , O , K ležely po řadě na přímkách AB , BC , CA a aby přímka NP rozdělila trojúhelník ABC na dvě části se stejným obsahem.

(K. Černeková)

C – S – 1

Najděte nejmenší přirozené číslo k , pro které platí: Vybereme-li libovolných k různých čísel z množiny $\{1, 4, 7, 10, 13, \dots, 1999\}$, pak mezi vybranými existují dvě různá čísla, jejichž součet se rovná 2000.

(J. Zhouf)

C – S – 2

Čtverec $ABCD$ a obdélník $AEFD$ mají takovou vzájemnou polohu, že bod B leží na kružnici vepsané trojúhelníku AEF . Vypočtete poměr délky a šířky obdélníku $AEFD$.

(J. Šimša)

C – S – 3

Jestliže celé kladné číslo N vydělíme číslem 19 a získaný neúplný podíl dále vydělíme číslem 99, vyjde nám při druhém dělení stejný neúplný podíl a stejný zbytek, jako když původní číslo N vydělíme číslem 1999. Určete jak nejmenší, tak největší takové číslo N .

(J. Šimša)

C – II – 1

Ze dřeva je vyrobeno šest shodných pravidelných čtyřbokých jehlanů a krychle. Stěna krychle je shodná s podstavami jehlanů. Určete poměr povrchu krychle a tělesa, které vznikne slepením podstav jehlanů se stěnami krychle, je-li poměr objemů těchto těles 1 : 2. (P. Leischner)

C – II – 2

Milan zapsal za sebe několik prvních přirozených čísel, vynechal při tom jen čísla 4, 9, 14, 19, 24, 29, ... Pak mezi zapsaná čísla vepsal střídavě znaky minus a plus, takže dostal výraz

$$1 - 2 + 3 - 5 + 6 - 7 + 8 - 10 + 11 - 12 + 13 - 15 + \dots$$

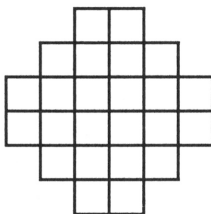
Nakonec ještě vepsal levou závorku za každý znak minus a stejný počet pravých závorek zapsal až na konec výrazu:

$$1 - (2 + 3 - (5 + 6 - (7 + 8 - (10 + 11 - (12 + 13 - (15 + \dots))))))$$

Výsledný výraz měl hodnotu 103. Kolik čísel v Milanově výrazu bylo? (Zjistěte všechny možnosti.) (P. Černek)

C – II – 3

Jaký největší počet figurek je možno rozestavit na jednotlivá pole hrací desky z obrázku tak, aby v žádné šikmé řadě nebyla figurkami obsazena



žádná tři sousední pole? Nezapomeňte zdůvodnit, proč větší počet figurek takto rozestavit nelze. (Šikmou řadou rozumíme takovou skupinu polí, jejichž úhlopříčky jednoho z obou směrů leží na jedné přímce.)

(J. Bábel)

C – II – 4

V rovině jsou dány body A , L , M takové, že $|AL| = 6,3$ cm, $|AM| = 5,6$ cm, $|LM| = 1,8$ cm. Sestrojte lichoběžník $ABCD$, jemuž lze vepsat kružnici, která se dotýká ramene BC v bodě L a základny CD v bodě M (body dotyku se základnou AB a ramenem AD lichoběžníku $ABCD$ nejsou dány).

(*J. Šimša*)

C – I – 1

Obě dělení hledaného čísla N vyjádříme rovnostmi

$$N = 19a + p \quad \text{a} \quad N = 99b + q,$$

kde a, b jsou příslušné neúplné podíly a p, q příslušné zbytky. Podle zadání jsou čísla p, q prvočísla, přičemž jako zbytky splňují nerovnosti $p < 19$ a $q < 99$. Nezáporná celá čísla a, b jsou dle zadání zase taková, že jejich součet se rovná číslu 1 999. Proto platí $b = 1\,999 - a$ a z dvojího vyjádření čísla N ,

$$N = 19a + p = 99 \cdot (1\,999 - a) + q,$$

odvodíme rovnost $118a + (p - q) = 197\,901$. Protože rozdíl zbytků $p - q$ je „malé“ číslo, přesněji $-99 < p - q < 19$, je podle poslední rovnosti číslo $118a$ takový násobek čísla 118, jenž leží mezi čísly $197\,901 - 19$ a $197\,901 + 99$. Dělením $197\,901 : 118$ zjistíme, že $197\,901 = 1\,677 \cdot 118 + 15$. Proto nutně platí $a = 1\,677$ (takže $b = 322$) a $p - q = 15$. Z poslední rovnosti plyne, že jedno z prvočísel p, q je liché a druhé sudé, tedy $q = 2$ a $p = 17$. Zbývá vypočítat hodnotu N :

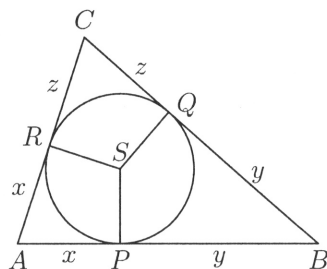
$$N = 19 \cdot 1\,677 + 17 = 99 \cdot 322 + 2 = 31\,880.$$

Poznámka. Můžeme také nejprve odhadnout velikost hledaného čísla N , například řešením „přibližné“ rovnice $N/19 + N/99 = 1\,999$, která má kořen $N \doteq 31\,865$. Pak pomocí nejbližších násobků čísel 19 a 99 ($19 \cdot 1\,677 = 31\,863$, $99 \cdot 322 = 31\,878$, $19 \cdot 1\,678 = 31\,882$) určíme jediné možné hodnoty neúplných podílů a, b (stačí diskutovat případy $N < 31\,878$, $31\,878 \leq N < 31\,882$ a $N \geq 31\,882$).

Odpověď: Hledané číslo je rovno 31 880.

C – I – 2

Část A. Předpokládejme, že kružnice vepsaná *obecnému* trojúhelníku ABC se dotýká stran AB, BC, AC po řadě v bodech P, Q a R (obr. 1). Z hodin geometrie žáci ví o souměrnosti obou tečen vedených z daného



Obr. 1

bodu k dané kružnici. Úsečky AP a AR jsou tudíž shodné stejně jako úsečky BP a BQ a úsečky CQ a CR . Ukážeme, jak lze délky

$$x = |AP| = |AR|, \quad y = |BP| = |BQ|, \quad z = |CQ| = |CR|$$

vyjádřit pomocí délek stran $a = |BC|$, $b = |AC|$, $c = |AB|$. Podle obr. 1 platí

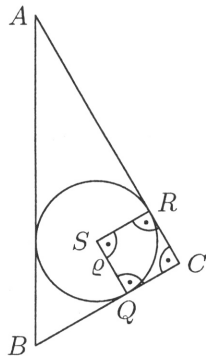
$$x + y = c, \quad y + z = a, \quad x + z = b,$$

což je soustava tří lineárních rovnic, z níž snadno plynou užitečné vzorce

$$x = \frac{b + c - a}{2}, \quad y = \frac{a + c - b}{2}, \quad z = \frac{a + b - c}{2}.$$

Část B. Nyní předpokládejme, že ABC je *pravoúhlý* trojúhelník s přeponou AB . Označme S střed kružnice vepsané a Q, R její body dotyku s odvěsnami BC, AC (obr. 2). Protože $SQCR$ je pravoúhelník, jehož sousední strany SQ a SR jsou shodné (mají délku rovnou poloměru ϱ kružnice vepsané), jedná se o čtverec o straně ϱ . Délku úseku $z = |CQ|$ jsme však vypočetli v části A. Tak pro poloměr ϱ kružnice vepsané pravoúhlému trojúhelníku s odvěsnami a, b a přeponou c dostáváme vzorec

$$\varrho = \frac{a + b - c}{2}.$$



Obr. 2

Část C. Předpokládejme konečně, že ABC je pravoúhlý trojúhelník s přeponou AB , který splňuje podmínku z textu úlohy. Podle Thaletovy věty je střed O kružnice opsané trojúhelníku ABC středem přepony AB . Podle zadání má jeden z úhlů SOA, SOB (kde S je střed

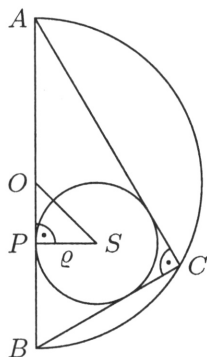
vepsané kružnice) velikost 45° ; necht' je to úhel SOB (obr. 3), jinak prohodíme označení vrcholů A, B . Bod P , v němž se kružnice vepsaná dotýká strany AB , je pak vnitřním bodem úsečky OB . Podle části A platí vzorec $|BP| = \frac{1}{2}(a + c - b)$, takže

$$|OP| = |OB| - |BP| = \frac{c}{2} - \frac{a + c - b}{2} = \frac{b - a}{2}.$$

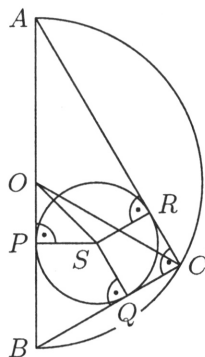
Vyjádřili jsme délku odvěsny OP pravoúhlého trojúhelníku SOP . Jeho druhá odvěsna SP má podle části B délku $|SP| = \varrho = \frac{1}{2}(a + b - c)$. Protože však úhel SOP má dle předpokladu velikost 45° , je trojúhelník SOP rovnoramenný:

$$|OP| = |SP|, \text{ neboli } \frac{b - a}{2} = \frac{a + b - c}{2}, \text{ neboli } 2a = c.$$

Strany pravoúhlého trojúhelníku ABC jsou proto v poměru $a : b : c = 1 : \sqrt{3} : 2$, takže jeho vnitřní úhly jsou, jak je dobře známo, 30° , 60° a 90° . Pro úplnost je třeba ještě ukázat, že takový trojúhelník skutečně požadovanou vlastnost má. To lze provést obrácením předchozího postupu: z rovnosti $2a = c$ se odvodí rovnost $|OP| = |SP|$, která znamená, že pravoúhlý trojúhelník SOP je rovnoramenný, takže úhel SOP má skutečně velikost 45° .



Obr. 3



Obr. 4

Stručné řešení podle obr. 4: Úsečka SO svírá s přeponou AB úhel 45° , právě když $|OP| = |SP|$. Protože však $|SP| = |SR| = |QC|$ a $|PB| = |QB|$, je rovnost $|OP| = |SP|$ ekvivalentní s rovností $|OP| + |PB| = |QC| + |QB|$, tedy s rovností $|OB| = |BC|$. Podle Thaletovy věty však

vždy platí $|OB| = |OC|$, takže rovnost $|OB| = |BC|$ nastane, právě když je trojúhelník OBC rovnostranný, tedy právě když úhel ABC měří 60° .

Odpověď: Požadovanou vlastnost mají právě ty pravoúhlé trojúhelníky, jejichž ostré vnitřní úhly mají velikost 30° a 60° .

C – I – 3

Danou řadu 11 písmen rozdělíme nejprve zleva doprava na skupiny po 4, 4 a 3 písmenech. Protože nikde za sebou nestojí čtyři písmena B, je v první skupině čtyř písmen obsaženo aspoň jedno A, totéž platí i o druhé skupině čtyř písmen. Tak jsme dokázali, že v celé řadě 11 písmen jsou aspoň dvě A. Mohou to být právě dvě A, jak ukazuje příklad řady BBBA-BBBABBB. Podruhé rozdělíme celou řadu zleva doprava na skupiny po 3, 3, 3 a 2 písmenech. Protože nikde za sebou nestojí tři písmena A, je v každé skupině tři sousedních písmen aspoň jedno B. Protože jsme vyčlenili tři takové (disjunktní) skupiny, jsou v celé řadě aspoň tři písmena B. Jinak řečeno, v celé řadě 11 písmen je nejvýše osm písmen A. Může to být právě osm A, jak ukazuje příklad řady AABAABAABAA. Kdybychom řešili obecnější úlohu o řadě N písmen A a B, ve které nikde za sebou nestojí ani a písmen A, ani b písmen B, o odpovědi na stejnou otázku by rozhodovala dělení $N : a$ a $N : b$ (se zbytky).

a) Všimněme si, že v každém řádku je možné obsadit nejvýše šest polí tak, aby mezi obsazenými nebyla žádná tři sousední pole (stačí rozdělit všech osm polí řádku na skupiny 3, 3 a 2 sousedních polí a zopakovat úvahu z předchozího odstavce). Proto lze na celé šachovnici obsadit nejvýše $8 \times 6 = 48$ polí tak, aby v žádném řádku nebyla obsazena tři sousední pole. Jinak řečeno, obsadíme-li *libovolných* 49 polí, pak obsazena budou některá tři sousední pole některého řádku. Tento jev nenastane pro (jediné) obsazení 48 polí, které vidíte na obr. 5, kde jsou obsazená pole vyznačena šrafováním (zmíněná jedinečnost plyne z toho, že je jediné možné obsazení šesti polí v každém řádku). Proto je hledané číslo k rovno 49.

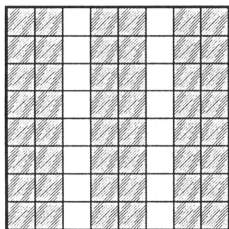
b) Zkusme postupovat obdobně jako při řešení části a). Vyberme tedy jeden z obou směrů šikmých řad, například směr „zleva zdola napravo nahoru“ a posuzujme, kolik polí lze obsadit v jednotlivých šikmých řadách vybraného směru tak, aby v žádné z nich nebyla obsazena žádná tři sousední pole. Počty všech polí v těchto 15 řadách jsou (shora dolů) 1, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4, 3, 2 a 1; proto v nich lze požadovaným způsobem (při požadované podmínce) obsadit po řadě nejvýše 1, 2, 2, 3, 4, 4, 5, 6,

5, 4, 4, 3, 2, 2 a 1 pole (znovu uplatníme úvahu o disjunktních trojicích sousedních polí z úvodního odstavce). Proto lze na celé šachovnici obsadit nejvýše

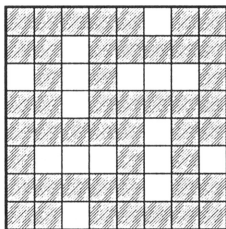
$$1 + 2 + 2 + 3 + 4 + 4 + 5 + 6 + 5 + 4 + 4 + 3 + 2 + 2 + 1 = 48$$

polí tak, aby nebyla obsazena žádná tři sousední pole žádné šikmé řady vybraného směru. Obsadíme-li s touto podmínkou v každé z uvažovaných 15 šikmých řad největší možný počet polí vhodným způsobem (v řadách o 1, 2, 5 a 8 polích je toto „maximální“ obsazení jediné, v řadách o 3, 4, 6 a 7 polích nikoliv, zvolme i v nich obsazení z pohledu zleva doprava typu XXOXXO...), dostaneme na celé šachovnici opět obsazení 48 polí z obr. 6. Co je důležité: při tomto obsazení také v šikmých řadách druhého směru nejsou nikde obsazena tři sousední pole! Shrňme, co jsme zjistili:

1. Obsadíme-li na šachovnici libovolných 49 polí, budou mezi nimi tři sousední pole některé šikmé řady zvoleného směru.
2. Na šachovnici lze obsadit 48 polí tak, aby mezi nimi nebyla žádná tři sousední pole žádné šikmé řady (jednoho i druhého směru).



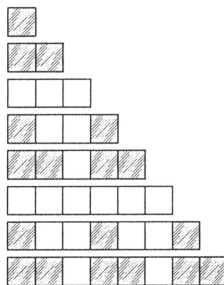
Obr. 5



Obr. 6

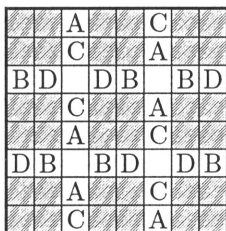
To znamená, že pro část b) úlohy je hledané číslo k rovno 49. Dodejme, že existují právě čtyři různá obsazení 48 polí zmíněná v (2). Kromě úplného obsazení šesti sloupců z obr. 5 a obsazení, které je jeho otočením o 90° (je tedy úplným obsazením šesti řad), je to zajímavé obsazení z obr. 6 a jeho „zrcadlové překlopení“.

DŮKAZ. Jak víme, při každém obsazení 48 polí zmíněném v (2) musí být v každé šikmé řadě šachovnice obsazen největší možný počet polí (výčet těchto maximálních počtů jsme uvedli výše); na obr. 7 jsou znázorněny řady polí délek 1 až 8, šrafováním jsou v nich vyznačena právě ta pole, která jsou *nutně obsazena*, je-li v celé řadě jakkoliv obsazen pří-

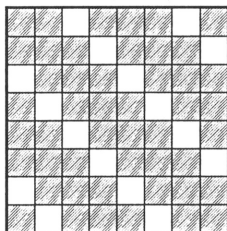


Obr. 7

slušný maximální počet polí (připomínáme podmínku: obsazena nejsou žádná tři sousední pole); znovu zopakujme, že všechna obsazená pole jsou vyznačena pouze v řadách o 1, 2, 5 a 8 polích. Odtud plyne, že při libovolném obsazení 48 polí zmíněném v (2) je nutně obsazeno 36 polí vyznačených na obr. 8; dále už je snadné ukázat, že zbývajících 12 obsazených polí je tvořeno dvěma šesticemi polí značených na obr. 8 jednou z dvojic písmen: A a C, B a C, A a D nebo B a D (obr. 5 odpovídá dvojici písmen B a D, obr. 6 dvojici A a D).



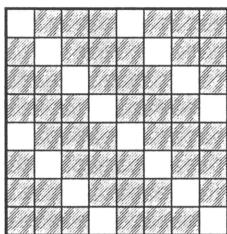
Obr. 8



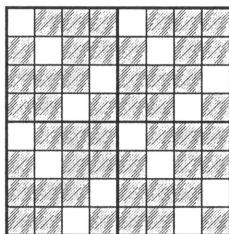
Obr. 9

c) Jako v části b) se zajímáme o skupiny polí dvou směrů, v tomto případě o řádky a sloupce. Posuzujme například nejdříve, kolik polí lze obsadit v jednotlivých řádcích tak, aby v žádném z nich nebyla obsazena čtyři sousední pole. Jak zjistíme pomocí rozdělení řádku na dvě čtveřice sousedních polí, obsazených polí bude v každém řádku nejvýše šest. Proto lze na celé šachovnici obsadit nejvýše $8 \times 6 = 48$ polí tak, aby v žádném řádku nebyla obsazena čtyři sousední pole. Jinak řečeno: obsadíme-li libovolných 49 polí, pak obsazena budou čtyři sousední pole některého řádku. (Je zajímavé srovnat tento závěr s obdobným závěrem z řešení části a), kde jsme se zajímali nikoliv o čtveřice, nýbrž o trojice sousedních polí.) Nyní samozřejmě vzniká otázka, zda je možné na

šachovnici obsadit 48 polí tak, aby nebyla obsazena čtyři sousední pole nejen v žádném řádku, ale ani v žádném sloupci. Taková obsazení existují, dva příklady vidíme na obr. 9 a 10. Proto i v části c) je hledané číslo k rovno 49. Bez důkazu dodejme ještě popis všech obsazení 48 polí šachovnice, kdy mezi obsazenými poli nejsou čtyři sousední pole žádného řádku ani sloupce: Celou šachovnici rozdělíme na čtyři čtvrtiny 4×4 , v jedné z nich, například levé horní části, obsadíme z 16 polí právě 12 tak, aby v ní zůstalo neobsazeno právě jedno pole v každém ze čtyř řádků i v každém ze čtyř sloupců (to je možné udělat $4 \cdot 3 \cdot 2 = 24$ způsoby, jeden z nich, odlišný od způsobů z obr. 9 a 10, je na obr. 11), a toto obsazení shodně přeneseme vodorovnými a svislými posunutími o čtyři pole do ostatních tří čtvrtin celé šachovnice.



Obr. 10



Obr. 11

Odpověď: Hledané číslo k je pro každou z částí a)–c) rovno témuž číslu 49 (pořadovému číslu tohoto ročníku MO).

C – I – 4

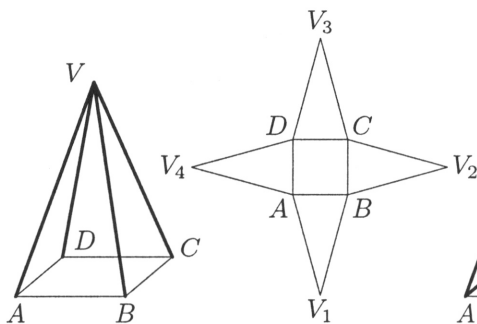
Počet různých sítí daného jehlanu určíme tak, že nejprve všechny možné sítě nakreslíme. Abychom některou možnost neopomenuli, měli bychom do výčtu sítí vnést *určitý systém*. Popíšeme dva přístupy, které takový systém vytvářejí.

Přístup 1 („od sítě k jehlanu“). Každá síť bude složena z jednoho čtverce o straně a a čtyř rovnoramenných trojúhelníků o stranách a , b , b , kde a značí délku podstavné hrany a b délku boční hrany daného jehlanu $ABCDV$. Přemýšlejme tedy o tom, jak takový čtverec a čtyři trojúhelníky „slepit“ podél shodných stran do „celku“ a zda tento celek skutečně vytvoří síť jehlanu. Je velmi přirozené rozčlenit řešení tohoto úkolu *podle počtu stran čtverce, které budou slepeny* (možné počty jsou 1 až 4).

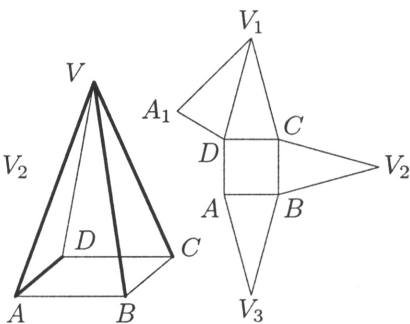
Přístup 2 („od jehlanu k síti“). Přemýšlejme o tom, jak rozříznout daný jehlan $ABCDV$ podél čtyř hran, abychom po rozvinutí dostali jeho síť. (Brzy si při tom uvědomíme jeden obecný poznatek: z každého vrcholu tělesa musí vycházet aspoň jedna hrana řezu.) Protože nám jde o počet různých (tj. po dvou neshodných) sítí, s ohledem na symetrii daného jehlanu není příliš vhodné systematizovat čtveřice hran řezu podle toho, zda obsahují některé konkrétní hrany (jako např. hrany AB , AV apod.). Výhodnější je rozdělení těchto čtveřic do skupin podle toho, *kolik hran řezu je v jehlanu podstavných* (a kolik bočních).

Protože oba popsané přístupy vedou ke shodné systematizaci (je-li právě k hran řezu podstavných, je v příslušné síti právě $4-k$ stran čtverce spleno s trojúhelníky), popíšeme výčet všech sítí jen podle přístupu 2:

1. Neleží-li v podstavě $ABCD$ žádná hrana řezu, je jehlan rozříznut podél všech čtyř bočních hran, příslušná síť je na obr. 12.



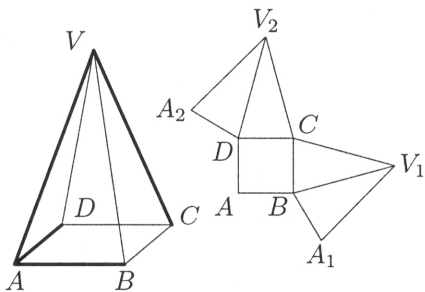
Obr. 12



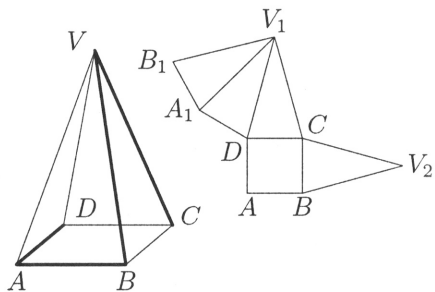
Obr. 13

2. Předpokládejme, že v podstavě $ABCD$ leží jediná hrana řezu, například hrana AD . Z vrcholů B a C musí vycházet nějaké hrany řezu, mohou to tedy být jediné hrany BV a CV . Tři hrany řezu jsou tedy AD , BV a CV , s ohledem na symetrii je lhostejno, zda je čtvrtou hranou řezu AV nebo DV , nechť je to tedy hrana AV jako na obr. 13.
3. Předpokládejme, že v podstavě $ABCD$ leží právě dvě hrany řezu. Rozlišíme, zda jsou to hrany sousední (např. AB a AD), nebo hrany protější (např. AD a BC); pro větší přehlednost oba případy posudíme v oddělených odstavcích:
 - (3a) Je-li podstava rozříznuta právě podél hran AB a AD (takže řezem v podstavě je lomená čára BAD), musí být třetí hranou řezu hrana CV , čtvrtá hrana řezu je pak jedna z hran AV , BV nebo DV . S ohledem na symetrii případů, kdy je čtvrtou hranou řezu

BV nebo DV , uvádíme jen obrázky pro hrany řezu AV (obr. 14) a BV (obr. 15).

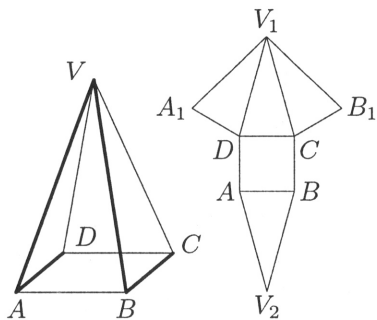


Obr. 14

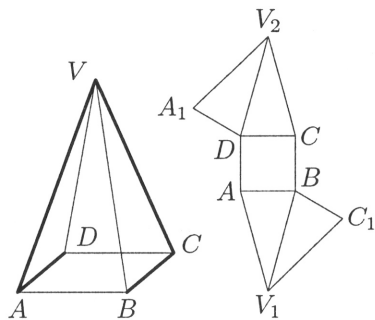


Obr. 15

(3b) Je-li podstava rozříznuta právě podél hran AD a BC , je třetí hranou řezu jedna z bočních hran AV , DV a čtvrtou hranou řezu jedna z bočních hran BV , CV (nemohou to totiž být ani obě hrany AV , DV , ani obě hrany BV , CV). S ohledem na symetrii stačí rozlišit jen dva případy: boční hrany řezu jsou buď AV a BV (obr. 16), nebo AV a CV (obr. 17).

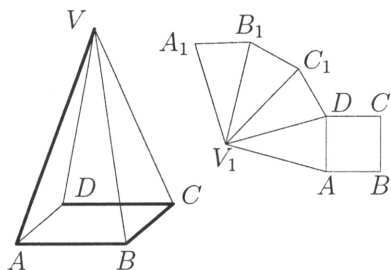


Obr. 16

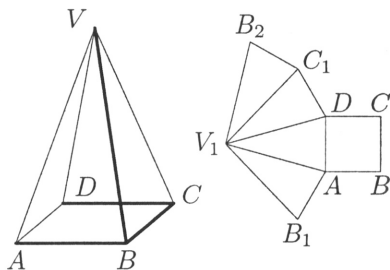


Obr. 17

4. Předpokládejme, že v podstavě $ABCD$ leží právě tři hrany řezu, například hrany AB , BC a CD , takže řezem v podstavě je lomená čára $ABCD$. S ohledem na symetrii nyní stačí rozlišit jen dva případy: čtvrtá hrana řezu vede do vrcholu V buďto z jednoho z obou krajních vrcholů zmíněné lomené čáry $ABCD$, například bodu A (obr. 18), nebo z jednoho z obou prostředních vrcholů, například bodu B (obr. 19).



Obr. 18



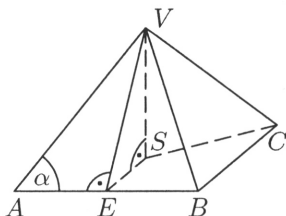
Obr. 19

Zjistili jsme, že daný jehlan má právě osm různých sítí. (Většina žáků asi správně všech osm sítí do svých řešení nakreslí, aniž pocítí nutnost vysvětlovat, proč jiné sítě neexistují. Diskutujme s nimi o této otázce.)

Přejdeme nyní k druhé části úlohy, otázce, kdy některá ze sítí daného jehlanu má tvar nekonzexního sedmiúhelníku. Podle obrázků vidíme, že každá síť má, obecně vzato, osm vrcholů; jejich počet se sníží na sedm, právě když se úhel u jednoho z osmi obecných vrcholů „napřímí“, tj. bude mít velikost 180° . Velikosti všech dotýčných úhlů lze snadno vyjádřit pomocí $\omega = |\sphericalangle AVB|$ a $\alpha = |\sphericalangle BAV|$; zjistíme tak, popsaná situace nastane, jen když jeden z úhlů

$$2\alpha, \alpha + 90^\circ, 2\alpha + 90^\circ, 2\omega, 3\omega \text{ nebo } 4\omega \quad (*)$$

bude 180° . Položme si nyní poněkud obecnější otázku: Jaké hodnoty α a ω jsou přípustné, tj. odpovídají nějakému jehlanu $ABCDV$? Označme S střed čtverce $ABCD$ a E střed hrany AB (obr. 20), z pravoúhlého



Obr. 20

trojúhelníku EVS plyne, že $|EV| > |ES|$ neboli $|EV| > |AE|$, proto pro úhel α v pravoúhlém trojúhelníku AVE platí $45^\circ < \alpha < 90^\circ$ (pro $\alpha = 45^\circ$ bychom dostali „zdegenerovaný“ jehlan s nulovou výškou, pro $\alpha = 90^\circ$ „jehlan“ s nekonečnou výškou, tedy hranol). Zároveň je jasné, že pro každé $\alpha \in (45^\circ, 90^\circ)$ odpovídající jehlan existuje. Odtud vzhledem

k rovnosti $2\alpha + \omega = 180^\circ$ plyne, že přípustné hodnoty ω zaplní interval $(0^\circ, 90^\circ)$. Proto z úhlů (*) mohou být přímé jediné úhly 3ω a 4ω . Pro $\omega = 60^\circ$ mají tvar sedmiúhelníku sítě z obr. 15 a 16, pro $\omega = 45^\circ$ sítě z obr. 18 a 19.

Odpověď: Jirka mohl dostat právě osm různých sítí. Úhel AVB měl velikost 45° nebo 60° .

C – I – 5

Rozdělíme je do pěti etap.

Část A. Podle způsobu opakování znamének rozdělíme daný výraz (ještě bez obou závorek) na 100 úseků po šesti číslech

$$\begin{aligned} &+1 + 2 + 3 - 4 - 5 - 6, \quad +7 + 8 + 9 - 10 - 11 - 12, \\ &\quad +13 + 14 + 15 - 16 - 17 - 18, \quad \dots, \\ &\quad +589 + 590 + 591 - 592 - 593 - 594, \\ &\quad +595 + 596 + 597 - 598 - 599 - 600 \end{aligned}$$

(říkejme jim dále stručně „úseky“). Při výpočtu celého výrazu bude výhodné určovat „díličí součty“ právě po těchto úsecích: po doplnění chybějící levé závorky se totiž naruší „celistvost“ jediného úseku (toho, do kterého závorku doplníme).

Uložte žákům úkol, aby vypsali čísla toho úseku, který obsahuje dané číslo x , například $x = 24$, $x = 100$, $x = 571$ apod. Žáci si tak uvědomí, jaký význam má při tom zbytek při dělení čísla x číslem 6, a tím se připraví na to, aby dokázali zapsat čísla v k -tém úseku ($k = 1, 2, 3, \dots, 100$) takto:

$$6k - 5, \quad 6k - 4, \quad 6k - 3, \quad 6k - 2, \quad 6k - 1, \quad 6k.$$

Část B. Určeme nyní hodnoty výrazů tvořených jednotlivými úseky. Po několika prvních výpočtech

$$1 + 2 + 3 - 4 - 5 - 6 = -9, \quad 7 + 8 + 9 - 10 - 11 - 12 = -9, \quad \dots$$

dojdeme k závěru, že pro každý úsek vyjde -9 . Ověříme to následujícím algebraickým postupem, přičemž první číslo úseku označíme písmenem x :

$$x + (x+1) + (x+2) - (x+3) - (x+4) - (x+5) = (3x+3) - (3x+12) = -9.$$

(V tomto místě ještě není důležité, že číslo x je tvaru $6k - 5$.)

Část C. Zjistíme nyní možné hodnoty V zkoumaného výrazu v případě, kdy chybějící levou závorku vepíšeme před číslo z některého konkrétního úseku, například toho, který obsahuje číslo 100:

$$V = 1 + \dots + \underbrace{*97 + *98 + *99 - *100 - *101 - *102}_{\text{někam sem doplníme závorku}} + 103 + \dots - 600$$

Hvězdičkami jsme označili možná místa pro doplňovanou závorku. Snadno určíme, že před číslem 97 je 16 úseků a že za číslem 102 je 83 úseků ($16 + 83 = 99$, nezapočítaný stý úsek je tvořen čísly od 97 do 102). Je zřejmé, že pokud umístíme závorku na místo před číslo 97, 98 nebo 99, tedy za znaménko plus, přispěje do výsledku V všech 100 úseků číslem -9 , takže vyjde $V = 100 \cdot (-9) = -900$. Umístíme-li závorku na místo před číslo 100, 101 nebo 102, tedy za znaménko minus, přispěje 16 úseků (před číslem 97) do výsledku V číslem -9 , zatímco 83 úseků (za číslem 102) přispěje číslem $-(-9) = 9$. Se závorkou před číslem 100 tak vyjde

$$V = 16 \cdot (-9) + 97 + 98 + 99 - 100 + 101 + 102 + 83 \cdot 9 = 1\,000,$$

se závorkou před číslem 101

$$V = 16 \cdot (-9) + 97 + 98 + 99 - 100 - 101 + 102 + 83 \cdot 9 = 798,$$

konečně se závorkou před číslem 102

$$V = 16 \cdot (-9) + 97 + 98 + 99 - 100 - 101 - 102 + 83 \cdot 9 = 594.$$

Část D. Konkrétní postup z části C nyní zopakujeme v obecné situaci, kdy závorku doplníme do k -tého úseku, na místo některé z hvězdiček:

$$V = 1 + \dots + *[6k - 5] + *[6k - 4] + *[6k - 3] - \\ - *[6k - 2] - *[6k - 1] - *[6k] + \dots - 600$$

(vyjádření čísel jsme zapsali do *hranatých* závorek kvůli odlišení od vepísané *kulaté* závorky). V části C bylo k rovno číslu 17, nyní je to libovolné přirozené číslo od 1 do 100 včetně. Před číslem $6k - 5$ je zřejmě $(k - 1)$ úseků a za číslem $6k$ je $(100 - k)$ úseků. Pokud umístíme závorku na místo před číslo $6k - 5$, $6k - 4$ nebo $6k - 3$, tedy za znaménko plus, přispěje do výsledku V všech 100 úseků číslem -9 , takže vyjde

$V = 100 \cdot (-9) = -900$. Umístíme-li závorku na místo před číslo $6k - 2$, $6k - 1$ nebo $6k$, tedy za znaménko minus, přispěje prvních $(k - 1)$ úseků číslem -9 , zatímco posledních $(100 - k)$ úseků přispěje číslem $-(-9) = 9$. Se závorkou před číslem $6k - 2$ tak vyjde

$$V = (k - 1) \cdot (-9) + [6k - 5] + [6k - 4] + [6k - 3] - [6k - 2] + [6k - 1] + [6k] + (100 - k) \cdot 9 = 898 + 6k,$$

se závorkou před číslem $6k - 1$

$$V = (k - 1) \cdot (-9) + [6k - 5] + [6k - 4] + [6k - 3] - [6k - 2] - [6k - 1] + [6k] + (100 - k) \cdot 9 = 900 - 6k,$$

konečně se závorkou před číslem $6k$

$$V = (k - 1) \cdot (-9) + [6k - 5] + [6k - 4] + [6k - 3] - [6k - 2] - [6k - 1] - [6k] + (100 - k) \cdot 9 = 900 - 18k.$$

Našli jsme všechny možné hodnoty V zkoumaného výrazu po doplnění levé závorky.

Část E. Zjistíme nyní všechny případy, kdy výsledek V má hodnotu 378 ze zadání úlohy. Podle části D stačí řešit rovnice

$$898 + 6k = 378, \quad 900 - 6k = 378, \quad 900 - 18k = 378$$

v oboru přirozených čísel k od 1 do 100. Řešení má pouze druhá a třetí rovnice, a to $k = 87$ resp. $k = 29$. Hodnotě $k = 87$ odpovídá umístění závorky před číslo $6k - 1 = 521$, hodnotě $k = 29$ odpovídá umístění závorky před číslo $6k = 174$.

Odpověď: Úloha má dvě řešení; závorku doplníme buďto bezprostředně před číslo 521, nebo bezprostředně před číslo 174.

C – I – 6

Jak je tomu u řešení konstrukčních úloh obvyklé, načrtne nejdříve do jednoho obrázku daný šestiúhelník $KLMNOP$ i hledaný trojúhelník ABC tak, aby jejich tvar i vzájemná poloha aspoň přibližně odpovídaly zadání (obr. 21). Speciální poloha daných bodů K, L, M, N, O, P

Ze vzájemné polohy úsečky NP a kružnice τ nad průměrem OK plyne, že pro každý pravidelný šestiúhelník $KLMNOP$ má úloha jediné řešení.

C – S – 1

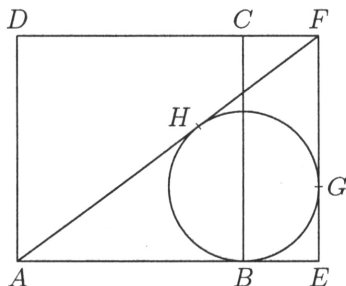
Označme $M = \{1, 4, 7, 10, 13, \dots, 1999\}$ a vypišme všechny součty dvou *různých* (to už nebudeme dále zdůrazňovat) čísel z M , které se rovnají číslu 2 000:

$$2\,000 = 1 + 1\,999 = 4 + 1\,996 = 7 + 1\,993 = \dots = 997 + 1\,003.$$

S výjimkou jediného čísla 1 000 vystupuje každé číslo z M v právě jednom součtu (součet $1\,000 + 1\,000$ se dle zadání neuvažuje). Protože $997 = 1 + 3 \cdot 332$, je vypsáno právě 333 součtů. Lze tedy vybrat 334 čísel z M (po jednom z každého vypsání součtu spolu s číslem 1 000, tedy například čísla $1, 4, 7, \dots, 997, 1\,000$) tak, že součet žádných dvou vybraných čísel není 2 000. Vybereme-li však libovolných 335 čísel z M , pak některá dvě vybraná čísla jsou sčítanci jednoho z vypsání součtů (zopakujme: vypsání součtů je 333 a chybí v nich jediné číslo z M). Proto je $k = 335$ hledaná hodnota.

C – S – 2

Protože oba pravoúhelníky musí ležet ve stejné polorovině s hraniční přímkou AD , leží bod B na polopřímce AE . Proto se zmíněná kružnice dotýká strany AE trojúhelníku AEF právě v bodě B . Body, v nichž se kružnice dotýká stran EF a FA , označme po řadě G a H (obr. 22). Označme ještě $a = |AB| = |EF|$ a necht $|AE| = ka$, $k > 1$. Ze sou-



Obr. 22

měrností dvojic tečen ke kružnici plynou rovnosti $|AH| = |AB| = a$, $|EG| = |EB| = |AE| - |AB| = (k-1)a$, $|FH| = |FG| = |EF| - |EG| = (2-k)a$, tudíž $|AF| = |AH| + |FH| = (3-k)a$. Pythagorova věta pro trojúhelník AEF tak dává rovnici $(3-k)^2 = k^2 + 1$, jež je po úpravě lineární a má (jediný) kořen $k = \frac{4}{3}$.

Odpověď: Hledaný poměr je 4 : 3.

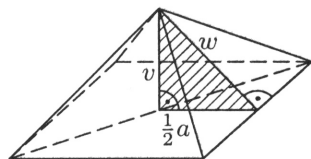
C – S – 3

Zmíněná tři dělení zapíšeme rovnostmi $N = 19a + b$, $a = 99c + d$ a $N = 1999c + d$. Odtud vyplývá, že $19(99c + d) + b = 1999c + d$, neboli $18d + b = 118c$. Nejmenší a největší vyhovující N najdeme podle nejmenšího a největšího možného neúplného podílu c (při dělení čísla N číslem 1999). Z rovnosti $18d + b = 118c$ plyne především, že $c > 0$ (kdyby bylo $c = 0$, bylo by $b = d = 0$, tedy i $N = 0$, ale N je kladné); pro $c = 1$ z rovnosti $18d + b = 118$ usoudíme, že $d = 6$ a $b = 10$ (neboť pro zbytek b při dělení $N : 19$ platí $0 \leq b \leq 18$). Proto je nejmenší N rovno číslu $1999 \cdot 1 + 6 = 2005$. Abychom zjistili největší N , poznamenejme nejdříve, že pro zbytky d a b při děleních $a : 99$ a $N : 19$ platí nerovnosti $d \leq 98$ a $b \leq 18$, z nichž plyne odhad $118c \leq 18 \cdot 98 + 18$, odkud $c \leq 15$. Pro $c = 15$ ovšem z rovnosti $18d + b = 118 \cdot 15$ vyplývá, že $d = 98$ a $b = 6$, neboť $0 \leq b \leq 18$ a $118 \cdot 15 = 1770 = 18 \cdot 98 + 6$. Největší N je tudíž rovno $1999 \cdot 15 + 98 = 30083$.

Odpověď: Nejmenší vyhovující N je 2005, největší takové N je 30083.

C – II – 1

Povrch vzniklého tělesa je tvořen všemi 24 bočními stěnami daných šesti jehlanů. Označme v výšku těchto jehlanů a a délku jejich podstavné hrany (jež je shodná s hranou dané krychle). Ze zadání úlohy vyplývá, že objem jednoho jehlanu je šestinou objemu krychle, tedy $\frac{1}{3}a^2v = \frac{1}{6}a^3$, odkud $v = \frac{1}{2}a$. Boční stěna jehlanu je rovnoramenný trojúhelník, pro jehož výšku w z hlavního vrcholu (obr. 23) platí podle Pythagorovy věty rovnost $w^2 = v^2 + (\frac{1}{2}a)^2$, odkud po dosazení $v = \frac{1}{2}a$ vychází $w = \frac{1}{2}\sqrt{2}a$. Proto je obsah boční stěny jehlanu roven $\frac{1}{2}aw = \frac{1}{4}\sqrt{2}a^2$. Výsledné těleso má povrch 24krát větší, tedy $6\sqrt{2}a^2$, zatímco povrch krychle je $6a^2$.



Obr. 23

Odpověď: Poměr povrchů původní krychle a výsledného tělesa je $1 : \sqrt{2}$.

Poznámka. Za daných předpokladů vznikne slepením těleso, které bude mít dvanáct shodných stěn (tzv. kosočtverečný dvanáctistěn). Uvedené shodné jehľany mají totiž v součtu stejný objem jako daná krychle a dostaneme je, když krychli rozdělíme na šest shodných jehľanů se společným hlavním vrcholem ve středu krychle.

C – II – 2

V daném výrazu odstraníme závorky a čísla sdružíme do čtveřic:

$$\begin{aligned} & \underbrace{1 - 2 - 3 + 5}_{+} + \underbrace{6 - 7 - 8 + 10}_{+} + \\ & + \underbrace{11 - 12 - 13 + 15}_{+} + \underbrace{16 - 17 - 18 + 20}_{+} + \dots \end{aligned}$$

(poslední skupina je kratší, není-li počet N všech čísel násobkem čtyř). Čísla v i -té skupině ($i = 1, 2, \dots$) tvoří výraz $(5i-4)-(5i-3)-(5i-2)+5i$, jehož hodnota je zřejmě rovna jedné (nezávisle na indexu i). Proto je hodnota V celého výrazu v případě $N = 4k$ rovna $V = k$, v případě $N = 4k + 1$ rovna $V = k + (5k + 1) = 6k + 1$, v případě $N = 4k + 2$ rovna $V = k + (5k + 1) - (5k + 2) = k - 1$ a v případě $N = 4k + 3$ rovna $V = k + (5k + 1) - (5k + 2) - (5k + 3) = -4k - 4$. Snadno se zjistí, kdy $V = 103$:

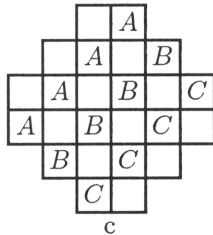
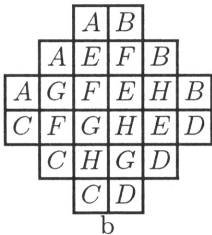
$N = 4k,$	$V = k = 103,$	$N = 412,$
$N = 4k + 1,$	$V = 6k + 1 = 103,$	$k = 17, \quad N = 69,$
$N = 4k + 2,$	$V = k - 1 = 103,$	$k = 104, \quad N = 418,$
$N = 4k + 3,$	$V = -4k - 4 = 103,$	$k \notin \mathbb{N}.$

Odpověď: V Milanově výrazu bylo buď 69, nebo 412, nebo 418 čísel.

C – II – 3

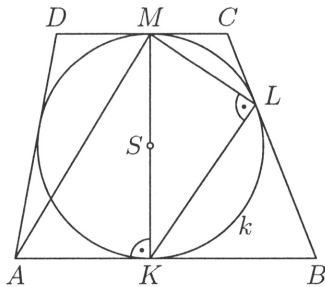
Příklad z obr. 24a ukazuje, že je možno požadovaným způsobem rozestavit 16 figurek (obsazená pole jsou označena křížky). Vysvětlíme nyní, proč více figurek rozestavit nelze. Na obr. 24b vidíte rozdělení všech polí desky do osmi šikmých řad po třech polích, když pole téže řady-trojice jsou označena stejným písmenem. Figurkami je možno obsadit nejvýše dvě pole v každé řadě-trojici (2 pole A , 2 pole B , $\dots$, 2 pole H), celkem nejvýše $8 \cdot 2 = 16$ polí.

a



Odpověď: Největší možný počet rozmístěných figurek je 16.

Označme k vepsanou kružnici, S její střed a K bod dotyku kružnice k se základnou AB (obr. 25). Protože $AB \parallel CD$, $SK \perp AB$ a $SM \perp CD$, je



KM průměr kružnice k . Proto jsou oba úhly AKM a KLM pravé, bod K tudíž sestrojíme jako průsečík Thaletovy kružnice τ nad průměrem AM s přímkou p , jež prochází bodem L a je kolmá na LM . Zbytek konstrukce je snadný: bod S určíme jako střed úsečky KM , sestrojíme kružnici

46

Kategorie B

Texty úloh

B – I – 1

Pro která reálná čísla t má funkce $f(x) = 5x + 44 + t|x - 2| - 3|x - t|$ maximum rovné 0?
(P. Černek)

B – I – 2

Označme S střed kružnice vepsané libovolnému trojúhelníku ABC . Dokažte, že rovnost $|AS| \cdot |BS| = |CS| \cdot |AB|$ platí, právě když úhel ACB je pravý.
(J. Švrček)

B – I – 3

Určete reálná čísla a, b , pro která má soustava

$$\begin{aligned}x^2 + y^2 + 2z^2 &= 16, \\xyz^2 + xy + z^2 &= a, \\x + y + 2z &= b\end{aligned}$$

v oboru reálných čísel právě jedno řešení.
(J. Bábelá)

B – I – 4

Jsou dány kružnice k a l s různými poloměry, které se vně dotýkají v bodě T . Průsečíkem M jejich společných vnějších tečen vedme sečnu s obou kružnic. Označme X ten z obou průsečíků kružnice k se sečnou s , který je vzdálenější od bodu M . Podobně označme Y ten z obou průsečíků kružnice l se sečnou s , který je vzdálenější od bodu M . Nechť P je takový bod, že $XTYP$ je rovnoběžník. Určete množinu bodů P odpovídajících všem takovým sečnám s .
(J. Zhouf)

B – I – 5

Devítistěn $ABCDEFGHV$ vznikl slepením krychle $ABCDEFGH$ a pravidelného čtyřbokého jehlanu $EFGHV$. Na každou stěnu tohoto devítistěnu jsme napsali číslo. Čtyři z napsaných čísel jsou 25, 32, 50 a 57. Pro každý vrchol devítistěnu $ABCDEFGHV$ sečteme čísla na všech stěnách, které ho obsahují. Dostaneme tak devět stejných součtů. Určete zbývajících pět čísel napsaných na stěnách tohoto tělesa.

(K. Černeková)

B – I – 6

Je dán rovnostranný trojúhelník XYZ s těžištěm T a stranou délky 5 cm. Sestrojte rovnoběžník $ABCD$ s obsahem 8 cm^2 a stranou AB délky 2 cm tak, aby body X, Y, Z, T ležely po řadě na přímkách AB, BC, CD, DA .

(M. Krállová)

B – S – 1

Pro která reálná čísla a, b je funkce

$$f(x) = a|x-1| + b(x-3) + |x-b| + x-1$$

omezená?

(J. Bábeľa)

B – S – 2

Je dána úsečka XZ délky 7 cm a její body S, Y tak, že $|XS| = 2\text{ cm}$, $|YZ| = 1\text{ cm}$. Sestrojte pravoúhlý trojúhelník ABC s přeponou AB tak, aby bod S byl střed kružnice vepsané trojúhelníku ABC a body X, Y, Z ležely po řadě na přímkách AC, AB, BC .

(P. Černek)

B – S – 3

Do výrazu

$$1 - 2 + 3 - 4 + 5 - 6 + \dots + 99 - 100$$

jsme vepsali několik závorek tak, že nakonec jsou v každé dvojici odpovídajících si závorek právě tři čísla a výraz neobsahuje žádný součin. Kolik různých výsledků lze takto dostat?

(P. Černek)

B – II – 1

Zjistěte všechna reálná čísla c , pro která má rovnice

$$(c^2 + c - 8)(x + 2) - 8|x - c + 2| = c|x + c + 14|$$

nekonečně mnoho řešení v oboru celých čísel.

(*J. Šimša*)

B – II – 2

Devítistěn vznikl slepením krychle a pravidelného čtyřbokého jehlanu. Na každé stěně tohoto devítistěnu je napsáno jedno číslo. Jejich součet je 3 003. Pro každou stěnu S uvažovaného devítistěnu sečteme čísla na všech stěnách, s nimiž má S společnou právě jednu hranu. Dostaneme tak devět stejných součtů. Určete všechna čísla napsaná na stěnách devítistěnu.

(*K. Černeková*)

B – II – 3

Je dán lichoběžník $ABCD$, v němž $|AB| = 8$ cm a $|\sphericalangle ABC| = 90^\circ$. Jeho obvod je 28 cm. Polokružnice k s průměrem AB se dotýká strany CD . Vypočítejte délky zbývajících stran daného lichoběžníku, je-li strana AB jeho

- a) základnou,
- b) ramenem.

(*Smutná*)

B – II – 4

Je dán obdélník $KLMN$, $|KN| > |KL|$. Sestrojte rovnoramenný trojúhelník ABC se základnou AB délky $|KL|$ tak, aby jeho výška v_a obsahovala body K , N , výška v_b bod L a výška v_c bod M . (Výškami zde rozumíme přímkami.)

(*K. Černeková*)

B – I – 1

Daná funkce je lineární lomená, protože obsahuje dva výrazy s absolutní hodnotou, které způsobují, že jejím grafem není přímka, nýbrž lomená čára. Její definiční obor, množinu $\mathbb{R}$ všech reálných čísel, můžeme v tomto případě rozdělit na tři disjunktní části podle toho, jak se příslušná absolutní hodnota chová (zda je výraz v absolutní hodnotě kladný či záporný). Protože jedna z absolutních hodnot závisí na parametru t , rozlišíme, zda je $t < 2$ (případ A), či $t \geq 2$ (případ B).

A. Nechť $t < 2$. Množina $\mathbb{R}$ se rozpadne na tři disjunktní intervaly, $\mathbb{R} = (-\infty, t) \cup (t, 2) \cup (2, \infty)$.

(a) V intervalu $(-\infty, t)$ je, jak snadno spočteme, $f(x) = (8 - t)x + 44 - t$. Protože za uvedeného předpokladu je $8 - t > 0$, je funkce f v tomto intervalu rostoucí a nabyde maxima v bodě $x = t$.

(b) V intervalu $(t, 2)$ je $f(x) = (2 - t)x + 44 + 5t$. Protože za uvedeného předpokladu je $2 - t > 0$, je funkce f i v tomto intervalu rostoucí a nabyde maxima v bodě $x = 2$. Přitom zřejmě platí $f(t) < f(2) = 2(2 - t) + 44 + 5t$.

(c) V intervalu $(2, \infty)$ je $f(x) = (2 + t)x + 44 + t$. Tato funkce je pro $2 + t > 0$ na tomto intervalu rostoucí a shora neomezená, takže nemůže mít maximum. Musí tedy nutně být $2 + t \leq 0$, tj. $t \leq -2$, funkce f bude v intervalu $(2, \infty)$ nerostoucí a její hodnota nebude větší než $f(2)$, kterou jsme spočítali v (b).

Zjistili jsme tedy, že za předpokladu $t < 2$ nabývá funkce f maxima jedině pro $t \leq -2$, přičemž její maximum je $f(2) = 2(2 - t) + 44 + 5t$. Toto maximum se rovná 0, právě když $2(2 - t) + 44 + 5t = 0$, neboli $t = -16$, což je našťástí číslo, které splňuje podmínku $t \leq -2$.

B. Nechť $t \geq 2$. Množina $\mathbb{R}$ se nám rozpadne na tři disjunktní intervaly, $\mathbb{R} = (-\infty, 2) \cup (2, t) \cup (t, \infty)$, přičemž „prostřední“ interval bude prázdný pro $t = 2$ (to však není pro další úvahy podstatné, jinak bychom mohli tento případ snadno rozebrat samostatně).

V intervalu $(-\infty, 2)$ je $f(x) = (8 - t)x + 44 - t$. Kdyby teď bylo $8 - t < 0$, byla by funkce f v tomto intervalu klesající a shora neomezená, takže by nemohla mít maximum. Proto je $8 - t \geq 0$, tj. $t \leq 8$. Pak ale je $f(2) = 2(8 - t) + 44 - t = 60 - 3t > 0$. Odtud hned vidíme, že za uvedeného předpokladu nemůže funkce f nikdy mít maximum rovné 0.

Z uvedeného rozboru vyplývá, že uvažovaná funkce má maximum rovné 0 jedině pro $t = -16$.

Jiné řešení. Grafem dané funkce f je lomená čára, která se v našem případě skládá ze dvou polopřímek (pro $t = 2$), resp. ze dvou polopřímek a jedné úsečky.

Dále bychom si měli uvědomit, že pokud má takováto funkce maximum, nabývá ho určitě v některém ze „zlomových“ bodů (tam, kde je příslušný výraz v absolutní hodnotě nulový). To samozřejmě neznamená, že funkce nemůže maximum nabýt i v jiných bodech (např. je-li konstantní na některém intervalu).

V našem případě jsou těmito zlomovými body pro $x = 2$ bod $A[2, 54 - 3|t - 2|]$, pro $x = t$ bod $B[t, 5t + 44 + t|t - 2|]$.

Protože jeden z bodů $x = 2$, $x = t$ má být bodem maxima funkce f rovného 0, zjistíme, pro která t je jedna z y -ových souřadnic bodů A a B nulová (a druhá nekladná).

$$\begin{aligned} A: \quad 54 - 3|t - 2| &= 0, \\ |t - 2| &= 18, \\ t &= 20 \text{ anebo } t = -16. \end{aligned}$$

$$\begin{aligned} B: \quad 5t + 44 + t|t - 2| &= 0, \\ t \geq 2 &\Rightarrow t^2 + 3t + 44 = 0, \\ &\text{nemá řešení.} \\ t < 2 &\Rightarrow t^2 - 7t - 44 = 0. \\ t &= 11 \text{ anebo } t = -4, \\ &\text{vyhovuje jen } t = -4. \end{aligned}$$

Máme tak tři možnosti:

Pro $t = 20$ je $A[2, 0]$, $B[20, 504]$, což nevyhovuje.

Pro $t = -16$ je $A[2, 0]$, $B[-16, -80 + 11 - 16 \cdot 18]$, zatím vyhovuje.

Pro $t = -4$ je $A[2, 36]$, $B[-4, 0]$, což nevyhovuje.

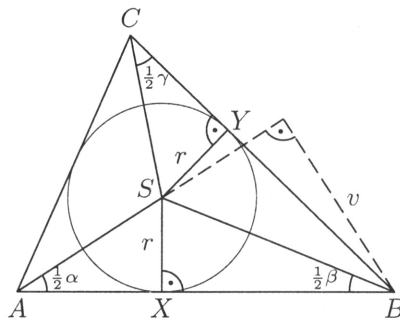
Zjistili jsme, že úloha má řešení nejvýše pro $t = -16$, kterému odpovídá funkce $f(x) = 5x + 44 - 16|x - 2| - 3|x + 16|$. Pro tuto funkci samozřejmě platí $f(2) = 0$. Ověřit, že tato hodnota je skutečně maximum funkce f , můžeme více způsoby. Například tak, že ověříme, že pro $x < -16$ je uvedená funkce neklesající (pro $x < -16$ je $f(x) = 24x + 60$) a současně pro $x > 2$ nerostoucí (pro $x > 2$ je $f(x) = -14x + 28$).

B – I – 2

Úhly v obecném trojúhelníku ABC označme obvyklým způsobem, poloměr vepsané kružnice označme r a její dotykové body se stranami AB , BC označme po řadě X , Y (obr. 27).

Úsečky AS a BS jsou stranami trojúhelníku ASB . Jeho obsah P můžeme vyjádřit dvěma způsoby:

$$P = \frac{1}{2}|AS|v = \frac{1}{2}|AB|r,$$



Obr. 27

neboť výška na stranu AB tohoto trojúhelníku je r ; pro výšku v na stranu AS přitom platí $v = |BS| \cos \frac{1}{2}\gamma$, protože vedlejší úhel při vrcholu S má velikost $\frac{1}{2}\alpha + \frac{1}{2}\beta = 90^\circ - \frac{1}{2}\gamma$. Je tedy

$$|AS| \cdot |BS| \cos \frac{\gamma}{2} = |AB|r$$

a následující rovnosti jsou ekvivalentní:

$$\begin{aligned} |AS| \cdot |BS| &= |CS| \cdot |AB|, \\ |AB|r &= |CS| \cdot |AB| \cos \frac{\gamma}{2}, \\ r &= |CS| \cos \frac{\gamma}{2}. \end{aligned} \quad (1)$$

V pravoúhlém trojúhelníku CSY však platí $\cos \frac{\gamma}{2} = \frac{|CY|}{|CS|}$, takže rovnost (1) je ekvivalentní rovnosti

$$r = |CY|,$$

což znamená, že trojúhelník CSY je rovnoramenný pravoúhlý a $\frac{1}{2}\gamma = 45^\circ$. Je tedy daná rovnost ekvivalentní tomu, že $\gamma = 90^\circ$.

Tím je tvrzení úlohy dokázáno.

Jiné řešení. Napišeme si daný vztah jako rovnost podílů tak, aby to byly poměry stran v trojúhelnících, a budeme se snažit použít podobnost nebo sinovou větu.

V našem případě vyjdeme z rovnosti $\frac{|AS|}{|CS|} = \frac{|AB|}{|BS|}$. Trojúhelníky ASC a BSC ale podobné nejsou, proto zkusíme sinovou větu:

V trojúhelníku ASC platí $\frac{|AS|}{|CS|} = \frac{\sin \frac{1}{2}\gamma}{\sin \frac{1}{2}\alpha}$ a v trojúhelníku ASB zase $\frac{|AB|}{|BS|} = \frac{\sin |\sphericalangle ASB|}{\sin \frac{1}{2}\alpha}$. Odtud dostáváme následující ekvivalentní rovnosti:

$$\begin{aligned}\frac{\sin \frac{1}{2}\gamma}{\sin \frac{1}{2}\alpha} &= \frac{\sin |\sphericalangle ASB|}{\sin \frac{1}{2}\alpha}, \\ \sin \frac{\gamma}{2} &= \sin |\sphericalangle ASB|, \\ \sin \frac{\gamma}{2} &= \sin \left(90^\circ + \frac{\gamma}{2}\right), \\ \frac{\gamma}{2} &= 180^\circ - \left(90^\circ + \frac{\gamma}{2}\right), \\ \gamma &= 90^\circ.\end{aligned}$$

Tím je tvrzení úlohy dokázáno.

Jiné řešení. Zkusíme vypočítat délky úseček AS , BS , CS , AB pomocí některých prvků trojúhelníku. My si zvolíme úhly trojúhelníku a poloměr r .

Zřejmě $|CS| = \frac{r}{\sin \frac{1}{2}\gamma}$, $|AS| = \frac{r}{\sin \frac{1}{2}\alpha}$, $|BS| = \frac{r}{\sin \frac{1}{2}\beta}$ a $|AB| = |AX| + |BX| = r \cotg \frac{1}{2}\alpha + r \cotg \frac{1}{2}\beta$. Po dosazení dostaneme ekvivalentní rovnosti

$$\begin{aligned}\frac{r}{\sin \frac{1}{2}\alpha} \cdot \frac{r}{\sin \frac{1}{2}\beta} &= \left(r \cotg \frac{\alpha}{2} + r \cotg \frac{\beta}{2}\right) \cdot \frac{r}{\sin \frac{1}{2}\gamma}, \\ \sin \frac{\gamma}{2} &= \cos \frac{\alpha}{2} \cdot \sin \frac{\beta}{2} + \cos \frac{\beta}{2} \cdot \sin \frac{\alpha}{2}, \\ \sin \frac{\gamma}{2} &= \sin \left(\frac{\alpha}{2} + \frac{\beta}{2}\right), \\ \sin \frac{\gamma}{2} &= \sin \left(90^\circ - \frac{\gamma}{2}\right), \\ \sin \frac{\gamma}{2} &= \cos \frac{\gamma}{2}, \\ \tg \frac{\gamma}{2} &= 1, \\ \gamma &= 90^\circ.\end{aligned}$$

Tím je tvrzení úlohy dokázáno.

Předpokládejme, že soustava má právě jedno řešení $x = s$, $y = t$, $z = u$. Protože ve všech rovnicích se neznámé x a y vyskytují ve stejném tvaru, lze vytušit a ověřit, že i $x = t$, $y = s$ a $z = u$ je řešením dané soustavy. A protože soustava má jediné řešení, musí být $t = s$, a tedy $x = y$. Po dosazení dostaneme soustavu

$$\begin{aligned}x^2 + z^2 &= 8, \\x^2 z^2 + x^2 + z^2 &= a, \\x + z &= \frac{1}{2}b.\end{aligned}\tag{*}$$

Pokud (x, z) je některé řešení této soustavy, je trojice (x, x, z) řešením původní soustavy. Má-li proto původní soustava jediné řešení, musí taková být i nová soustava (*). Ta je však opět symetrická vůči neznámým x a z . Proto bude mít jediné řešení, jen když bude platit $x = z$.

Po dosazení dostaneme soustavu

$$\begin{aligned}x^2 &= 4, \\x^4 + 2x^2 &= a, \\x &= \frac{1}{4}b,\end{aligned}$$

která má jediné řešení. Podle první rovnice je to buď $x = 2$ (pak $b = 8$, $a = 24$), anebo $x = -2$ (pak $b = -8$, $a = 24$).

Těmito úvahami jsme dospěli k následujícímu závěru:

Pokud má daná soustava právě jedno řešení, tak jen pro $a = 24$, $b = 8$, a to $x = 2$, $y = 2$, $z = 2$, anebo pro $a = 24$, $b = -8$, a to $x = -2$, $y = -2$, $z = -2$.

Ještě musíme ověřit, zda v těchto dvou případech nemá daná soustava jiné řešení (než to symetrické, které jsme vypočetli nikoli ekvivalentními úpravami, nýbrž zjednodušováním).

Nechť $a = 24$, $b = 8$. Po dosazení dostaneme soustavu

$$\begin{aligned}x^2 + y^2 + 2z^2 &= 16, \\xyz^2 + xy + z^2 &= 24, \\x + y + 2z &= 8.\end{aligned}$$

Tato soustava se dá řešit více způsoby. My tu uvedeme dva.

a) Vyloučíme neznámé x , y , například tak, že nejprve rovnice upravíme:

$$\begin{aligned}x^2 + y^2 &= 16 - 2z^2, \\ xy &= \frac{24 - z^2}{1 + z^2}, \\ x + y &= 8 - 2z.\end{aligned}$$

Dostáváme tak

$$(8 - 2z)^2 = (x + y)^2 = x^2 + y^2 + 2xy = 16 - 2z^2 + 2 \cdot \frac{24 - z^2}{1 + z^2}.$$

Po úpravě vychází

$$3z^4 - 16z^3 + 28z^2 - 16z = 0.$$

Vzhledem k tomu, že víme, že $z = 2$ je kořenem této rovnice, můžeme ji postupně upravit až na tvar

$$z(z - 2)^2(3z - 4) = 0.$$

Odtud plyne, že je buď $z = 0$, $z = \frac{4}{3}$, anebo $z = 2$.

Pokud $z = 0$, dostaneme

$$\begin{aligned}x^2 + y^2 &= 16, \\ xy &= 24, \\ x + y &= 8\end{aligned}$$

a snadno se přesvědčíme, že tato soustava nemá řešení (čísla x , y by musela být kořeny kvadratické rovnice $t^2 - 8t + 24 = 0$, která má záporný diskriminant).

Pokud $z = \frac{4}{3}$, dostaneme

$$\begin{aligned}x^2 + y^2 &= \frac{112}{9}, \\ xy &= 8, \\ x + y &= \frac{16}{3}\end{aligned}$$

a opět se snadno přesvědčíme, že tato soustava nemá řešení.

Pokud $z = 2$, dostaneme

$$x^2 + y^2 = 8,$$

$$xy = 4,$$

$$x + y = 4.$$

Snadno zjistíme, že tato soustava má jediné řešení $x = y = 2$.

b) Šikvnější přístup využívá jen první a třetí rovnici a nerovnost mezi kvadratickým a aritmetickým průměrem:

$$4 = 2^2 = \left(\frac{1}{4}(x + y + z + z) \right)^2 \leq \frac{1}{4}(x^2 + y^2 + z^2 + z^2) = 4.$$

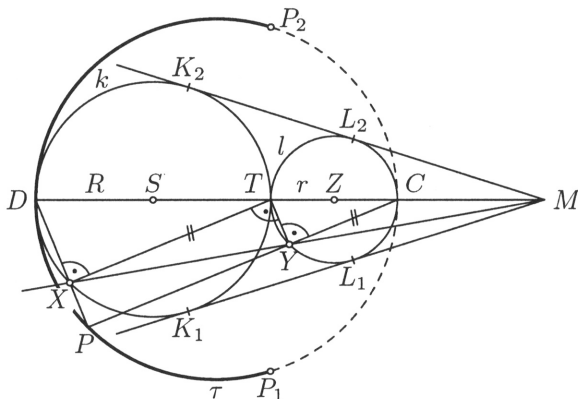
Mezi aritmetickým a kvadratickým průměrem nastane rovnost, právě když se všechny členy rovnají. Odtud $x = y = z = 2$.

Případ $a = 24$, $b = -8$ posoudíme podobně, i tehdy je řešení jediné.

Odpověď. Daná soustava má jediné řešení pro $a = 24$, $b = 8$ nebo $a = 24$, $b = -8$.

B – I – 4

Označme S , Z středy obou kružnic k , l a R , r jejich poloměry (bez újmy na obecnosti můžeme předpokládat, že $r < R$). Označme dále C a D od T různé průsečíky kružnic l a k s přímkou SZ a K_1 , K_2 , L_1 , L_2 po řadě dotykové body obou společných vnějších tečen ke kružnicím k a l (obr. 28).



Obr. 28

Bod M je středem stejnolehlosti h obou kružnic s koeficientem R/r . Přitom je například $h(L_1) = K_1$, $h(Z) = S$, $h(C) = T$, $h(T) = D$, $h(Y) = X$. Odtud plyne, že přímky CY , TX jsou rovnoběžné ($h(CY) = TX$). Protože úhel CYT je pravý podle Thaletovy věty, je také $|\sphericalangle YTX| = 90^\circ$ (TY je příčka rovnoběžek CY , TX). Rovnoběžník $XTYP$ je tedy vždy obdélník.

Zároveň je zřejmé, že body C , Y , P leží v přímce a podobně i body D , X , P leží v přímce. Je tudíž $|\sphericalangle CPD| = 90^\circ$ a bod P leží na Thaletově kružnici τ nad průměrem CD . Leží na ní i vrcholy P_1 , P_2 rovnoběžníků $K_1TL_1P_1$, $K_2TL_2P_2$, protože pro ně můžeme zopakovat předchozí úvahu (jako pro rovnoběžník $XTYP$).

Nyní už není problém ukázat, že hledanou množinou bodů je větší z oblouků P_1P_2 kružnice τ vyjma body P_1 , P_2 a D (neboť body Y tvoří větší z oblouků L_1L_2 kružnice l vyjma body T , L_1 , L_2).

Ještě naznačíme hlavní myšlenky *jiných dvou přístupů*:

a) Abychom odhadli tvar hledané množiny, zvolíme několik význačných poloh přímky XY . Vhodné jsou následující polohy: a) $X = K_1$, $Y = L_1$ (PT je kolmé na SZ), b) XS a YZ jsou kolmé na SZ (tehdy vyjde, že pata kolmice z bodu P na SZ leží ve středu J úsečky CD a $|JC| = |JP|$).

Z toho už se dá odhadnout, že bod P leží nejspíš na kružnici se středem J a poloměrem $\frac{1}{2}(R + r)$. Zbývá už jen dokázat (tedy obecně vypočítat), že vzdálenost $|PJ|$ je rovna $\frac{1}{2}(r + R)$. (Není to lehké.)

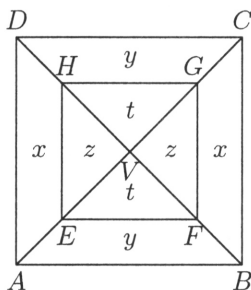
b) Pomocí shodných a podobných zobrazení je nejelegantnější následující postup: Pomocí souřadnic (bod M zvolíme za počátek souřadného systému) je $P = X + Y - T = Y + h(Y) - T = Y + (R/r)Y - T = (1 + R/r)Y - T$, bod P tedy vznikne z bodu Y (a všechny body Y tvoří větší z oblouků L_1L_2 kružnice l bez bodů T , L_1 , L_2) složením stejnolehlosti se středem M a koeficientem $1 + R/r$ a posunutí o vektor TM .

B – I – 5

Protože dva sousední vrcholy leží vždy ve dvou společných stěnách, budeme si všimát především takovýchto dvojic vrcholů.

Vrcholy A a B (B a C) leží ve společných stěnách $ABFE$ a $ABCD$ ($BCGF$ a $BCDA$). Proto porovnáním jim přiřazených čísel dostaneme, že na stěnách $ADHE$ a $BCGF$ ($ABFE$ a $CGHD$) je stejné číslo. Označme ho x (y).

Podobně vrcholy E a F (F a G) leží ve společných stěnách $EFBA$ a EFV ($FGCB$ a FGV) a navíc už víme, že stěny $ADHE$ a $FBCG$ ($ABFE$ a $GHDC$) mají stejná čísla. Proto porovnáním jim přiřazených čísel dostaneme, že na stěnách HEV a FGV (EFV a GHV) je stejné číslo. Označme ho z (t , obr. 29).



Obr. 29

Porovnáním čísel příslušných vrcholům A a E (mají společné stěny $EABF$ a $EADH$) dále dostaneme, že stěna $ABCD$ má číslo $s = z + t$.

Nakonec porovnejme vrcholy E a V (mají společné stěny EFV a HEV). Dostaneme $z + t = x + y$.

Když to vše shrneme, zjistíme, že jednotlivé stěny jsou nutně očíslovány čísla x (stěny $BCGF$ a $DAEH$), z (stěny FGV a EHV), s (stěna $ABCD$), $s - x$ (stěny $ABFE$ a $CDHG$), $s - z$ (stěny EFV a GHV). A snadno se přesvědčíme, že takovéto očíslování má vždy požadovanou vlastnost (všem vrcholům je přiřazeno číslo $2s$).

My známe čtyři různá čísla z devíti čísel $x, x, z, z, s, s - x, s - x, s - z, s - z$, tedy čtyři čísla z pěti čísel $x, z, s, s - x, s - z$.

a) Pokud je neznámé páté číslo s , tvoří známá čísla dvě dvojice se stejným součtem: $x + (s - x) = z + (s - z)$. Pro daná čísla tak máme jedinou možnost $25 + 57 = 32 + 50 = 82$. Hledaná čísla jsou pak 25, 32, 50, 57 a 82.

b) Není-li páté neznámé číslo s , je jedno známé číslo (a to s) součtem dalších dvou známých: $s = x + (s - x)$, nebo $s = z + (s - z)$. Pro daná čísla je jediná možnost: $25 + 32 = 57$. Potom je $s = 57$ a hledanou pěticí tvoří čísla 7, 7, 25, 32 a 50.

Odpověď. Hledaná čísla jsou buď 25, 32, 50, 57 a 82, nebo čísla 7, 7, 25, 32 a 50.

Ještě naznačíme, jak by mohl vypadat čistě *algebraický přístup* — řešením devíti rovnic o deseti neznámých.

Kvůli přehlednosti si musíme dát záležet na označení jednotlivých čísel napsaných na stěnách. Označme čísla na stěnách $ABFE$, $BCGF$, $CDHG$, $DAEH$, EFV , FGV , GHV , HEV a $ABCD$ postupně a_1 , a_2 , a_3 , a_4 , b_1 , b_2 , b_3 , b_4 , c a necht' společný součet na stěnách při každém vrcholu je s . Dostaneme tak následujících devět rovnic:

$$a_1 + a_2 + b_1 + b_2 = s, \quad (\text{F})$$

$$a_2 + a_3 + b_2 + b_3 = s, \quad (\text{G})$$

$$a_3 + a_4 + b_3 + b_4 = s, \quad (\text{H})$$

$$a_4 + a_1 + b_4 + b_1 = s, \quad (\text{E})$$

$$a_1 + a_2 + c = s, \quad (\text{B})$$

$$a_2 + a_3 + c = s, \quad (\text{C})$$

$$a_3 + a_4 + c = s, \quad (\text{D})$$

$$a_4 + a_1 + c = s, \quad (\text{A})$$

$$b_1 + b_2 + b_3 + b_4 = s. \quad (\text{V})$$

Porovnáním rovnic (B) a (C) máme $a_1 = a_3$. Porovnáním rovnic (D) a (C) máme $a_2 = a_4$.

Pomocí těchto vztahů dále dostaneme: porovnáním rovnic (F) a (G) vyjde $b_1 = b_3$; porovnáním rovnic (G) a (H) vyjde $b_2 = b_4$.

To znamená, že pro čísla a_1 , a_2 , b_1 , b_2 a c zůstaly rovnice

$$a_1 + a_2 + b_1 + b_2 = s,$$

$$a_1 + a_2 + c = s,$$

$$b_1 + b_2 = \frac{1}{2}s.$$

Odtud už snadno dostaneme, že $c = a_1 + a_2 = b_1 + b_2 = \frac{1}{2}s$.

B – I – 6

Podstatou řešení jsou následující dvě úlohy:

A. Jsou dány body K , L . Veďte jimi po řadě rovnoběžky k , l , je-li dána jejich vzdálenost d .

B. Jsou dány body K , L a přímka m . Veďte body K , L po řadě rovnoběžky k , l , které na přímce m vytínají úsečku dané délky d .

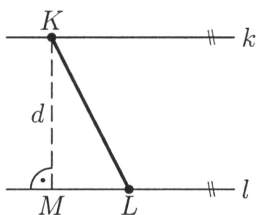
Řešení úlohy A (obr. 30). Nechť M je pata kolmice vedené z bodu K na přímkou l . V trojúhelníku KLM s pravým úhlem při vrcholu M známe vrcholy K , L a délku odvěsny $|KM| = d$, vrchol M tedy umíme sestavit (jako průsečík Thaletovy kružnice t nad průměrem KL s kružnicí $\kappa(K; d)$). Potom ML je přímka l .

Pokud bychom požadovali diskusi, víme, že počet řešení závisí na existenci průsečíku kružnic t a κ :

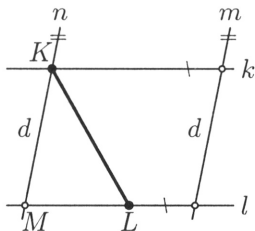
Pokud $|KL| < d$, nemá úloha řešení.

Pokud $|KL| = d$, má úloha jedno řešení (kolmice na KL).

Pokud $|KL| > d$, mají kružnice k a t dva průsečíky, takže úloha má dvě řešení.



Obr. 30



Obr. 31

Řešení úlohy B (obr. 31). Vedme bodem K rovnoběžku n s přímkou m a označme M její průsečík s přímkou l . Potom $|KM| = d$, takže konstrukce bodu M je zřejmá. Přímka l je pak určena body L a M .

Pokud bychom požadovali diskusi, snadno zjistíme, že na přímce n existují dva body M požadovaných vlastností, a počet řešení závisí na tom, zda $M = L$.

Pokud současně neplatí, že KL je rovnoběžná s m a $|KL| = d$, má úloha dvě řešení.

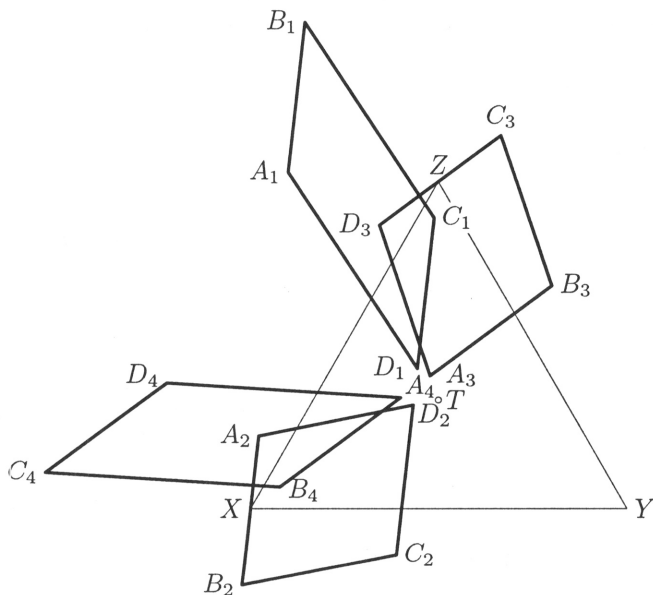
Pokud je KL rovnoběžná s m a $|KL| = d$, vznikne pro jednu z možných poloh bodu M v předcházejícím případě nekonečně mnoho řešení (za přímkou l můžeme vzít libovolnou přímku procházející bodem L).

Řešení původní úlohy. Z obsahu rovnoběžníku $ABCD$ a délky strany AB snadno vypočítáme výšku v na stranu AB : je $v = 8 \text{ cm}^2 : 2 \text{ cm} = 4 \text{ cm}$. Odtud plyne, že vzdálenost rovnoběžek AB a CD je 4 cm, přičemž známe bod X přímky AB a bod Z přímky CD . Podle úlohy A tedy umíme sestavit přímky AB a CD .

V poloze, která je dána, má tato část dvě řešení.

Když už máme přímku AB , jsou AD a BC dvě neznámé rovnoběžky, které procházejí danými body T a Y a na (známé) přímce AB vytínají úsečku dané délky $|AB| = 2$ cm. Proto můžeme rovnoběžky AD a BC sestrojit na základě úlohy B.

Je zřejmé, že speciální poloha daných bodů X, Y, Z a T nemá na postup řešení vliv, zaručuje nám však snadnou diskusi počtu řešení. Pro obě polohy přímky AB má úloha v dané situaci dvě řešení. Tím je rovnoběžník $ABCD$ sestrojen. (Přímkami AB, BC, CD a AD jsou vrcholy A, B, C, D určeny.) Úloha má 4 řešení (obr. 32).



Obr. 32

B – S – 1

Uvažovaná funkce f je po částech lineární, proto je omezená na každém omezeném intervalu. Stačí tedy funkci f vyšetřit zvlášť pro $x \leq \min(1, b)$ a zvlášť pro $x \geq \max(1, b)$, kdy mají oba výrazy v absolutních hodnotách stejné znaménko.

a) Je-li $x \leq \min(1, b)$, je

$$f(x) = a(1 - x) + b(x - 3) + b - x + x - 1 = (b - a)x - 2b + a - 1.$$

Funkce f bude na tomto intervalu omezená, právě když zde bude konstantní, tj. právě když $a = b$.

a) Je-li $x \geq \max(1, b)$, je

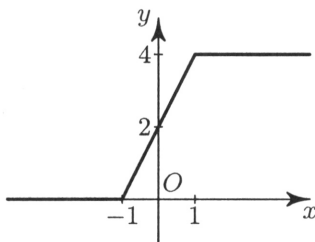
$$f(x) = a(x-1) + b(x-3) + x - b + x - 1 = (a+b+2)x - a + 4b - 1.$$

Funkce f bude na tomto intervalu omezená, právě když tu bude konstantní, tj. právě když $a + b = -2$.

Spojením obou podmínek dostáváme, že funkce f bude omezená, právě když bude omezená na obou uvedených neomezených intervalech, tj. právě když $a = b = -1$. Pro funkci f pak dostaneme vyjádření

$$f(x) = |x+1| - |x-1| + 2.$$

Její graf vidíme na obr. 33.

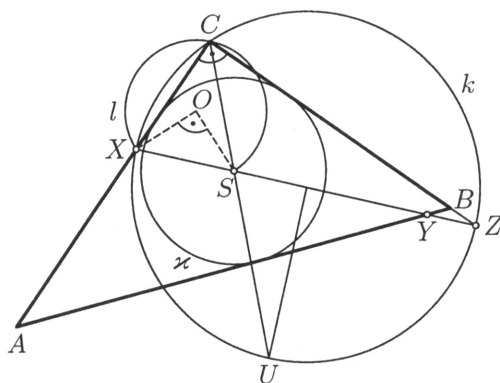


Obr. 33

B – S – 2

Předpokládejme, že trojúhelník ABC je řešením úlohy. Z daného pořadí bodů X, S, Y, Z na jedné přímce a z toho, že bod S je vnitřním bodem trojúhelníku ABC , vyplývá, že body X a Y jsou vnitřními body příslušných stran AC a AB , zatímco bod Z musí ležet na polopřímce opačné k polopřímce BC . Vrchol C neznámého trojúhelníku ABC budeme hledat jen v jedné z polorovin určených přímkou XZ , protože ke každému řešení existuje řešení souměrně sdružené podle osy XZ .

Vrchol C trojúhelníku ABC je vrcholem pravého úhlu XCZ (obr. 34), leží tedy na Thaletově kružnici k nad průměrem XZ . Protože bod S je středem kružnice vepsané trojúhelníku ABC , leží na ose pravého úhlu, takže $\angle SCX = 45^\circ$ a vrchol C leží zároveň na oblouku kružnice l určené tětivou SX a obvodovým úhlem 45° . (Vzhledem k uvedené souměrnosti



Obr. 34

stačí uvažovat jen ten ze dvou souměrných oblouků, který leží ve zvolené polorovině.) Odtud už plyne konstrukce trojúhelníku ABC :

1. sestrojíme kružnici k nad průměrem XZ ;
2. v jedné z polorovin určených přímkou XZ sestrojíme vrchol O rovnoarmenného pravoúhlého trojúhelníku XSO , $|\sphericalangle SOX| = 90^\circ$, a v téže polorovině narýsujeme oblouk SX kružnice $l(O, |OS|)$;
3. vrchol $C = k \cap \widehat{SX}$, $C \neq X$;
4. sestrojíme kružnici $\kappa(S; \varrho)$, kde ϱ je vzdálenost bodu S od přímky CX (poloměr kružnice vepsané trojúhelníku ABC);
5. bodem Y vedeme tečnu t ke kružnici κ tak, aby její bod dotyku ležel v polorovině opačné k polorovině XZC ;
6. vrcholy A, B dostaneme jako průsečíky přímky t s přímkami XC , resp. ZC .

Z popsané konstrukce je zřejmé, že pro bod S ležící mezi body X a Z mají kružnice k a l právě jeden průsečík různý od bodu X . Abychom mohli sestrojit tečnu t , musí bod Y ležet vně kruhu omezeného kružnicí κ , musí tedy být $|SY| \geq \varrho$. Aby existoval průsečík tečny t s přímkou XC uvnitř úhlu XCZ , musí být dokonce $|YS| > |XS| > \varrho$. V našem případě je to splněno a úloha má dvě shodná řešení souměrně sdružená podle osy XZ .

Úlohu bychom řešili stejně, i kdyby dané body X, S, Y, Z neležely na jedné přímce.

Poznámka. Bod C můžeme sestrojit i jiným postupem. Protože body X a Z leží po řadě na polopřímkách CA a CB , je pravý úhel XCZ

totožný s pravým úhlem ACB . Osa tohoto úhlu prochází středem S kružnice vepsané trojúhelníku ABC ; zároveň tato osa protne kružnici k opsanou trojúhelníku XCZ v takovém bodě $U \neq C$, že tětivy XU a UZ jsou shodné (tyto tětivy jsou totiž z bodu C vidět pod týmž úhlem (45 stupňů), obr. 34). Proto (aniž známe bod C) můžeme bod U sestrojít jako střed oblouku XZ kružnice k (oblouky XU a UZ jsou tedy čtvrtkružnice). Bod C pak určíme jako průsečík kružnice k s polopřímku US .

B – S – 3

V daném výrazu se pravidelně střídají plusy a minusy, přičemž lichá čísla mají znaménko plus a sudá minus. Zřejmě musí každá dvojice odpovídajících si závorek obsahovat právě tři po sobě jdoucí čísla. Umístíme-li levou závorku mezi plus a příslušné liché číslo, nemají závorky na hodnotu výrazu žádný vliv. Zajímavý je tedy jen případ, kdy levou závorku dáme mezi minus a následující sudé číslo, což změní výsledné znaménko druhého a třetího čísla v závorkách. Je-li zmíněné sudé číslo $2k$ ($1 \leq k \leq 49$), dostaneme místo původního součtu $-2k + (2k + 1) - (2k + 2) = -2k - 1$ součet $-(2k + (2k + 1) - (2k + 2)) = -2k - (2k + 1) + (2k + 2) = -2k + 1$. Vidíme tedy, že přidáním jednoho páru závorek popsáním způsobem zvětšíme celkovou hodnotu výrazu o 2 bez ohledu na to, kterou trojici po sobě jdoucích čísel začínající sudým číslem zvolíme. Zároveň je jasné, že takto umístěný pár závorek obsahuje další sudé číslo (kromě čísla $2k$ ještě $2k + 2$), které už nebudeme moci pro umístění závorek využít. (Nebudeme tedy zbytečně rozmisťovat závorky před lichá čísla, protože bychom se zbavili dalšího sudého čísla, před které lze umístit levou z dvojice závorek, jež by měly vliv na hodnotu daného výrazu.)

Daný výraz obsahuje celkem 50 sudých čísel. Můžeme tedy vybrat nejvýše 25 dvojic po sobě jdoucích sudých čísel, jež obklopíme závorkami. Tomu odpovídá 26 různých hodnot daného výrazu s k dvojicemi závorek, kde $0 \leq k \leq 25$. Příslušné hodnoty jsou $-50, -48, -46, \dots, 4, 2, 0$ (nejmenší hodnota je $1 - 2 + 3 - 4 + 5 - 6 + \dots + 99 - 100 = -50$, největší $1 - (2 + 3 - 4) + 5 - (6 + 7 - 8) + \dots - (98 + 99 - 100) = 0$).

B – II – 1

Označíme-li pro dané reálné c

$$f_c(x) = c|x + c + 14| + 8|x - c + 2| - (c^2 + c - 8)(x + 2)$$

odpovídající po částech lineární funkci, je zřejmé, že rovnice $f_c(x) = 0$ bude mít nekonečně mnoho celočíselných řešení, právě když bude funkce f_c identicky rovna nule na některém z nekonečných intervalů $(-\infty, \min(c-2, -c-14))$ nebo $(\max(c-2, -c-14), \infty)$. Vyšetříme postupně obě možnosti.

a) Nechť $x \leq \min(c-2, -c-14)$, pro taková x platí

$$\begin{aligned} f_c(x) &= -c(x+c+14) - 8(x-c+2) - (c^2+c-8)(x+2) = \\ &= -c(2+c)x - 3c^2 - 8c = -c(x(c+2) + 3c+8). \end{aligned}$$

Na tomto intervalu bude funkce f_c identicky rovna nule, právě když $c = 0$ (soustava $c+2=0$, $3c+8=0$ nemá žádné řešení).

b) Nechť $x \geq \max(c-2, -c-14)$, pro taková x platí

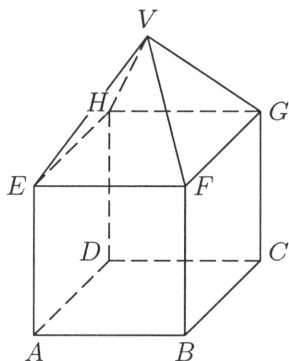
$$\begin{aligned} f_c(x) &= c(x+c+14) + 8(x-c+2) - (c^2+c-8)(x+2) = \\ &= (16-c^2)x - c^2 + 4c + 32. \end{aligned}$$

Na tomto intervalu bude funkce f_c identicky rovna nule, právě když bude současně platit $c^2 = 16$ a $c^2 - 4c - 32 = 0$. Dosazením $c^2 = 16$ do druhé rovnice vychází $c = -4$, což je zřejmě jediné řešení obou rovnic.

Závěr. Daná rovnice má v oboru celých čísel nekonečně mnoho řešení, právě když $c = 0$ nebo $c = -4$ (v prvním případě rovnici vyhovují všechna celá čísla $x \leq -14$, v druhém pak všechna celá čísla $x \geq -6$).

B – II – 2

Označme A, B, C, D, E, F, G, H vrcholy zmíněné krychle a V vrchol přilepeného jehlanu (obr. 35). Čísla napsaná na bočních stěnách $ABFE$,



Obr. 35

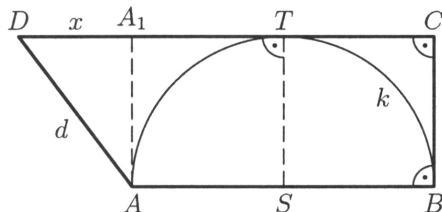
$BCGF$, $CDHG$, $DAEH$ označme postupně a_1 , a_2 , a_3 a a_4 , čísla na bočních stěnách EFV , FGV , GHV a HEV přilepeného jehlanu označme po řadě b_1 , b_2 , b_3 a b_4 , číslo na podstavě $ABCD$ označme c . Dále označme s uvedený společný součet.

Porovnáním součtů příslušných stěnám EFV a GHV dostaneme rovnost $a_1 = a_3$, analogicky pro další dvojici stěn vyjde $a_2 = a_4$. Porovnáním součtů příslušných stěnám $ABFE$ a $CDHG$ vyjde $b_1 = b_3$ a analogicky pro další takovou dvojici stěn $b_2 = b_4$. Porovnáním součtů příslušných stěnám $CDHG$ a HEV dostaneme rovnost $b_1 = c + a_2$ a analogicky pro dvojici stěn $DAHE$ a GHV rovnost $b_2 = c + a_1$. Porovnáním součtů dvou sousedních stěn krychle vychází $a_2 + a_4 + b_1 = a_1 + a_3 + b_2$, neboli $2a_2 + b_1 = 2a_1 + b_2$, což dosazením z posledních dvou získaných rovností dává rovnost $a_2 = a_1$, a tedy také $b_2 = b_1 = c + a_1$. Můžeme proto psát $a_1 = a_2 = a_3 = a_4 = a$, $b_1 = b_2 = b_3 = b_4 = b = a + c$ a z rovnosti součtů příslušných podstavě a jedné z bočních stěn krychle vychází $4a = 2a + b + c = 3a + 2c$, takže $a = 2c$, $b = 3c$ a celkový součet všech čísel je $c + 4a + 4b = 21c$. Z rovnice $21c = 3\,003$ plyne $c = 143$. Na stěnách devítistěnu jsou napsána čísla 143, 286 (čtyřikrát) a 429 (čtyřikrát).

Poznámka. Úlohu je možno řešit i vypsáním a následným řešením soustavy deseti lineárních rovnic pro devět neznámých čísel zapsaných na stěnách tělesa a desátou neznámou rovnou jednotnému součtu.

B – II – 3

Označme S střed strany AB a T bod dotyku polokružnice k se stranou CD . Jestliže je AB základnou daného lichoběžníku, je $CD \parallel AB$ a $|AB| + |BC| + |CT| = 2|AB| = 16$ cm (obr. 36). Označme A_1 kolmý

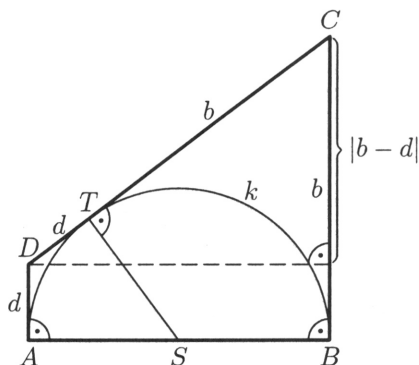


Obr. 36

průmět vrcholu A na přímku CD . Protože $|TD| + |DA| = 28$ cm – 16 cm = 12 cm $> |AA_1| + |A_1T| = 8$ cm, leží vrchol D na polopřímce TA_1 za bodem A_1 . Označme velikost $|A_1D| = x$ cm, $|DA| = d$ cm. Pro čísla x ,

d dostáváme soustavu rovnic $d + x = 8$, $d^2 = x^2 + 4^2$ (Pythagorova věta pro trojúhelník AA_1D), kterou snadno upravíme na tvar $d + x = 8$, $(d - x)(d + x) = 16$, tj. $d + x = 8$, $d - x = 2$. Soustava má jediné řešení $d = 5$, $x = 3$. Zbývající strany daného lichoběžníku mají tedy velikosti 4 cm, 11 cm a 5 cm.

Je-li AB ramenem daného lichoběžníku $ABCD$, je $AB \perp BC \parallel AD$ (obr. 37), takže obě základny BC a AD se dotýkají polokružnice k v od-



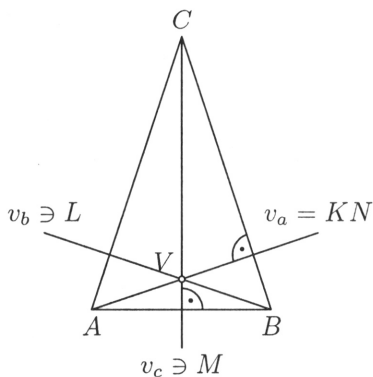
Obr. 37

povídajících vrcholech B a A . Označme b shodné úseky tečen z vrcholu C a d shodné úseky tečen z vrcholu D k polokružnici k . Ze znalosti obvodu tak dostáváme (v centimetrech) rovnost $28 = 8 + 2b + 2d$ neboli $b + d = 10$. Z rovnoběžnosti $BC \parallel AD$ plyne $|b - d| = \sqrt{(b + d)^2 - 8^2} = 6$. Vzhledem k souměrnosti podle osy dané polokružnice k stačí uvažovat jen jednu z možností, např. $b > d$. Soustava $b + d = 10$, $b - d = 6$ má jediné řešení $b = 8$, $d = 2$, takže zbývající strany daného lichoběžníku mají v tomto případě velikosti 8 cm, 10 cm a 2 cm, což platí i v případě $b < d$.

B – II – 4

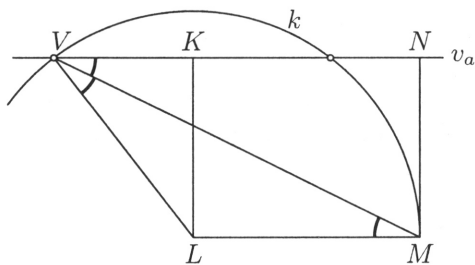
Podle zadání známe přímku KN , na níž leží výška v_a . Protože výška v_b je souměrně sdružená s v_a podle osy základny AB hledaného rovnoramenného trojúhelníku ABC , na níž zároveň leží jeho třetí výška v_c , pokusíme se najít průsečík V těchto výšek. Ten má tu vlastnost, že bod L leží na přímce souměrně sdružené s $v_a = KN$ podle $VM = v_c$ (obr. 38). Jakmile bod V najdeme, budeme zároveň znát polohu všech tří výšek trojúhel-

níku ABC , takže až na podobnost můžeme sestrojít i hledaný trojúhelník ABC .



Obr. 38

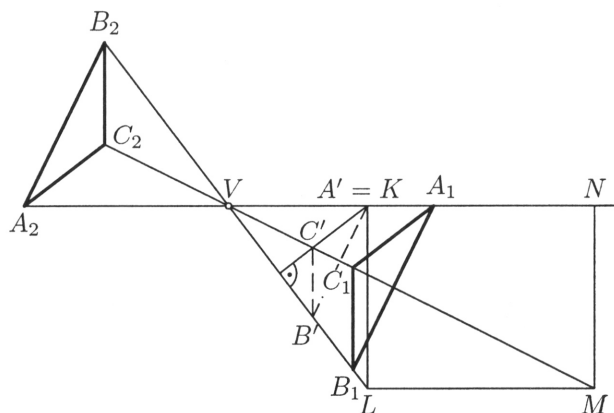
Předpokládejme, že bod V na přímkce KN má požadovanou vlastnost (obr. 39). Ze souměrnosti přímk VL a VN podle VM plyne rovnost vyznačených úhlů s vrcholem V . Z rovnoběžnosti přímk KN a LM dostáváme, že stejnou velikost má i úhel LMV , takže trojúhelník MVL je rovnoramenný se základnou MV . Je tudíž $|LV| = |LM|$ a bod V najdeme jako průsečík přímky KN s kružnicí $k(L; |LM|)$. Protože dle předpokladu je $|KL| < |KN| = |LM|$, existují takové průsečíky dva.



Obr. 39

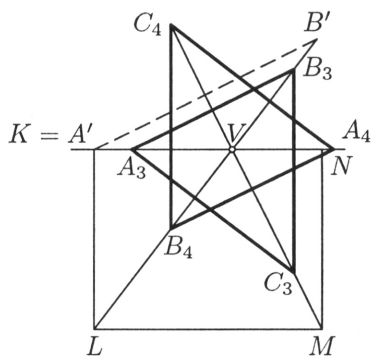
Nyní dokončíme konstrukci trojúhelníku ABC . Nejprve sestrojíme pomocný trojúhelník $A'B'C'$, který bude stejnohlý s hledaným trojúhelníkem ABC , a to tak, že na přímkce KN libovolně zvolíme bod $A' \neq V$ (na obr. 40 je jako bod A' zvolen daný vrchol K), sestrojíme bod B' souměrně sdružený s bodem A' podle VM a vrchol C' , v němž kolmice na $B'V$ vedená bodem A' protne přímkku VM . Protože má platit

$|AB| = |KL|$, trojúhelník ABC sestrojíme užitím té stejnolehlosti se středem V , která známou úsečku $A'B'$ převede na hledanou úsečku AB



Obr. 40

dané délky $|KL|$ (takové stejnoolehlosti jsou dvě). Pro každý z možných bodů V tak bude mít úloha dvě řešení (na obr. 40 trojúhelníky $A_1B_1C_1$ a $A_2B_2C_2$, na obr. 41 trojúhelníky $A_3B_3C_3$ a $A_4B_4C_4$) středově souměrná podle příslušného průsečíku výšek.



Obr. 41

Kategorie A

Texty úloh

A – I – 1

Nechť P , Q jsou kvadratické trojčleny takové, že tři z kořenů rovnice $P(Q(x)) = 0$ jsou čísla -22 , 7 , 13 . Určete čtvrtý kořen této rovnice.

(P. Černek)

A – I – 2

Nechť K , L , M jsou po řadě vnitřní body stran BC , CA , AB daného trojúhelníku ABC takové, že kružnice vepsané dvojicím trojúhelníků ABK a CAK , BCL a ABL , CAM a BCM mají vnější dotyk. Pak platí

$$|BK| \cdot |CL| \cdot |AM| = |CK| \cdot |AL| \cdot |BM|.$$

Dokažte.

(J. Švrček)

A – I – 3

V oboru kladných čísel řešte soustavu

$$\sqrt{xy} + \sqrt{xz} - x = a,$$

$$\sqrt{yz} + \sqrt{yx} - y = b,$$

$$\sqrt{zx} + \sqrt{zy} - z = c,$$

kde a , b , c jsou daná kladná čísla.

(R. Horenský)

A – I – 4

V rovině je dáno 1999 shodných trojúhelníků o obsahu 1, které jsou obrazy téhož trojúhelníku v různých posunutích. Je-li průnikem všech daných trojúhelníků množina M , která obsahuje těžiště každého z nich, je obsah množiny M alespoň $\frac{1}{9}$. Dokažte.

(M. Beneš)

A – I – 5

Je dána funkce $f: \mathbb{N} \rightarrow \mathbb{N}$ taková, že $f(n) = 1$, je-li n liché, a $f(n) = k$ pro každé sudé číslo $n = 2^k l$, kde k je přirozené číslo a l číslo liché. Určete největší přirozené číslo n , pro něž platí

$$f(1) + f(2) + \dots + f(n) \leq 123\,456.$$

(S. Trávníček)

A – I – 6

Je dán čtyřboký jehlan $ABCDV$ s podstavou $ABCD$. Jeho hrany AB , CD jsou rovnoběžné a roviny ABV a CDV vzájemně kolmé. Označme P patu výšky z vrcholu V na stranu AB v trojúhelníku ABV a Q patu výšky z vrcholu V na stranu CD v trojúhelníku CDV . Dokažte nerovnost

$$|AV|^2 + |BV|^2 + |CV|^2 + |DV|^2 \geq |PQ|^2 + 2(S_{ABV} + S_{CDV} + S_{PQV}),$$

kde S_{XYZ} značí obsah trojúhelníku XYZ . Zjistěte rovněž, kdy platí rovnost.

(J. Bábela)

A – S – 1

Určete, pro která reálná čísla p má soustava rovnic

$$\begin{aligned}(x - y)^2 &= p^2, \\ x^3 - y^3 &= 16\end{aligned}$$

právě jedno řešení v oboru reálných čísel.

(J. Bábela)

A – S – 2

Je dán trojúhelník ABC . Uvnitř jeho stran BC , CA , AB uvažujme po řadě body K , L , M takové, že úsečky AK , BL , CM se protínají v bodě U . Jestliže trojúhelníky AMU a KCU mají obsah P a trojúhelníky MBU a CLU obsah Q , pak $P = Q$. Dokažte.

(J. Švrček)

A – S – 3

Určete nejmenší přirozené číslo k , pro které platí: Vybereme-li libovolných k různých čísel z množiny $\{1, 2, 3, \dots, 2\,000\}$, pak mezi vybranými čísly existují dvě, jejichž součet nebo rozdíl je 667.

(J. Šimša)

A – II – 1

Nechť $P(x)$ je kvadratický trojčlen. Určete všechny kořeny rovnice

$$P(x^2 + 4x - 7) = 0,$$

víte-li, že mezi nimi je číslo 1 a aspoň jeden kořen je dvojnásobný.

(P. Černek)

A – II – 2

Je dán rovnoramenný lichoběžník $UVST$, v němž $3|ST| < 2|UV|$. Se-strojte rovnoramenný trojúhelník ABC se základnou AB tak, aby body B, C ležely na přímce VS , bod U na přímce AB a bod T byl těžištěm trojúhelníku ABC .

(P. Černek)

A – II – 3

Dokažte, že pro libovolná kladná čísla a, b platí nerovnost

$$\sqrt[3]{\frac{a}{b}} + \sqrt[3]{\frac{b}{a}} \leq \sqrt[3]{2(a+b)\left(\frac{1}{a} + \frac{1}{b}\right)}.$$

Zjistěte, kdy nastane rovnost.

(J. Šimša)

A – II – 4

Určete všechny konvexní čtyřúhelníky $ABCD$ s následující vlastností: Uvnitř čtyřúhelníku $ABCD$ existuje bod E takový, že každá přímka, která prochází tímto bodem a protíná strany AB a CD ve vnitřních bodech, dělí čtyřúhelník $ABCD$ na dvě části o stejném obsahu. Svou odpověď zdůvodněte.

(P. Černek, J. Švrček)

A – III – 1

Nechť n je přirozené číslo. Dokažte, že součet $4 \cdot 3^{2^n} + 3 \cdot 4^{2^n}$ je dělitelný třináctí, právě když n je sudé.

(J. Šimša)

A – III – 2

Je dán rovnoramenný trojúhelník ABC se základnou AB . Na jeho výšce CD je zvolen bod P tak, že kružnice vepsané trojúhelníku ABP a čtyřúhelníku $PECF$ jsou shodné; přitom bod E je průsečík přímky AP se

stranou BC a F průsečík přímky BP se stranou AC . Dokažte, že i kružnice vepsané trojúhelníkům ADP a BCP jsou shodné.

(*J. Šimša, K. Horák*)

A – III – 3

V rovině je dáno 2000 shodných trojúhelníků o obsahu 1, které jsou obrazy téhož trojúhelníku v různých posunutích. Každý z těchto trojúhelníků obsahuje těžiště všech zbývajících. Dokažte, že obsah sjednocení těchto trojúhelníků je menší než $\frac{22}{9}$.

(*P. Calábek*)

A – III – 4

Pro které kvadratické funkce f existuje taková kvadratická funkce g , že kořeny rovnice $g(f(x)) = 0$ jsou čtyři různé po sobě jdoucí členy aritmetické posloupnosti a současně i kořeny rovnice $f(x)g(x) = 0$?

(*P. Černek*)

A – III – 5

Monika zhotovila papírový model trojbokého jehlanu, jehož podstavou byl pravoúhlý trojúhelník. Když model rozřízla podél odvěsen podstavy a podél těžnice jedné ze stěn, vznikl po rozvinutí do roviny čtverec o straně a . Určete objem tohoto jehlanu.

(*P. Leischner*)

A – III – 6

Najděte všechna čtyřmístná čísla $\overline{abcd}$ (v desítkové soustavě), pro něž platí rovnost

$$\overline{abcd} + 1 = (\overline{ac} + 1)(\overline{bd} + 1).$$

(*J. Zhouf*)

A - I - 1

Vzhledem k tomu, že rovnice $P(Q(x)) = 0$ má reálný kořen, má kvadratická rovnice $P(x) = 0$ dva reálné kořeny r_1, r_2 (nevylučujeme, že $r_1 = r_2$). Mnohočlen $P(Q(x))$ lze proto zapsat ve tvaru

$$P(Q(x)) = a(Q(x) - r_1)(Q(x) - r_2),$$

kde a je reálné číslo $a \neq 0$. Rovnice $P(Q(x)) = 0$ má podle zadání čtyři reálné kořeny, proto každá z kvadratických rovnic $Q(x) - r_1 = 0$, $Q(x) - r_2 = 0$ musí mít dva reálné kořeny. Z Viètových vzorců plyne, že součet kořenů v obou kvadratických rovnicích je týž, neboť obě rovnice mají stejný koeficient u lineárního členu. Přitom tři ze čtyř reálných kořenů obou kvadratických rovnic $Q(x) - r_1 = 0$, $Q(x) - r_2 = 0$ jsou dle zadání čísla $-22, 7, 13$, čtvrtý kořen označme q . Dále mohou nastat tři možnosti:

- (i) Jedna z kvadratických rovnic má kořeny $-22, 7$, druhá má kořeny 13 a q . Pak platí $-22 + 7 = 13 + q$, tedy $q = -28$.
- (ii) Jedna z kvadratických rovnic má kořeny $-22, 13$, druhá má kořeny 7 a q . Pak platí $-22 + 13 = 7 + q$, tedy $q = -16$.
- (iii) Jedna z kvadratických rovnic má kořeny $13, 7$, druhá má kořeny -22 a q . Potom však platí $13 + 7 = -22 + q$, tedy $q = 42$.

Je zřejmé, že v každém z případů (i), (ii), (iii) existují příslušné kvadratické trojčleny $P(x)$ a $Q(x)$. Má-li mít jedna z kvadratických rovnic $Q(x) - r_1 = 0$, $Q(x) - r_2 = 0$ kořeny $-22, 7$ a druhá $13, -28$, položíme $Q(x) = x^2 + 15x$, $-r_1 = (-22) \cdot 7 = -154$, $-r_2 = 13 \cdot (-28) = -364$, $P(x) = (x - 154)(x - 364) = x^2 - 518x + 56\,056$. Obdobně lze postupovat ve zbývajících případech.

Čtvrtým kořenem rovnice $P(Q(x)) = 0$ může být kterékoli z čísel $-28, -16, 42$.

Jiné řešení. Úvahy o koeficientu u lineárního členu s využitím Viètových vztahů lze nahradit následující úvahou o grafech kvadratických funkcí.

Protože grafy kvadratických funkcí $f_1: y = Q(x) - r_1$ a $f_2: y = Q(x) - r_2$ mají tutéž osu souměrnosti a přitom existují čtyři reálné kořeny rovnice $P(Q(x)) = 0$, jsou tyto kořeny na ose x po dvou středově souměrné podle průsečíku os souměrnosti grafů obou funkcí f_1 a f_2 s osou x . Vzhledem k poloze daných tří kořenů na ose x lze dále uvažovat tři možnosti stejně jako v předcházejícím řešení. Např.

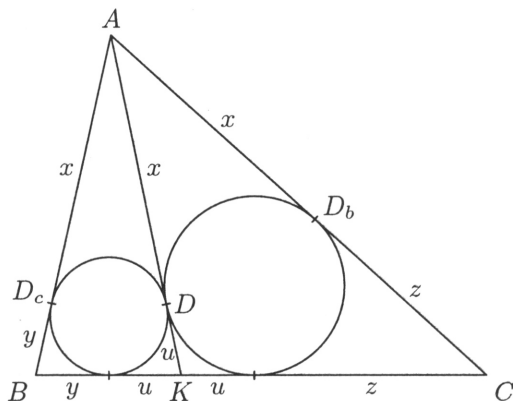
- (i) Střed souměrnosti je $-7,5 = \frac{1}{2}(-22+7)$, čtvrtý kořen leží na ose x a je symetrický s obrazem čísla 13 dle středu souměrnosti v bodě $-7,5$. Čtvrtým hledaným kořenem je tudíž číslo -28 .

Podobně lze postupovat ve zbylých dvou případech a dospějeme tak ke stejnému výsledku.

A - I - 2

Uvnitř strany BC trojúhelníku ABC uvažujme bod K takový, že kružnice vepsané trojúhelníkům BKA a CKA mají vnější dotyk v bodě D . Nechť dále (při obvyklém označení délek stran trojúhelníku ABC) platí označení podle obr. 42, tj.

$$|AD_b| = |AD_c| = x, |BD_c| = y, |CD_b| = z, |BK| = y+u, |CK| = z+u.$$



Obr. 42

Z předešlého obrázku snadno vidíme, že platí následující soustava rovnic

$$y + z = a - 2u,$$

$$z + x = b,$$

$$x + y = c.$$

Jednoduchou úpravou odtud dostáváme $2y + 2u = a - b + c$ (analogicky vyjádříme $2z + 2u$), a tudíž platí

$$|BK| = y + u = \frac{1}{2}(a - b + c) = s - b,$$

$$|CK| = z + u = \frac{1}{2}(a + b - c) = s - c,$$

kde $2s = a + b + c$. To značí, že bod K je bodem dotyku kružnice vepsané trojúhelníku ABC se stranou BC . Pro body L a M platí využitím analogického postupu následující vztahy:

$$|CL| = s - c, \quad |AL| = s - a, \quad |AM| = s - a, \quad |BM| = s - b.$$

Z předešlých rovností již bezprostředně plyne

$$|BK| \cdot |CL| \cdot |AM| = (s - a)(s - b)(s - c) = |CK| \cdot |AL| \cdot |BM|.$$

Tím je důkaz ukončen.

Poznámka. Úsečky AK , BL , CM vyhovující podmínkám úlohy se tedy protínají podle Čèvovy věty v jediném bodě G , zvaném *Gergonnův* bod daného trojúhelníku ABC .

A – I – 3

Z textu úlohy plyne, že neznámé x , y , z jsou kladná čísla, lze proto danou soustavu upravit do následujícího tvaru

$$\begin{aligned} -\sqrt{x} + \sqrt{y} + \sqrt{z} &= \frac{a}{\sqrt{x}}, \\ \sqrt{x} - \sqrt{y} + \sqrt{z} &= \frac{b}{\sqrt{y}}, \\ \sqrt{x} + \sqrt{y} - \sqrt{z} &= \frac{c}{\sqrt{z}}. \end{aligned}$$

Sečteme-li po dvojicích jednotlivé rovnice předešlé soustavy, dostaneme tak soustavu

$$\begin{aligned} \frac{b}{\sqrt{y}} + \frac{c}{\sqrt{z}} &= 2\sqrt{x}, \\ \frac{c}{\sqrt{z}} + \frac{a}{\sqrt{x}} &= 2\sqrt{y}, \\ \frac{a}{\sqrt{x}} + \frac{b}{\sqrt{y}} &= 2\sqrt{z}. \end{aligned}$$

Dále po snadné úpravě

$$\begin{aligned} b\sqrt{z} + c\sqrt{y} &= 2\sqrt{xyz}, \\ c\sqrt{x} + a\sqrt{z} &= 2\sqrt{xyz}, \\ a\sqrt{y} + b\sqrt{x} &= 2\sqrt{xyz}. \end{aligned}$$

Odečtením první a třetí, resp. druhé a třetí rovnice poslední soustavy dále získáme

$$\begin{aligned}b\sqrt{z} + (c - a)\sqrt{y} &= b\sqrt{x}, \\a\sqrt{z} - a\sqrt{y} &= (b - c)\sqrt{x}.\end{aligned}$$

Obě strany první rovnice předešlé soustavy násobíme číslem a , obě strany druhé rovnice pak násobíme číslem $-b$. Sečteme-li obě takto upravené rovnice, obdržíme

$$a(b + c - a)\sqrt{y} = b(c + a - b)\sqrt{x},$$

obdobným způsobem dostaneme rovněž

$$a(b + c - a)\sqrt{z} = c(a + b - c)\sqrt{x}.$$

Jestliže pro kladná čísla a, b, c platí vztah $b + c - a = 0$, pak z předešlých dvou rovnic plyne, že také $a + b - c = 0$, $c + a - b = 0$. Potom však $a = b = c = 0$, což není možné. Je tudíž $b + c - a \neq 0$. Z poslední dvojice rovnic vyjádříme $\sqrt{y}$ a $\sqrt{z}$ pomocí $\sqrt{x}$ následujícím způsobem:

$$\begin{aligned}\sqrt{y} &= \frac{b(c + a - b)}{a(b + c - a)}\sqrt{x}, \\ \sqrt{z} &= \frac{c(a + b - c)}{a(b + c - a)}\sqrt{x}.\end{aligned}$$

Odtud snadno vidíme, že výrazy $b + c - a$, $c + a - b$, $a + b - c$ jsou současně všechny kladné nebo všechny záporné. Po dosazení $\sqrt{y}$ a $\sqrt{z}$ do původní soustavy rovnic získáme (po úpravách) řešení (x, y, z) , kde

$$\begin{aligned}x &= \frac{a^2(b + c - a)}{(c + a - b)(a + b - c)}, \\ y &= \frac{b^2(c + a - b)}{(b + c - a)(a + b - c)}, \\ z &= \frac{c^2(a + b - c)}{(c + a - b)(b + c - a)}.\end{aligned}$$

Vzhledem k tomu, že jsme k řešení soustavy rovnic dospěli výhradně ekvivalentními úpravami, není třeba zkoušku provádět.

Soustava má přitom výše uvedené řešení v oboru kladných čísel, právě když současně platí $b + c - a > 0$, $c + a - b > 0$, $a + b - c > 0$, tj. právě když kladná čísla a, b, c jsou délkami stran trojúhelníku.

A – I – 4

Nechť $A_s B_s C_s$, kde $s \in \{1, 2, \dots, 1999\}$, jsou trojúhelníky vyhovující podmínkám úlohy. Každý z daných trojúhelníků $A_s B_s C_s$ je průnikem vždy tří polorovin $A_s B_s C_s$, $B_s C_s A_s$ a $C_s A_s B_s$, proto je (neprázdná) množina M průnikem $3 \cdot 1999 = 5997$ takových polorovin. Vzhledem k tomu, že poloroviny $A_s B_s C_s$, kde $s \in \{1, 2, \dots, 1999\}$, se navzájem liší jen posunutím, je jejich průnikem polorovina $A_i B_i C_i$, kde i je pevný index z množiny $\{1, 2, \dots, 1999\}$. Podobně průnikem všech polorovin $B_s C_s A_s$ je určitá polorovina $B_j C_j A_j$ a průnikem všech polorovin $C_s A_s B_s$ je určitá polorovina $C_k A_k B_k$, kde $j, k \in \{1, 2, \dots, 1999\}$.

Množina M je proto průnikem tří výše zmíněných polorovin $A_i B_i C_i$, $B_j C_j A_j$ a $C_k A_k B_k$, M je tedy trojúhelník ABC , kde A je průsečík přímk $A_i B_i$ a $C_k A_k$, B je průsečík přímk $A_i B_i$ a $B_j C_j$ a konečně C je průsečík přímk $B_j C_j$ a $C_k A_k$. Tento trojúhelník je podobný všem trojúhelníkům $A_s B_s C_s$, přičemž pro poměr podobnosti λ platí $0 < \lambda \leq 1$. (Případ $A = B = C$ lze dle textu úlohy vyloučit.)

Vzhledem k tomu, že obsah trojúhelníku ABC je λ^2 , stačí dokázat, že $\lambda \geq \frac{1}{3}$. Označme v výšku z vrcholu C_i na stranu $A_i B_i$ v trojúhelníku $A_i B_i C_i$. Protože přímka $A_i B_i$ je totožná s přímkou AB , je vzdálenost těžiště T_i trojúhelníku $A_i B_i C_i$ od přímky AB rovna $\frac{1}{3}v$. Podle zadání obsahuje množina M těžiště všech trojúhelníků $A_s B_s C_s$, musí tudíž obsahovat těžiště T_i trojúhelníku $A_i B_i C_i$.

Vzdálenost vrcholu C trojúhelníku ABC od jeho strany AB je tedy alespoň $\frac{1}{3}v$. Porovnáním velikostí výšek z vrcholů C_i a C v podobných trojúhelnících $A_i B_i C_i$ a ABC dostáváme již přímo žádanou nerovnost $\lambda \geq \frac{1}{3}$, tj. $\lambda^2 \geq \frac{1}{9}$, což jsme chtěli dokázat.

A – I – 5

Označme

$$S(n) = f(1) + f(2) + \dots + f(n).$$

Ze zadání plyne $S(1) = 1$. Protože $f(n) \geq 1$ pro všechna přirozená čísla n , je $S: \mathbb{N} \rightarrow \mathbb{N}$ rostoucí funkce. Je-li n přirozené číslo tvaru $n = 2^k$, kde k je přirozené, určíme součet $S(n)$ následujícím způsobem: Počet lichých čísel, která nejsou větší než n , je 2^{k-1} . Každé liché číslo se na součtu $S(n)$ podílí hodnotou 1. Počet sudých čísel, která nejsou větší než n , je rovněž 2^{k-1} , přitom každé sudé číslo se na součtu $S(n)$ podílí hodnotou minimálně 1. Je-li navíc toto číslo dělitelné čtyřmi, podílí se na součtu

další 1. Je-li dále číslo dělitelné osmi, podílí se další 1, atd. (Hodnotu $S(n)$ tak tvoříme načítáním hodnot 1 „po vrstvách“). Celkem je tedy

$$\begin{aligned} S(2^k) &= 2^{k-1} + 2^{k-1} + 2^{k-2} + \dots + 2 + 1 = 2^{k-1} + \frac{2^k - 1}{2 - 1} = \\ &= 2^k + 2^{k-1} - 1 = 3 \cdot 2^{k-1} - 1. \end{aligned}$$

Nechť p je přirozené číslo, které lze zapsat ve tvaru $p = 2^m s$, kde m je celé nezáporné číslo a s liché přirozené číslo. Nechť k je přirozené číslo takové, že $p < 2^k$ (tedy $m < k$), a nechť l je liché přirozené číslo. Pak

$$f(2^k l + p) = f(2^k l + 2^m s) = f(2^m(2^{k-m} l + s)).$$

Číslo $2^{k-m} l + s$ je liché, proto $f(2^m(2^{k-m} l + s)) = f(2^m s) = f(p)$. Celkem tedy dostáváme $f(2^k l + p) = f(p)$.

Jsou-li k, m nezáporná celá čísla, $k > m$, a l liché číslo, platí podle předcházejícího odstavce

$$\begin{aligned} S(2^k l + 2^m) &= f(1) + f(2) + \dots + f(2^k l) + f(2^k l + 1) + f(2^k l + 2) + \\ &\quad + \dots + f(2^k l + 2^m) = \\ &= f(1) + f(2) + \dots + f(2^k l) + f(1) + f(2) + \dots + f(2^m) = \\ &= S(2^k l) + S(2^m). \end{aligned}$$

A odtud již matematickou indukcí lehce dokážeme, že jsou-li $k_1 > k_2 > \dots > k_i$ nezáporná celá čísla, pak platí

$$S(2^{k_1} + 2^{k_2} + \dots + 2^{k_i}) = S(2^{k_1}) + S(2^{k_2}) + \dots + S(2^{k_i}).$$

Největší nezáporné celé číslo k_1 takové, že $3 \cdot 2^{k_1-1} - 1 = S(2^{k_1}) \leq 123\,456$, je $k_1 = 16$. Přitom $S(2^{16}) = 98\,303$.

Největší nezáporné celé číslo k_2 takové, že $3 \cdot 2^{k_2-1} - 1 = S(2^{k_2}) \leq 123\,456 - 98\,303 = 25\,153$, je $k_2 = 14$. Přitom $S(2^{14}) = 24\,575$.

Největší nezáporné celé číslo k_3 takové, že $3 \cdot 2^{k_3-1} - 1 = S(2^{k_3}) \leq 25\,153 - 24\,575 = 578$, je $k_3 = 8$. Přitom $S(2^8) = 383$.

Největší nezáporné celé číslo k_4 takové, že $3 \cdot 2^{k_4-1} - 1 = S(2^{k_4}) \leq 578 - 383 = 195$, je $k_4 = 7$. Přitom $S(2^7) = 191$.

Největší nezáporné celé číslo k_5 takové, že $3 \cdot 2^{k_5-1} - 1 = S(2^{k_5}) \leq 195 - 191 = 4$, je $k_5 = 1$. Přitom $S(2^1) = 2$.

Největší nezáporné celé číslo k_6 takové, že $S(2^{k_6}) \leq 4 - 2 = 2$, je $k_6 = 0$. Přitom $S(2^0) = 1$.

Tedy

$$\begin{aligned} S(82\,307) &= S(2^{16} + 2^{14} + 2^8 + 2^7 + 2 + 1) = \\ &= S(2^{16}) + S(2^{14}) + S(2^8) + S(2^7) + S(2) + S(1) = \\ &= 123\,455 \leq 123\,456. \end{aligned}$$

Přitom $S(82\,308) = S(82\,307) + f(82\,308) = 123\,455 + 2 = 123\,457 > 123\,456$.

Největší přirozené číslo n , pro něž platí $S(n) \leq 123\,456$, je $n = 82\,307$.

Jiné řešení. Na základě úvahy o načítání hodnot „po vrstvách“ jako v předešlém řešení zjistíme, že

$$S(n) = n + \left\lfloor \frac{n}{4} \right\rfloor + \left\lfloor \frac{n}{8} \right\rfloor + \left\lfloor \frac{n}{16} \right\rfloor + \dots$$

Přitom $\lfloor r \rfloor$ znamená celou část reálného čísla r , což je největší celé číslo, které není větší než r .

Protože pro každé reálné číslo r platí $\lfloor r \rfloor \leq r$, platí též

$$S(n) \leq n + \frac{n}{4} + \frac{n}{8} + \dots = \frac{n}{2} + \frac{n}{2} \left(1 + \frac{1}{2} + \frac{1}{4} + \dots \right) = \frac{n}{2} + n = \frac{3n}{2}.$$

Největší přirozené číslo n , pro něž platí, že $\frac{3}{2}n \leq 123\,456$, je $n = 82\,304$. Přitom

$$\begin{aligned} S(82\,304) &= 82\,304 + \left\lfloor \frac{82\,304}{4} \right\rfloor + \left\lfloor \frac{82\,304}{8} \right\rfloor + \left\lfloor \frac{82\,304}{16} \right\rfloor + \dots + \\ &\quad + \left\lfloor \frac{82\,304}{65\,536} \right\rfloor + \left\lfloor \frac{82\,304}{131\,072} \right\rfloor + \dots = \\ &= 82\,304 + 20\,576 + 10\,288 + 5\,144 + 2\,572 + 1\,286 + \\ &\quad + 643 + 321 + 160 + 80 + 40 + 20 + 10 + 5 + \\ &\quad + 2 + 1 + 0 + 0 + \dots = \\ &= 123\,452. \end{aligned}$$

Dále

$$S(82\,305) = S(82\,304) + f(82\,305) = 123\,452 + 1 = 123\,453,$$

$$S(82\,306) = S(82\,305) + f(82\,306) = 123\,453 + 1 = 123\,454,$$

$$S(82\,307) = S(82\,306) + f(82\,307) = 123\,454 + 1 = 123\,455,$$

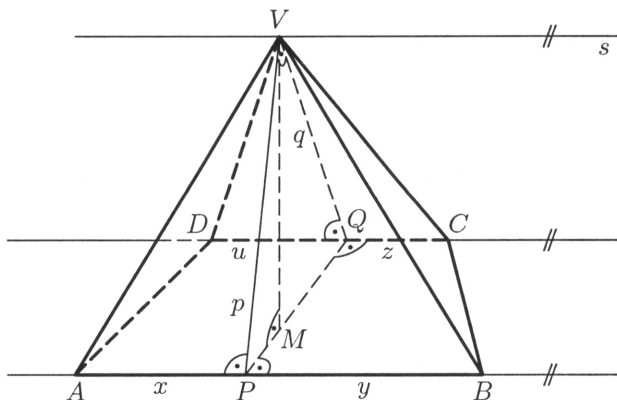
$$S(82\,308) = S(82\,307) + f(82\,308) = 123\,455 + 2 = 123\,457.$$

Největší přirozené číslo n , pro něž $S(n) \leq 123\,456$, je tedy $n = 82\,307$.

A – I – 6

Přímka AB je průsečnicí roviny ABV s rovinou podstavy $ABCD$ čtyřbokého jehlanu $ABCDV$, podobně přímka CD je průsečnicí roviny CDV s rovinou podstavy $ABCD$ uvažovaného jehlanu. Vzhledem k tomu, že obě průsečnice jsou dle zadání rovnoběžné, je rovněž průsečnice s rovin ABV a CDV s nimi rovnoběžná (obr. 43). Rovina kolmá k přímce s , procházející vrcholem V daného jehlanu, protíná přímky AB , CD po řadě v bodech P , Q , které jsou patami výšek z vrcholu V po řadě na strany AB , CD v trojúhelnících ABV , CDV . Roviny ABV a CDV jsou podle zadání vzájemně kolmé, trojúhelník PQV má proto pravý úhel u vrcholu V . Pata M výšky z vrcholu V na přeponu PQ je přitom totožná s patou tělesové výšky z vrcholu V jehlanu $ABCDV$. Pro polohu bodů P a Q na přímce AB , resp. CD , je třeba dále rozlišit tři případy:

- (i) Oba body P a Q leží na odpovídajících hranách AB , CD .
- (ii) Jeden z bodů P , Q leží na odpovídající hraně, druhý na prodloužení odpovídající hrany.
- (iii) Žádný z bodů P , Q neleží na odpovídající hraně.



Obr. 43

Dokážeme dále nerovnost z textu úlohy pro případ (i). Zavedme označení ve shodě s obr. 43, tj.

$$|AP| = x, |BP| = y, |CQ| = z, |DQ| = u, |VP| = p, |VQ| = q.$$

Využitím Pythagorovy věty v pravoúhlých trojúhelnících APV , BPV , CQV , DQV a PQV dostáváme postupně vztahy:

$$|AV|^2 = x^2 + p^2, \quad |BV|^2 = y^2 + p^2, \quad |CV|^2 = z^2 + q^2, \\ |DV|^2 = u^2 + q^2, \quad |PQ|^2 = p^2 + q^2.$$

Pro obsahy trojúhelníků ABV , CDV a PQV platí vzorce

$$2S_{ABV} = (x + y)p, \quad 2S_{CDV} = (z + u)q, \quad 2S_{PQV} = pq.$$

Dosadíme-li nyní za $|AV|^2$, $|BV|^2$, $|CV|^2$, $|DV|^2$, $|PQ|^2$ a $2S_{ABV}$, $2S_{CDV}$, $2S_{PQV}$ do nerovnosti v textu úlohy, dostáváme po snadné úpravě

$$x^2 + y^2 + z^2 + u^2 + p^2 + q^2 \geq xp + yp + zq + uq + pq.$$

Nyní dokážeme, že předešlá nerovnost platí pro libovolná nezáporná reálná čísla x, y, z, u a libovolná kladná čísla p, q . Vynásobením rozdílu levé a pravé strany této nerovnosti číslem 4 dostáváme po úpravě

$$(4x^2 - 4xp + p^2) + (4y^2 - 4yp + p^2) + (4z^2 - 4zq + q^2) + \\ + (4u^2 - 4uq + q^2) + 2(p^2 - 2pq + q^2) = \\ = (2x - p)^2 + (2y - p)^2 + (2z - q)^2 + (2u - q)^2 + 2(p - q)^2 \geq 0.$$

Vzhledem k tomu, že všechny provedené úpravy byly ekvivalentní, platí též nerovnost uvedená v textu úlohy, což jsme měli dokázat.

Podobně lze postupovat i v případech (ii) a (iii). Odlišné je zde pouze vyjádření hodnot $2S_{ABV}$ a $2S_{CDV}$.

Rovnost může nastat pouze v případě (i), ve zbylých dvou případech je vyloučena. V případě (i) přitom rovnost nastává, právě když platí

$$2x = 2y = 2z = 2u = p = q,$$

tj. právě když podstavou daného čtyřbokého jehlanu $ABCDV$ je obdélník $ABCD$, pata M výšky VM uvažovaného jehlanu je průsečíkem úhlopříček AC a BD v obdélníku $ABCD$ a současně platí

$$|AB| : |BC| : |VM| = 2 : 2\sqrt{2} : \sqrt{2}.$$

První rovnice je splněna, právě když platí $x = y + p$ nebo $x = y - p$. Po dosazení do druhé rovnice dané soustavy dostaneme po úpravě v prvním případě kvadratickou rovnici

$$3py^2 + 3p^2y + p^3 - 16 = 0,$$

v druhém případě pak kvadratickou rovnici

$$3py^2 - 3p^2y + p^3 + 16 = 0$$

o neznámé y . Daná soustava rovnic bude mít právě jedno řešení v oboru reálných čísel, právě když jedna ze dvou předešlých kvadratických rovnic bude mít jediný (dvojnásobný) kořen a druhá z nich nebude mít žádný reálný kořen nebo bude mít stejný dvojnásobný kořen jako rovnice první (můžeme předpokládat, že $p \neq 0$, protože pro $p = 0$ daná soustava zřejmě nemá řešení). První kvadratická rovnice má diskriminant $D_1 = 3p(64 - p^3)$, druhá má diskriminant $D_2 = -3p(64 + p^3)$. Hledáme tedy ta $p \neq 0$, pro něž je jedno z čísel D_1, D_2 rovno nule a druhé záporné (případ $D_1 = D_2 = 0$ pro $p \neq 0$ totiž nenastane).

Je-li $D_1 = 0$, je $p = 4$ a $D_2 < 0$. Pokud $D_2 = 0$, je $p = -4$ a $D_1 < 0$. Hodnoty $p = 4$ a $p = -4$ jsou tedy jediné, které mají požadovanou vlastnost.

Daná soustava rovnic má přitom pro obě uvedené hodnoty parametru p jediné reálné řešení $(x, y) = (2, -2)$.

Jiné řešení. Z první rovnice máme $|x - y| = |p|$, z druhé rovnice však vidíme, že $x^3 > y^3$, což je ekvivalentní s nerovností $x > y$ (je $x^3 - y^3 = (x - y)(x^2 + xy + y^2)$ a $x^2 + xy + y^2 > 0$ pro libovolná reálná x, y s výjimkou případu $x = y = 0$). Je tedy $x = y + |p|$, $|p| > 0$. Po dosazení do druhé rovnice soustavy (pro jednoduchost píšme q místo $|p|$) dostaneme pro y kvadratickou rovnici

$$3qy^2 + 3q^2y + q^3 - 16 = 0$$

s diskriminantem $D(q) = 3q(64 - q^3) = 3q(4 - q)(16 + 4q + q^2)$. Má-li daná soustava v oboru reálných čísel jediné řešení, je nutně diskriminant $D(q)$ předešlé rovnice roven 0, tj. musí platit $(4 - q)(16 + 4q + q^2) = 0$ (víme, že $q = |p| > 0$). Protože pro libovolné reálné q je $16 + 4q + q^2 > 0$,

musí být $q = |p| = 4$, tj. $p = 4$ nebo $p = -4$. Zároveň hned dostáváme, že $y = -\frac{1}{2}q = -2$ a $x = y + 4 = 2$.

Daná soustava rovnic má právě jedno reálné řešení, právě když $p = 4$ nebo $p = -4$, a to $(x, y) = (2, -2)$.

Jiné řešení. Z první rovnice máme $x = y + q$, kde $q = p$ nebo $q = -p$. Po dosazení do druhé rovnice soustavy dostaneme pro y kvadratickou rovnici

$$3qy^2 + 3q^2y + q^3 - 16 = 0$$

s diskriminantem $D(q) = 3q(64 - q^3) = 3q(4 - q)(16 + 4q + q^2)$. Má-li daná soustava v oboru reálných čísel jediné řešení, je nutně diskriminant $D(q)$ předešlé rovnice roven 0, tj. musí platit $(4 - q)(16 + 4q + q^2) = 0$. Protože pro $p = 0$ nemá soustava řešení, je $q \neq 0$, navíc pro libovolné reálné q je $16 + 4q + q^2 > 0$, takže musí být $q = 4$, tj. $p = 4$ nebo $p = -4$.

Vraťme se znovu na počátek našeho řešení. Zkouškou se snadno přesvědčíme, že pro $q = 4$ má daná soustava právě jedno reálné řešení $(x, y) = (2, -2)$. Pro $q = -4$ (po dosazení za $x = y - 4$) však nemá rovnice $(y - 4)^3 - y^3 = 16$ žádný reálný kořen.

Daná soustava rovnic má právě jedno reálné řešení, právě když $p = 4$ nebo $p = -4$, a to $(x, y) = (2, -2)$.

A – S – 2

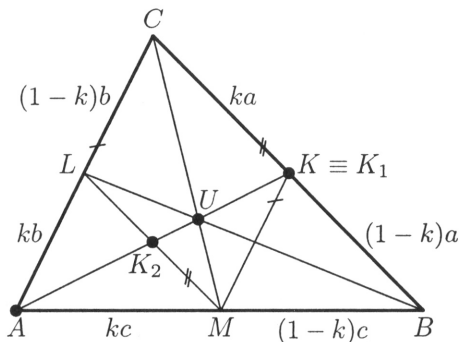
Z rovnosti obsahů trojúhelníků AMU a KCU plyne rovnost obsahů trojúhelníků AMC a AKC . Body K, M mají tedy stejnou vzdálenost od přímky AC . Odtud plyne, že $CA \parallel MK$ a čtyřúhelník $CAMK$ je tedy lichoběžník. Podobně dokážeme, že čtyřúhelník $BCLM$ je rovněž lichoběžník, kde $BC \parallel LM$. Trojúhelníky AML a ABC , resp. BKM a BCA jsou tedy stejnohlé a platí (obráz. 44)

$$|AL| = kb, \quad |AM| = kc, \quad |BM| = (1 - k)c, \quad |BK| = (1 - k)a$$

a dále

$$|CK| = ka, \quad |CL| = (1 - k)b, \quad \text{kde } k \in (0; 1).$$

Užitím Cèvyovy věty pro trojici úseček AK, BL a CM , které se dle textu



Obr. 44

úlohy protínají v bodě U , dostáváme

$$\frac{kc}{(1-k)c} \cdot \frac{(1-k)a}{ka} \cdot \frac{(1-k)b}{kb} = 1.$$

Odtud plyne $(1-k)/k = 1$, neboli $k = \frac{1}{2}$. Tyto úsečky jsou tedy těžnice a jejich průsečík U je těžištěm daného trojúhelníku. Ze shodnosti úseček AM a BM již plyne rovnost obsahů trojúhelníků AMU a BMU , tedy rovnost $P = Q$, což jsme měli dokázat.

Jiné řešení (bez užití Cèvovy věty). Stejně jako v prvním řešení ukážeme, že úsečky BC a LM jsou rovnoběžné, takže si navzájem odpovídají v jisté stejnolehlosti se středem U a zároveň i v jisté stejnolehlosti se středem A . Označme K_1 , K_2 po řadě středy obou uvažovaných úseček. Vzhledem k tomu, že body K_1 , K_2 si odpovídají v obou zmíněných stejnolehlostech, leží body A a U (středy obou stejnolehlostí) na přímce K_1K_2 . Odtud plyne, že střed K_1 strany BC leží na přímce AU , je tedy totožný s bodem K z textu úlohy. Úsečka AK je tudíž těžnicí trojúhelníku ABC . Podobně dokážeme, že i úsečka BL je těžnicí daného trojúhelníku. Bod U je tedy jeho těžištěm. Závěr je pak stejný jako v prvním řešení.

A – S – 3

Ukážeme, že hledaným k je číslo 1 001. Rozdělme všechna čísla z množiny $\{1, 2, 3, \dots, 2\,000\}$ do 1 000 dvojic

$$\{1, 666\}, \{2, 665\}, \{3, 664\}, \dots, \{333, 334\}, \\ \{667, 1334\}, \{668, 1335\}, \{669, 1336\}, \dots, \{1\,333, 2\,000\}.$$

(Číslo 667 z textu úlohy je rovno součtu čísel každé dvojice v prvním řádku a rozdílu čísel každé dvojice v druhém řádku. Všimněme si, že skutečně každé z čísel $1, 2, 3, \dots, 2\,000$ je zastoupeno právě v jedné dvojici.)

Číslo 1 001 má požadovanou vlastnost, protože pokud vybereme libovolně 1 001 čísel z množiny $\{1, 2, 3, \dots, 2\,000\}$, budou mezi nimi obě čísla aspoň jedné z uvedených dvojic (máme vybráno 1 001 čísel, ale jen 1 000 dvojic). Součet nebo rozdíl čísel v nalezené dvojici je však 667.

Nyní ukážeme, že žádné číslo $k \leq 1\,000$ požadovanou vlastnost nemá. Stačí to zřejmě ukázat pro $k = 1\,000$: vybereme-li 1 000 sudých čísel $2, 4, 6, \dots, 2\,000$, je součet i rozdíl libovolných dvou vybraných čísel sudý, takže se nemůže rovnat lichému číslu 667.

A – II – 1

Označme $Q(x) = x^2 + 4x - 7$, potom $0 = P(Q(1)) = P(-2)$. Odtud plyne, že $P(x) = a(x+2)(x-p)$, kde a a p jsou reálná čísla, $a \neq 0$. Je tedy

$$\begin{aligned} P(Q(x)) &= a(x^2 + 4x - 7 + 2)(x^2 + 4x - 7 - p) = \\ &= a(x-1)(x+5)(x^2 + 4x - 7 - p). \end{aligned}$$

To znamená, že kořeny dané rovnice jsou kromě čísel 1 a -5 ještě kořeny kvadratické rovnice

$$x^2 + 4x - 7 - p = 0. \tag{1}$$

Protože aspoň jeden z kořenů dané rovnice má být dvojnásobný, je buď aspoň jedno z čísel 1 a -5 kořenem rovnice (1), nebo má tato rovnice sama dvojnásobný kořen. Přitom z tvaru rovnice (1) plyne, že součet jejích kořenů je -4 (číslo opačné ke koeficientu u lineárního členu), takže tato rovnice má kořen 1, právě když má kořen -5 .

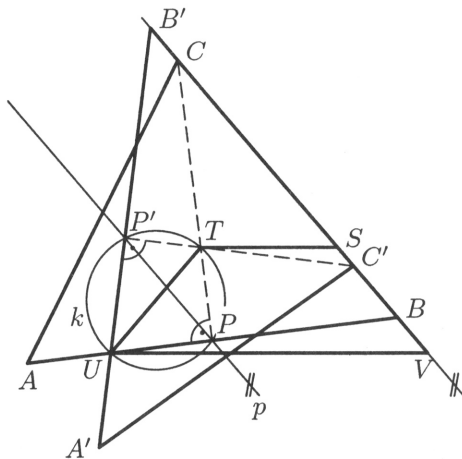
Jsou tedy dvě možnosti:

a) Rovnice (1) má dva kořeny 1 a -5 (takže je $p = -2$) a rovnice $P(Q(x)) = a(x-1)^2(x+5)^2 = 0$ má dva dvojnásobné kořeny 1 a -5 .

b) Rovnice (1) má sama dvojnásobný kořen. Protože součet jejích kořenů je -4 , je dvojnásobným kořenem číslo $(-4) : 2 = -2$. (V tomto případě je $p = -11$ a $P(Q(x)) = a(x-1)(x+5)(x+2)^2 = 0$.)

Závěr: Úloha má dvě řešení: daná rovnice má buď dva dvojnásobné kořeny 1 a -5 , nebo má dva jednoduché kořeny 1 a -5 a dvojnásobný kořen -2 .

Označme P střed základny AB hledaného rovnoramenného trojúhelníku ABC . Protože vrchol U daného lichoběžníku $UVST$ leží na přímce AB , je buď $U = P$, anebo tvoří body T, U, P vrcholy pravoúhlého trojúhelníku (obr. 45). V obou případech bod P leží na Thaletově kružnici k sestrojené



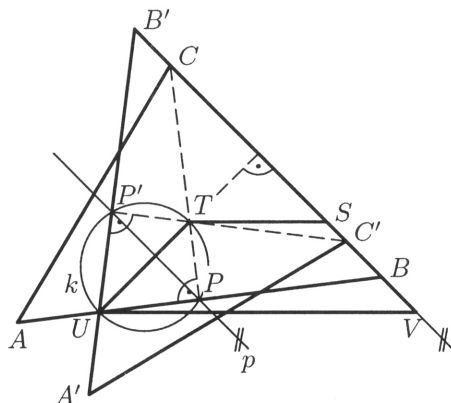
Obr. 45

nad průměrem TU . Označme d vzdálenost vrcholu T daného lichoběžníku od přímky VS . Vzhledem k tomu, že T je těžištěm trojúhelníku ABC , má jeho výška z vrcholu A velikost $3d$, tudíž bod P leží na přímce p , která je s přímkou VS rovnoběžná, má od ní vzdálenost $\frac{3}{2}d$ a leží v polovině VST . Odtud již plyne *konstrukce* trojúhelníku ABC :

1. sestrojíme kružnici k s průměrem TU ;
2. sestrojíme v polovině VST přímku $p \parallel VS$ ve vzdálenosti $\frac{3}{2}d$ od VS ;
3. sestrojíme bod $P \in k \cap p$;
4. sestrojíme přímku $PB \perp TP$, $B \in VS$;
5. sestrojíme vrcholy A ($A \in PB$, $A \neq B$, $|AP| = |PB|$) a C ($C \in PT \cap VS$).

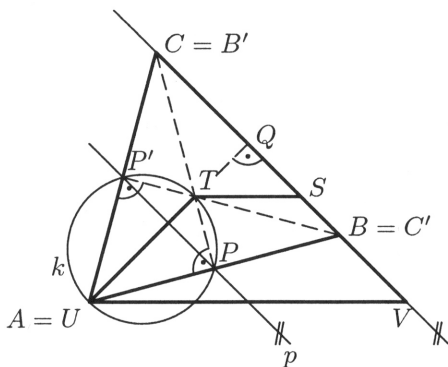
Diskuse: Protože dle předpokladu je $ST \parallel UV$ a $\frac{3}{2}|ST| < |UV|$, protne přímka p stranu TU daného lichoběžníku ve vnitřním bodě, bude tedy sečnou kružnice k a protne ji ve dvou různých bodech P a P' (obr. 45). Pro každý z nich dostáváme jedno řešení, trojúhelníky ABC a $A'B'C'$. Z konstrukce je dále zřejmé, že pokud bude $TU \perp SV$, budou

oba body P, P' souměrně sdružené podle osy TU , takže dostaneme dva shodné (souměrně sdružené) trojúhelníky ABC a $A'B'C'$ (obr. 46). Obě



Obr. 46

souměrně sdružená řešení splynou v jedno v případě, kdy vyjde $A = U$. Přitom bude těžiště T trojúhelníku ABC zároveň průsečíkem jeho výšek, takže výsledný trojúhelník ABC bude rovnostranný (obr. 47). To nastane, právě když obě ramena daného lichoběžníku jsou navzájem kolmá



Obr. 47

a navíc platí $3|ST| = |UV|$, jak plyne z podobnosti trojúhelníků UVQ a TSQ . V tomto jediném případě má úloha jedno řešení. Ve všech ostatních případech má úloha dvě řešení (která jsou pro $TU \perp SV$ shodná).

Umocněním obou stran dané nerovnosti na třetí dostaneme ekvivalentní nerovnost

$$\frac{a}{b} + 3\sqrt[3]{\frac{a}{b}} + 3\sqrt[3]{\frac{b}{a}} + \frac{b}{a} \leq 4 + 2\frac{a}{b} + 2\frac{b}{a}$$

neboli

$$\frac{a}{b} + \frac{b}{a} + 4 \geq 3 \left(\sqrt[3]{\frac{a}{b}} + \sqrt[3]{\frac{b}{a}} \right). \quad (1)$$

V předcházející nerovnosti položíme $x = \sqrt[3]{a/b}$ ($x > 0$). Po vynásobení obou stran nerovnosti (kladným) číslem x^3 a snadné úpravě obdržíme ekvivalentní nerovnost

$$x^6 - 3x^4 + 4x^3 - 3x^2 + 1 \geq 0.$$

Nejdříve zjistíme, zda rovnice $x^6 - 3x^4 + 4x^3 - 3x^2 + 1 = 0$ nemá celočíselný kořen. Takový kořen musí dělit absolutní člen, takže jsou jen dvě možnosti, 1 a -1 . Snadno ověříme, že uvedená rovnice má kořen $x = 1$, a po dělení dvojčlenem $(x - 1)$ zjistíme, že jde dokonce o kořen dvojnásobný a že platí rozklad

$$x^6 - 3x^4 + 4x^3 - 3x^2 + 1 = (x - 1)^2(x^4 + 2x^3 + 2x + 1).$$

Pro každé $x > 0$ je $x^4 + 2x^3 + 2x + 1 > 0$. Platí tedy

$$x^6 - 3x^4 + 4x^3 - 3x^2 + 1 = (x - 1)^2(x^4 + 2x^3 + 2x + 1) \geq 0,$$

což jsme měli dokázat. Rovnost v předchozí nerovnosti přitom nastává, právě když $x = 1$, tj. právě když platí $a = b$.

Druhé řešení. Užitím nerovnosti mezi aritmetickým a geometrickým průměrem pro trojici kladných čísel a/b , 1, 1 dostaneme

$$\frac{a}{b} + 1 + 1 \geq 3\sqrt[3]{\frac{a}{b}},$$

a obdobně

$$\frac{b}{a} + 1 + 1 \geq 3\sqrt[3]{\frac{b}{a}}.$$

V obou předchozích nerovnostech nastává rovnost, právě když $a = b$. Jejich součtem pak výjde

$$\frac{a}{b} + \frac{b}{a} + 4 \geq 3\sqrt[3]{\frac{a}{b}} + 3\sqrt[3]{\frac{b}{a}},$$

což je nerovnost (1).

Třetí řešení. Podle nerovnosti mezi mocninnými průměry stupně $\frac{1}{3}$ a $\frac{1}{2}$ dostáváme pro kladná čísla a/b , b/a nerovnost¹

$$\left(\frac{\sqrt[3]{a/b} + \sqrt[3]{b/a}}{2} \right)^3 \leq \left(\frac{\sqrt{a/b} + \sqrt{b/a}}{2} \right)^2,$$

v níž nastává rovnost, právě když $a/b = b/a$, tj. právě když $a = b$.

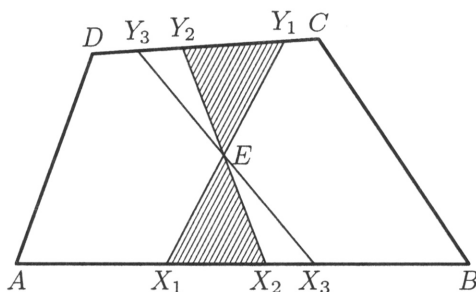
Protože

$$\left(\sqrt{\frac{a}{b}} + \sqrt{\frac{b}{a}} \right)^2 = (a+b) \left(\frac{1}{a} + \frac{1}{b} \right),$$

dostáváme odtud po jednoduché úpravě dokazovanou nerovnost, v níž nastává rovnost, právě když $a = b$.

A – II – 4

Nechť E je vnitřním bodem takového konvexního čtyřúhelníku $ABCD$, který vyhovuje podmínkám úlohy. Uvažujme přímky X_1Y_1 , X_2Y_2 a X_3Y_3 , které procházejí bodem E a protínají po řadě strany AB a CD v bodech X_1 a Y_1 , X_2 a Y_2 , X_3 a Y_3 (obr. 48). Jestliže všechny tři uvažované



Obr. 48

¹ Viz např. J. Herman, R. Kučera, J. Šimša: Metody řešení matematických úloh I, str. 174.

přímky dělí čtyřúhelník $ABCD$ na dvě části o stejném obsahu, rovnají se i obsahy trojúhelníků EX_1X_2 a EY_1Y_2 , resp. EX_2X_3 a EY_2Y_3 . Protože tyto trojúhelníky mají vždy shodné vnitřní úhly při vrcholu E , plyne z rovnosti jejich obsahů rovnost

$$|EX_1| \cdot |EX_2| = |EY_1| \cdot |EY_2|,$$

resp.

$$|EX_2| \cdot |EX_3| = |EY_2| \cdot |EY_3|.$$

Z obou předešlých rovností dostáváme

$$\frac{|EX_1|}{|EX_3|} = \frac{|EY_1|}{|EY_3|}.$$

Trojúhelníky EX_1X_3 a EY_1Y_3 jsou tedy podobné (podle věty sus) a mají též obsah. Jsou proto středově souměrné podle středu E a platí tudíž $X_1X_3 \parallel Y_1Y_3$. Čtyřúhelník $ABCD$ má tedy nutně rovnoběžné strany AB a CD .

Naopak každý (konvexní) čtyřúhelník $ABCD$, v němž platí $AB \parallel CD$, vyhovuje podmínkám úlohy. Za bod E pak zvolíme střed úsečky spojující středy rovnoběžných stran AB a CD ; požadovaná vlastnost takového bodu je zřejmá.

Závěr: Podmínkám úlohy vyhovují právě všechny konvexní čtyřúhelníky $ABCD$, v nichž $AB \parallel CD$.

A – III – 1

Označme $a_n = 4 \cdot 3^{2^n} + 3 \cdot 4^{2^n}$. Ukážeme nejprve, že pro každé přirozené číslo n je rozdíl $a_{n+2} - a_n$ dělitelný třináctí. Po úpravách obdržíme rovnost

$$a_{n+2} - a_n = 4 \cdot (81^{2^n} - 3^{2^n}) + 3 \cdot (256^{2^n} - 4^{2^n}). \quad (1)$$

Položme ve známém vzorci

$$A^p - B^p = (A - B)(A^{p-1} + A^{p-2}B + \dots + B^{p-1}),$$

který platí pro každé přirozené p a pro libovolná dvě reálná čísla A a B , nejprve $p = 2^n$, $A = 81$, $B = 3$ a poté $p = 2^{n-1}$, $A = 256^2$, $B = 4^2$. Protože je $81 - 3 = 78 = 13 \cdot 6$ a také $256^2 - 4^2 = (256 - 4)(256 + 4) = = 252 \cdot 260 = 13 \cdot 20 \cdot 252$, jsou oba sčítanci na pravé straně rovnosti (1)

číslo dělitelná 13. Je proto také rozdíl $a_{n+2} - a_n$ dělitelný 13. Číslo a_1 není dělitelné 13, neboť $a_1 = 84 = 13 \cdot 6 + 6$, kdežto číslo a_2 číslem 13 dělitelné je ($a_2 = 1092 = 13 \cdot 84$). Užitím principu matematické indukce již snadno zjistíme, že a_n je dělitelné 13, právě když n je sudé. Tím je důkaz hotov.

Jiné řešení. Sestavme tabulku zbytků při dělení čísla $a_n = 4 \cdot 3^{2^n} + 3 \cdot 4^{2^n}$ třinácti.

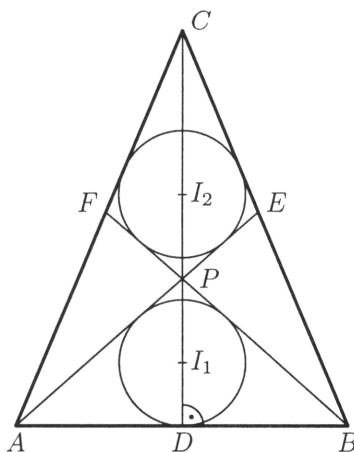
n	1	2	3	4	5	...
3^{2^n}	9	3	9	3	9	...
4^{2^n}	3	9	3	9	3	...
$4 \cdot 3^{2^n}$	10	12	10	12	10	...
$3 \cdot 4^{2^n}$	9	1	9	1	9	...
a_n	6	0	6	0	6	...

Zbytky obou čísel tvaru N^{2^n} určujeme rekurentně pomocí rovností $N^{2^{n+1}} = N^{2^n} \cdot N^{2^n} = (N^{2^n})^2$. Protože $3^2 = 9$ a $9^2 = 81 \equiv 3 \pmod{13}$, vidíme, že v druhém i třetím řádku tabulky se pravidelně střídá trojka s devítkou, zbytky čísla a_n při dělení třinácti se tedy (vzhledem k číslu n) rovněž opakují s periodou 2. Číslo a_n je tedy dělitelné třinácti, právě když n je sudé.

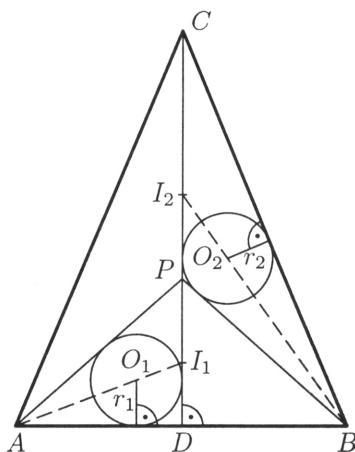
A – III – 2

Protože přímka CD je osou souměrnosti dvou vrcholových úhlů APB a EPF , leží střed I_1 kružnice vepsané trojúhelníku ABP na úsečce DP a zároveň střed I_2 kružnice vepsané čtyřúhelníku $PECF$ leží na úsečce CP (obr. 49). Navíc platí $|I_1P| = |I_2P|$, neboť obě zmíněné kružnice jsou shodné. Střed O_1 , O_2 kružnic vepsaných trojúhelníkům ADP a BCP (obr. 50) pak leží po řadě na úsečkách AI_1 , BI_2 , neboť polopřímky AI_1 a BI_2 jsou osy odpovídajících úhlů DAP a CBP . Z rovnosti $|I_1P| = |I_2P|$ navíc plyne, že trojúhelníky API_1 a BPI_2 mají stejný obsah, protože se rovnají i příslušné výšky $|AD| = |BD|$.

Označme r_1 , r_2 poloměry kružnic vepsaných trojúhelníkům ADP a BCP . Vyjádříme-li pomocí nich oba zmíněné obsahy ($S(XYZ)$ značí



Obr. 49



Obr. 50

obsah trojúhelníku XYZ), dostaneme

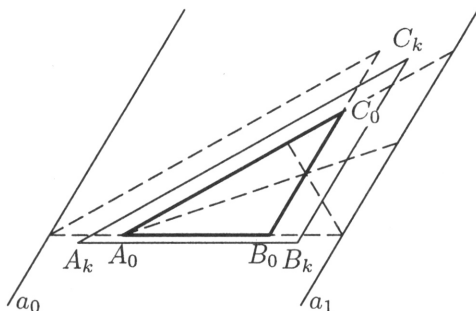
$$\begin{aligned}
 S(API_1) &= S(AO_1P) + S(O_1PI_1) = \\
 &= \frac{1}{2}|AP| \cdot r_1 + \frac{1}{2}|I_1P| \cdot r_1 = \frac{r_1}{2}(|AP| + |I_1P|), \\
 S(BPI_2) &= S(BO_2P) + S(O_2PI_2) = \\
 &= \frac{1}{2}|BP| \cdot r_2 + \frac{1}{2}|I_2P| \cdot r_2 = \frac{r_2}{2}(|BP| + |I_2P|).
 \end{aligned}$$

Vzhledem k tomu, že $S(API_1) = S(BPI_2)$, $|I_1P| = |I_2P|$ a $|AP| = |BP|$, plyne odtud $r_1 = r_2$, což jsme měli dokázat.

Jiné řešení. Vzhledem k souměrnosti trojúhelníku ABC podle osy CD stačí ukázat, že se shodují kružnice vepsané trojúhelníkům BDP a BPC .

Vnější společné tečny shodných kružnic vepsaných úhelníkům ABP a $PECF$ jsou rovnoběžné se střednou CD , tedy kolmé na přímkou AB . Uvažujme tu z nich, která protíná úsečky AD , PF a polopřímku opačnou CB . Tyto průsečíky označme po řadě D' , P' , C' (obr. 51). Ve stejno-
lehlosti, která zobrazí trojúhelník $BD'C'$ na trojúhelník BDC , odpovídají trojúhelníkům $BD'P'$ a $BP'C'$ trojúhelníky BDP a BPC . Protože kružnice vepsaná čtyřúhelníku $PECF$ je zároveň vepsána i trojúhelníku $BP'C'$ a kružnice vepsaná trojúhelníku ABP je zároveň vepsána trojúhelníku $BD'P'$ a obě uvedené kružnice jsou dle předpokladu shodné, jsou shodné i jejich obrazy ve zmíněné stejno-
lehlosti, tedy kružnice vepsané trojúhelníkům BDP a BPC .

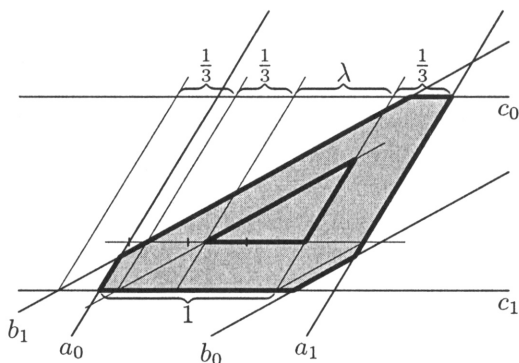
pásu omezeného dvěma rovnoběžkami $a_0 \parallel a_1 \parallel B_0C_0$ (obr. 52), jejichž vzdálenost od B_0C_0 je v_a a $\min(\frac{1}{3}, 1-\lambda) \cdot v_a$. Analogické tvrzení můžeme vyslovit i pro další dva směry C_0A_0 a A_0B_0 . Sjednocení všech daných trojúhelníků musí tedy ležet v průniku všech tří odpovídajících pásů.



Obr. 52

Rozlišíme nyní dva případy podle toho, čemu se rovná $\min(\frac{1}{3}, 1-\lambda)$.

1. Nechť $\frac{1}{3} \leq \lambda < \frac{2}{3}$. Průnikem odpovídajících tří pásů je šestiúhelník, který vznikne z trojúhelníku T určeného trojicí přímek (a_1, b_1, c_1) odstraněním tří trojúhelníků T_a, T_b, T_c určených trojicemi přímek (a_0, b_1, c_1) , (a_1, b_0, c_1) a (a_1, b_1, c_0) . V obr. 53 jsou vyznačeny některé poměrné vzdálenosti vzhledem k $|AB|$, s jejichž pomocí zjistíme, že trojúhelník T je



Obr. 53

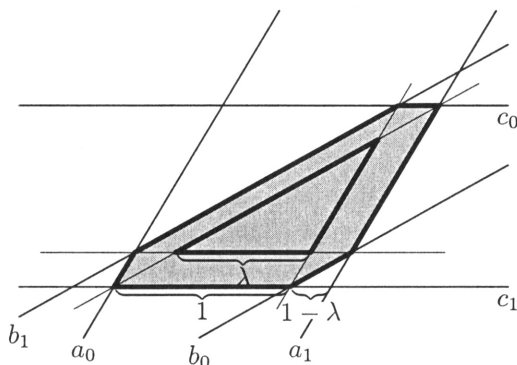
podobný trojúhelníku ABC s poměrem podobnosti $1 + \lambda$ a trojúhelníky T_a, T_b, T_c jsou podobné trojúhelníku ABC s poměrem podobnosti $\lambda - \frac{1}{3}$. Z vypočtených poměrů je zároveň zřejmé, že pro $\lambda = \frac{1}{3}$ se trojúhelníky

T_a, T_b, T_c stáhnou do jediného bodu, takže uvedený šestiúhelník se zredukuje na trojúhelník T .

Pro obsah $S(\lambda)$ vyznačeného útvaru tak pro $\lambda < \frac{2}{3}$ platí

$$\begin{aligned} S(\lambda) &= (1 + \lambda)^2 - 3\left(\lambda - \frac{1}{3}\right)^2 = \\ &= -2\lambda^2 + 4\lambda + \frac{2}{3} = -2(\lambda - 1)^2 + \frac{8}{3} < \frac{8}{3} - 2 \cdot \frac{1}{9} = \frac{22}{9}. \end{aligned}$$

2. Necht' $\frac{2}{3} \leq \lambda \leq 1$. Průnikem odpovídajících tří pásů je opět šestiúhelník (obr. 54), přičemž odpovídající trojúhelník T je podobný troj-



Obr. 54

úhelníku ABC s poměrem podobnosti $3 - 2\lambda$ a trojúhelníky T_a, T_b, T_c jsou podobné trojúhelníku ABC s poměrem podobnosti $1 - \lambda$ (v tomto případě se šestiúhelník zredukuje na trojúhelník T pro $\lambda = 1$).

Pro obsah $S(\lambda)$ v tomto případě platí

$$\begin{aligned} S(\lambda) &= (3 - 2\lambda)^2 - 3(1 - \lambda)^2 = \\ &= \lambda^2 - 6\lambda + 6 = (\lambda - 3)^2 - 3 \leq \frac{49}{9} - 3 = \frac{22}{9} \end{aligned}$$

s rovností pro $\lambda = \frac{2}{3}$.

Zjistili jsme, že sjednocení všech trojúhelníků $A_k B_k C_k$ ($k = 1, 2, \dots, 2000$) je pro $\lambda \neq \frac{2}{3}$ částí rovinného útvaru, jehož obsah je menší než $\frac{22}{9}$. Pro $\lambda = \frac{2}{3}$ je pak částí šestiúhelníku s obsahem $\frac{22}{9}$. Strana tohoto šestiúhelníku, která leží např. na přímce a_0 , může obsahovat jen konečně mnoho vrcholů A_i daných trojúhelníků $A_i B_i C_i$, takže v šestiúhelníku

určitě najdeme trojúhelníček kladného obsahu, který do uvažovaného sjednocení nepatří. Obsah sjednocení uvažovaných trojúhelníků je proto i v tomto případě menší než $\frac{22}{9}$. Tím je důkaz hotov.

A – III – 4

Ze zadání plyne, že každá z rovnic $f(x) = 0$, $g(x) = 0$ má dva reálné kořeny, přitom všechny čtyři kořeny obou uvažovaných rovnic jsou navzájem různé. Označme x_1 , x_2 kořeny rovnice $f(x) = 0$. Platí tedy $f(x) = a(x - x_1)(x - x_2)$, kde a je reálné číslo, $a \neq 0$. Číslo x_1 je podle zadání rovněž kořenem rovnice $g(f(x)) = 0$, platí tudíž $g(f(x_1)) = g(0) = 0$. Odtud vyplývá, že rovnice $g(x) = 0$ má jeden kořen 0. Označme b ($b \neq 0$) druhý kořen této rovnice. Je tedy $g(x) = cx(x - b)$, kde c je reálné číslo, $c \neq 0$. Čísla 0 a b jsou podle zadání rovněž kořeny rovnice $g(f(x)) = 0$:

$$g(f(0)) = cf(0)(f(0) - b) = 0 \quad a \quad g(f(b)) = cf(b)(f(b) - b) = 0.$$

Jelikož čísla 0 a b nemohou být kořeny rovnice $f(x) = 0$, plyne odtud $f(0) = f(b) = b$.

Na číselné ose jsou proto jak body 0 a b , tak i body x_1 a x_2 souměrně sdružené podle x -ové souřadnice vrcholu paraboly $y = f(x)$. Čísla 0, b , x_1 a x_2 (tvořící dle zadání aritmetickou posloupnost) mohou tedy být uspořádána dvěma způsoby:

• Čísla x_1 a x_2 leží uvnitř intervalu s krajními body 0 a b . Pak $x_1 = \frac{1}{3}b$ a $x_2 = \frac{2}{3}b$ (při vhodné volbě indexů), tudíž

$$b = f(0) = a\left(-\frac{b}{3}\right)\left(-\frac{2b}{3}\right) = \frac{2ab^2}{9},$$

takže $b = \frac{9}{2a}$ a

$$f(x) = a\left(x - \frac{3}{2a}\right)\left(x - \frac{3}{a}\right) = ax^2 - \frac{9}{2}x + \frac{9}{2a}.$$

• Čísla 0 a b leží uvnitř intervalu s krajními body x_1 a x_2 . Pak $x_1 = -b$ a $x_2 = 2b$ (při vhodné volbě indexů), tudíž

$$b = f(0) = ab(-2b) = -2ab^2,$$

takže $b = -\frac{1}{2a}$ a

$$f(x) = a\left(x - \frac{1}{2a}\right)\left(x + \frac{1}{a}\right) = ax^2 + \frac{1}{2}x - \frac{1}{2a}.$$

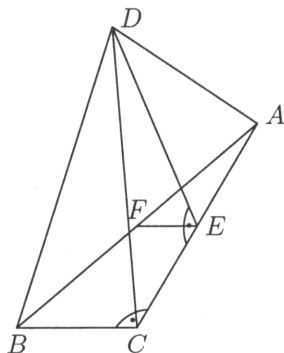
Závěr: Úloze vyhovují všechny kvadratické funkce f tvaru

$$f(x) = ax^2 - \frac{9}{2}x + \frac{9}{2a} \quad \text{nebo} \quad f(x) = ax^2 + \frac{1}{2}x - \frac{1}{2a},$$

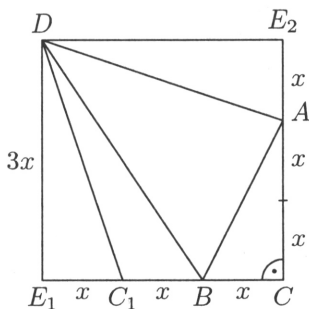
kde a je libovolné nenulové reálné číslo.

A – III – 5

Označme $ABCD$ uvažovaný jehlan s podstavou ABC , kde $|\sphericalangle ACB| = 90^\circ$. Podle textu úlohy byl model rozříznut podél obou odvěsen AC a BC podstavy a dále podél těžnice z vrcholu D jedné ze stěn BCD , ACD . Při řezu podél těžnice ve stěně ABD by totiž nebylo možné rozvinout model do roviny. Bez újmy na obecnosti předpokládejme dále, že řez je veden podél těžnice DE ve stěně ACD (obr. 55), kdy po rozvinutí do roviny vznikne útvar s hranicí $BCAE_2DE_1C_1B$ a pravým úhlem u vrcholu C (obr. 56). Protože tento útvar je čtverec (označme ho C), jsou úhly AE_2D a DE_1C_1 pravé (žádný z nich nemůže být přímý, neboť jejich součet je 180°). Proto je těžnice DE trojúhelníku ACD zároveň jeho výškou a body E_1 , E_2 jsou vrcholy čtverce C .

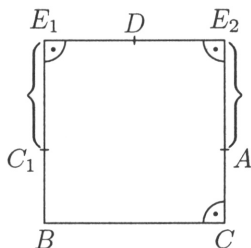


Obr. 55



Obr. 56

Kdyby vrcholy E_1 a E_2 čtverce C byly sousední, z rovnosti $|E_1C_1| = |E_2A|$ a z toho, že C má vrchol C , by plynulo, že $C = CE_2E_1B$, a tak $|BC| = |BE_1|$, což ale odporuje rovnosti $|BC| = |BC_1|$ (obr. 57). Tak



Obr. 57

jsme (sporem) dokázali, že vrcholy E_1 a E_2 čtverce C *nej*sou sousední, proto k vrcholům C patří (kromě bodů E_1, E_2 a C) nutně bod D (z úseku E_2DE_1 hranice $BCAE_2DE_1C_1B$).

Popsané body rozdělují hranici čtverce $C = CE_2DE_1$ na úseky, jejichž délky jsou vyznačeny na obr. 56 pomocí výhodného označení $x = \frac{1}{3}a$. Délky ostatních hran jehlanu spočteme podle Pythagorovy věty:

$$\begin{aligned} |DC| &= |DA| = \sqrt{(3x)^2 + (x)^2} = x\sqrt{10}, \\ |DB| &= \sqrt{(3x)^2 + (2x)^2} = x\sqrt{13}, \quad |AB| = x\sqrt{5}. \end{aligned}$$

Abychom zjistili objem jehlanu $ABCD$, potřebujeme určit velikost jeho tělesové výšky.

Označíme-li F střed hrany AB , vidíme, že hrana AC je kolmá na rovinu EFD , neboť $AC \perp BC \parallel EF$ a $AC \perp DE$. Rovina EFD je tedy kolmá na základnu ABC .

Tělesová výška jehlanu je proto výškou (z vrcholu D) trojúhelníku DEF . Protože DF tvoří těžnici trojúhelníku ABD , ze známého vzorce pro velikost těžnice dostaneme

$$2|DF|^2 = |DA|^2 + |DB|^2 - \frac{1}{2}|AB|^2 = \frac{41}{2}x^2,$$

takže strany trojúhelníku DEF mají délky $|DF| = \frac{1}{2}x\sqrt{41}$, $|DE| = 3x$ a $|EF| = \frac{1}{2}|BC| = \frac{1}{2}x$. Podle Heronova vzorce je obsah S takového trojúhelníku roven

$$\begin{aligned} S &= \frac{x^2}{4} \sqrt{\left(3 + \frac{1}{2} + \frac{\sqrt{41}}{2}\right)\left(3 + \frac{1}{2} - \frac{\sqrt{41}}{2}\right)\left(3 + \frac{\sqrt{41}}{2} - \frac{1}{2}\right)\left(\frac{\sqrt{41}}{2} + \frac{1}{2} - 3\right)} = \\ &= \frac{x^2}{16} \sqrt{(7 + \sqrt{41})(7 - \sqrt{41})(5 + \sqrt{41})(\sqrt{41} - 5)} = \frac{x^2\sqrt{2}}{2}, \end{aligned}$$

a tak má jeho výška v z vrcholu D velikost $v = 2S/|EF| = 2x\sqrt{2}$. Objem V jehlanu $ABCD$ je proto roven

$$V = \frac{1}{3} \cdot \frac{1}{2} |AC| \cdot |BC| \cdot v = \frac{2\sqrt{2}}{3} x^3 = \frac{2\sqrt{2}}{81} a^3.$$

Závěr: Objem uvažovaného jehlanu je $\frac{2}{81}\sqrt{2}a^3$.

Poznámka. Ze čtverce lze popsáním způsobem čtyřstěn $ABCD$ požadovaných vlastností vytvořit, když je součet dvou ze tří předpokládaných stěnových úhlů při vrcholu D větší než úhel třetí. Protože jejich součet je 90° , stačí ověřit, že každý z těchto tří úhlů je menší než 45° . Nerovnost $|\sphericalangle CDB| < 45^\circ$ je zřejmá, zbylé dvě nerovnosti jsou důsledkem výpočtů, podle kterých $\cos |\sphericalangle ADB| = 9/\sqrt{130} > \frac{1}{2}\sqrt{2}$ a $\operatorname{tg} |\sphericalangle CDA| = \operatorname{tg} 2|\sphericalangle C_1DE_1| = \frac{3}{4} < 1$.

A – III – 6

Rovnost $1000a + 100b + 10c + d + 1 = (10a + c + 1)(10b + d + 1)$ lze upravit na tvar

$$100a(9 - b) + 10a(9 - d) + 10b(9 - c) + c(9 - d) = 0.$$

Přitom každý ze čtyř sčítanců na levé straně je nezáporné celé číslo, proto bude tato rovnost splněna, právě když bude každý z nich roven nule. Protože je $a > 0$, musí být $b = d = 9$ a následně i $c = 9$. V tom případě rovnici vyhovuje libovolná číslice a , $a \in \{1, 2, \dots, 9\}$. Řešením úlohy jsou tedy právě všechna následující čtyřmístná čísla: 1 999, 2 999, 3 999, 4 999, 5 999, 6 999, 7 999, 8 999 a 9 999.

Kategorie P

Texty úloh

P – I – 1

V továrně bude zahájena výroba nového typu výrobku. Výroba je popsána přesným technologickým plánem, jenž stanoví, jaké všechny výrobní operace je třeba vykonat a jak dlouho každá z těchto operací trvá. Pro každou výrobní operaci P je dále znám seznam operací, které musí být provedeny před začátkem operace P (nazveme je *předcházející operace*).

Majitelé továrny chtějí výrobu zorganizovat tak, aby bylo možné dokončit první nový výrobek v co nejkratším čase. V továrně lze současně provádět libovolný počet výrobních operací, každou operaci je však možné zahájit až po dokončení všech jí předcházejících operací. Jinak mohou být jednotlivé výrobní operace prováděny v jakémkoliv pořadí. Můžete předpokládat, že výroba podle zadaného technologického plánu je možná.

Napište program, který ze vstupního textového souboru TOVARNA.IN přečte technologický plán výroby nového výrobku a do výstupního textového souboru TOVARNA.OUT vypíše nejkratší čas, v jakém lze vyrobit první nový výrobek.

Soubor TOVARNA.IN obsahuje na prvním řádku počet výrobních operací N ($N \leq 100$). Na dalších N řádcích se nacházejí informace o jednotlivých výrobních operacích, na i -tém řádku o operaci číslo i . Informace o každé výrobní operaci se skládá z několika čísel oddělených mezerou. První číslo udává, jak dlouho operace trvá, dále následují čísla předcházejících operací. Řádek je ukončen číslem -1 . Pro zhotovení nového výrobku je třeba vykonat všech N zadaných výrobních operací. Jednotlivé operace je možné provádět v libovolném pořadí, každá operace však může být zahájena až po dokončení všech jí předcházejících operací.

Soubor TOVARNA.OUT bude obsahovat jediný řádek, na kterém bude zapsán čas, za jak dlouho od zahájení výroby lze dokončit první nový výrobek.

Příklad: Soubor TOVARNA.IN

```
4
2 3 -1
3 3 -1
3 -1
1 1 2 -1
```

Soubor TOVARNA.OUT

```
7
```

P – I – 2

Je dána množina $A = \{a_1, a_2, a_3, \dots, a_N\}$ ($N \geq 1$), kterou budeme nazývat *abeceda*. Prvky abecedy budeme nazývat *znaky*. *Řetězcem* nad abecedou A je konečná posloupnost prvků z množiny A (znaků). *Vybraný podřetězec* vznikne z řetězce vynecháním některých jeho znaků, přičemž pořadí zbývajících znaků řetězce zachováme beze změn. *Permutace* prvků množiny A je řetězec, v němž se každý prvek množiny A vyskytuje právě jednou.

Příklad: Necht $N = 3$, $A = \{a, b, c\}$. Potom z řetězce *abcabccb* můžeme vytvořit například vybrané podřetězce *ab*, *aac*, *bccb*, *abbb*, *abc*, *cab*. Z nich pouze *abc* a *cab* jsou permutace množiny A .

Soutěžní úloha. Nalezněte co nejkratší řetězec nad abecedou $A = \{a, b, c, \dots, o\}$ ($N = 15$), který obsahuje všechny permutace prvků množiny A jako vybrané podřetězce, přičemž každý znak se v něm vyskytuje nejvýše N -krát. Podrobně popište také způsob, jak jste tento řetězec našli, a přiložte a popište všechny algoritmy a programy, které jste přitom použili. Výsledný řetězec uložte do souboru RETEZEC.TXT.

Příklad: Například pro abecedu $A = \{a, b, c\}$ ($N = 3$) vyhovuje řetězec *acbacad*, protože se v něm nacházejí všechny permutace *abc*, *acb*, *bac*, *bca*, *cab*, *cba* jako vybrané podřetězce. Zároveň je tento řetězec nejkratším řetězcem nad abecedou A splňujícím tuto podmínku.

P – I – 3

V jisté zemi existuje N politických stran. Každá ze stran má mezi obyvatelstvem určité preference, přičemž žádné dvě strany nemají stejné preference. Agentury pro výzkum veřejného mínění dokážou provést průzkum, který pro libovolné dvě politické strany umožňuje přesně zjistit, která

z nich má preference nižší a která vyšší. Tento průzkum je však poměrně drahý a dá se použít vždy jen pro dvě vybrané politické strany.

Jedna agentura chce zjistit, která ze stran v zemi má nejvyšší a která nejnižší preference. Bude postupovat tak, že pro různé dvojice stran provede výše popsany průzkum veřejného mínění. Chce přitom vykonat co nejméně průzkumů.

Napište program, který bude řídit činnost agentury. Tento program nejprve přečte počet stran N ($1 \leq N \leq 100$). Strany si pro jednoduchost očíslujeme čísly $1, 2, \dots, N$. Program potom v cyklu vždy vypíše, pro které dvě strany se má provést průzkum, a následně přečte ze vstupu výsledek tohoto průzkumu — číslo strany s většími preferencemi. Když program tímto způsobem získá dostatek údajů potřebných k tomu, aby určil politické strany s nejvyššími a s nejnižšími preferencemi, vypíše odpověď a skončí.

Dbejte zejména na to, aby program dal vždy správnou odpověď a aby použil co nejméně průzkumů. Odhadněte, kolik nejvíce průzkumů může váš program potřebovat pro N stran.

Příklad: Uvedeme příklad činnosti programu pro tři strany, přičemž texty vypisované programem jsou od začátku řádku a odpovědi uživatele jsou zadány v řádcích začínajících znakem >.

Zadej počet stran:

> 3

Proveď průzkum 1,2

> 2

Proveď průzkum 1,3

> 3

Proveď průzkum 2,3

> 2

Nejvyšší preference má strana 2, nejnižší 1

P – I – 4

Minského registrový stroj

Minského registrový stroj je jednoduché výpočetní zařízení. K dispozici má několik registrů označených $R_0, R_1, R_2, \dots$, přičemž v každém registru může být uloženo jedno libovolně velké nezáporné celé číslo.

Minského registrový stroj může mít jeden nebo více vstupů, jejichž hodnoty jsou na začátku výpočtu uloženy v registrech $R_1, \dots, R_k$, kde k je počet těchto vstupů. Ostatní registry jsou na začátku výpočtu inicia-

lizovány na hodnotu nula. Po skončení výpočtu Minského registrového stroje je výsledkem hodnota uložená v registru R_0 .

Každý Minského registrový stroj je řízen pevně daným programem. V programu se mohou vyskytovat tyto příkazy:

- ▷ *Zvýšení* hodnoty v daném registru o 1.
- ▷ *Snížení* hodnoty v daném registru o 1 (pokud je v registru nula, hodnota se nezmění).
- ▷ *Test na nulu* zjistí, zda je v daném registru nula.

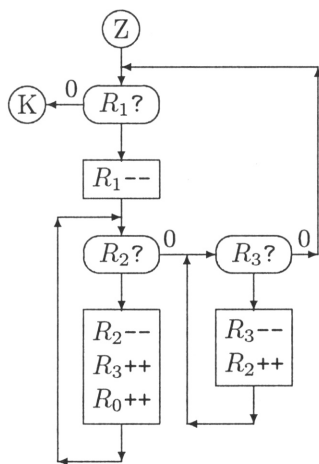
Registr je určen svým číslem. Číslo registru je v příkazu vždy pevně zadáno, není tedy možné k určení registru použít obsah jiného registru.

Programy budeme zakreslovat do schémat, přičemž zvýšení hodnoty registru R_i budeme značit obdélníkem s nápisem R_i++ , snížení hodnoty registru R_i budeme značit obdélníkem s nápisem R_i-- , test na nulu v registru R_i značíme oválem s nápisem $R_i?$. Začátek programu označujeme písmenem Z v kroužku a konec programu písmenem K v kroužku (program může mít i více konců, ale jen jeden začátek). Jednotlivé příkazy jsou pospojovány šípkami, které určují tok řízení programu. Z obdélníku vychází vždy jediná šipka. Z oválu vycházejí dvě šipky, přičemž jedna z nich je označena nulou (po ní pokračuje výpočet tehdy, když byla v testovaném registru nula, v opačném případě výpočet sleduje druhou šipku). Jestliže v programu za sebou následuje několik příkazů zvýšení nebo snížení, můžeme je napsat pod sebe do jednoho obdélníku.

Příklad: Sestrojte stroj, který bude mít na vstupu čísla x a y (uložená v registrech R_1 a R_2) a který vypočítá jejich součin $x \cdot y$ (uloží ho do registru R_0).

ŘEŠENÍ. Stroj řešící tuto úlohu vidíme na obrázku. Tento stroj vždy odečte od registru R_1 jedničku a k registru R_0 přičte číslo y (obsah registru R_2). To provádí tak dlouho, dokud v registru R_1 není nula. Číslo y se tedy do registru R_0 přičte celkem x -krát, takže dostaneme správný výsledek.

Podívejme se nyní na to, jak se provádí přičtení čísla y k R_0 . Opět v cyklu snižujeme hodnotu registru R_2 a zvyšujeme hodnotu registru R_0 , dokud nemáme v registru R_2 nulu. Tehdy jsme



do R_0 přidali přesně y . Problém však spočívá v tom, že v registru R_2 máme nyní nulu, ale pro další průběh výpočtu tam potřebujeme vložit zpět hodnotu y . Proto v cyklu kromě snížení R_2 a zvýšení R_0 ještě zvýšíme R_3 . Tím si zajistíme, že po skončení cyklu máme hodnotu y uloženou v registru R_3 . Nyní ji musíme odtamtud přenést zpět do registru R_2 . To provedeme v dalším cyklu, v němž současně snižujeme R_3 a zvyšujeme R_2 , dokud v R_3 není nula. V tom okamžiku máme v R_2 opět původní hodnotu y a můžeme začít s dalším kolem výpočtu.

Soutěžní úlohy:

- Sestrojte Minského registrový stroj s jedním vstupem n , který vypočítá číslo 2^n .
- Sestrojte Minského registrový stroj s jedním vstupem n , který vypočítá Fibonacciho číslo F_n . Váš stroj přitom může používat pouze registry R_0, R_1, R_2, R_3 .

Fibonacciho čísla jsou definována následujícím vztahem:

$$F_n = \begin{cases} 1, & \text{jestliže } n < 2, \\ F_{n-1} + F_{n-2}, & \text{jestliže } n \geq 2. \end{cases}$$

P – II – 1

Síť

Ve výpočetním středisku mají N počítačů očíslovaných od 1 do N . Počítače jsou navzájem propojeny jednosměrnými spoji. Jestliže je „počítač i “ připojen k počítači j , znamená to, že počítač i může posílat zprávy počítači j (ale ne naopak). Jestliže je počítač i připojen k počítači j , může, ale nemusí být připojen také počítač j k počítači i . (Žádný počítač není připojen sám k sobě.)

Do výpočetního střediska nyní koupili nový centrální počítač. Je třeba zajistit, aby tento nový počítač mohl poslat zprávu všem ostatním počítačům. Musíme ho proto připojit k některým dalším počítačům tak, aby se z něj dala poslat zpráva libovolnému počítači p buď přímo (tj. centrální počítač je připojen k počítači p), nebo přes několik jiných počítačů (tj. existují počítače $a_1, a_2, \dots, a_k$ takové, že centrální počítač je připojen k počítači a_1 , pro $i = 1, 2, \dots, k-1$ je počítač a_i připojen k počítači a_{i+1} a počítač a_k je připojen k počítači p). Kvůli úspoře kabelů je přitom nutné minimalizovat počet počítačů, ke kterým bude nový centrální počítač připojen.

Soutěžní úloha. Je dán počet počítačů N . Pro každý počítač i ($1 \leq i \leq N$) je dán počet počítačů d_i , k nimž je daný počítač připojen, a dále je dán seznam těchto počítačů $a_{i1}, a_{i2}, \dots, a_{id_i}$. Napište program, který vypíše seznam počítačů, k nimž je třeba připojit nový centrální počítač tak, aby jejich počet byl minimální. Pokud je možné centrální počítač připojit více způsoby, vypíše libovolný jeden z nich.

Příklad: vstup:

$N = 7$

$d_1 = 1$, připojení: 2

$d_2 = 2$, připojení: 1 5

$d_3 = 1$, připojení: 6

$d_4 = 2$, připojení: 3 6

$d_5 = 1$, připojení: 3

$d_6 = 1$, připojení: 5

$d_7 = 0$, připojení: $\emptyset$

výstup:

2,4,7

(Jiné možné řešení je 1,4,7.)

P – II – 2

Posloupnost

Nechť $A = (a_1, a_2, \dots, a_M)$ je posloupnost celých čísel. Posloupnost A' nazveme *vybranou podposloupností* této posloupnosti, jestliže vznikne z posloupnosti A vynecháním některých jejích členů, přičemž pořadí ostatních prvků zachováme. Společná vybraná podposloupnost posloupností A , B je každá taková posloupnost C , která je vybranou podposloupností obou posloupností A i B .

Příklad: Nechť $A = (1, 11, 2, 1, 4, 99)$, $B = (9, 4, 1, 2, 7, 1, 99)$. Posloupnost $(1, 2, 1, 99)$ je společnou vybranou podposloupností posloupností A , B . Posloupnost $(1, 2, 4)$ je vybranou podposloupností posloupnosti A , ale není vybranou podposloupností posloupnosti B , takže není ani společnou vybranou podposloupností A a B .

Soutěžní úloha. Máme dány dvě posloupnosti celých čísel $A = (a_1, a_2, \dots, a_M)$ a $B = (b_1, b_2, \dots, b_N)$ s délkami M , resp. N . Tyto posloupnosti jsou uloženy v polích $a[1..M]$, resp. $b[1..N]$, jejichž obsah není dovoleno měnit. Uvažujme takovou společnou vybranou podposloupnost posloupností A a B , ve které součet všech jejích členů je největší možný. Napište program, který vypíše součet členů takovéto posloupnosti.

Poznámka. Existuje algoritmus, který tuto úlohu řeší v čase úměrném $M \cdot N$ a paměti, jejíž velikost je úměrná menšímu z čísel M , N .

Příklad:

vstup:

$M = 6, N = 7$

$A = (1, 11, 2, 1, 4, 99)$

$B = (9, 4, 1, 2, 7, 1, 99)$

výstup:

103

(Vybrané podposloupnosti se součtem 103 existují dvě: 4, 99 a 1, 2, 1, 99.)

P – II – 3

Volby

V jisté zemi právě skončily volby. Každý volič v nich hlasoval pro jednoho z navržených kandidátů. Vítězem voleb se stane kandidát, pro kterého hlasovala nadpoloviční většina voličů. Máte k dispozici výsledky hlasování jednotlivých voličů a máte co nejrychleji zjistit, který kandidát vyhrál volby.

Soutěžní úloha. Voleb se zúčastnilo N voličů očíslovaných od 1 do N a M kandidátů očíslovaných od 1 do M (někteří kandidáti však nemuseli získat ani jeden hlas). Výsledky hlasování jsou uloženy v poli a , přičemž platí, že volič i ($1 \leq i \leq N$) hlasoval pro kandidáta číslo a_i . Obsah tohoto pole nesmíte měnit.

Napište program, který zjistí, zda existuje kandidát, pro kterého hlasovalo více než $\frac{1}{2}N$ voličů. Jestliže takový kandidát existuje, program vypíše jeho číslo, pokud ne, program vypíše, že takový kandidát neexistuje.

Poznámka. Čísla M a N mohou být *velmi* velká. Snažte se proto, aby váš program pracoval co nejefektivněji a aby používal co nejméně paměti.

Příklad: vstup:

$N = 9$

$A = \{2, 3, 2, 2, 3, 50\,001, 3, 2, 2\}$

vstup:

$N = 10$

$A = \{2, 2, 1, 4, 3, 2, 1, 2, 100, 2\}$

výstup:

Vyhrál kandidát 2.

výstup:

Nevyhrál žádný kandidát.

P – II – 4

Minského registrové stroje

(Oproti domácímu kolu je popis registrového stroje rozšířen o nový druh příkazu — výpočet hodnoty funkce.)

Minského registrový stroj je jednoduché výpočetní zařízení. K dispozici má několik registrů označených $R_0, R_1, R_2, \dots$, přičemž v každém registru může být uloženo jedno libovolné velké nezáporné celé číslo.

Minského registrový stroj může mít jeden nebo více vstupů, jejichž hodnoty jsou na začátku výpočtu uloženy v registrech $R_1, \dots, R_k$, kde k je počet těchto vstupů. Ostatní registry jsou na začátku výpočtu inicializovány na hodnotu nula. Po skončení výpočtu Minského registrového stroje je výsledkem hodnota uložená v registru R_0 .

Každý Minského registrový stroj je řízen pevně daným programem. V programu se mohou vyskytovat tyto příkazy:

- ▷ *Zvýšení* hodnoty v daném registru o 1.
- ▷ *Snížení* hodnoty v daném registru o 1 (pokud je v registru nula, hodnota se nezmění).
- ▷ *Test na nulu* zjistí, zda je v daném registru nula.
- ▷ *Výpočet hodnoty funkce F* . Tento příkaz vezme obsah určeného vstupního registru (označme tuto hodnotu registru jako x), vypočítá hodnotu $F(x)$ a uloží ji do určeného výstupního registru. Obsah vstupního registru po provedení tohoto příkazu není definován (může v něm být libovolné nezáporné číslo).

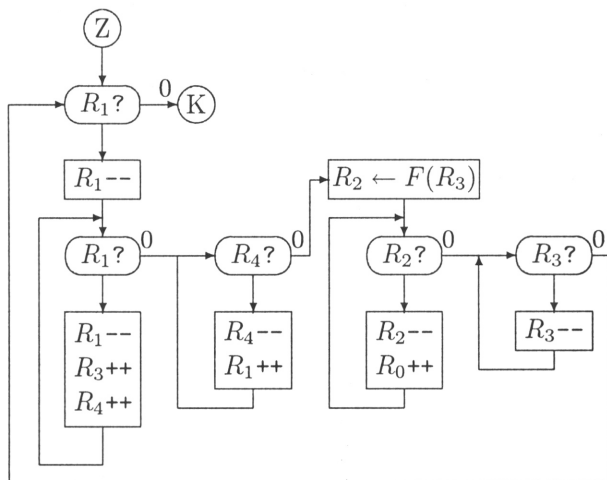
Registr je určen svým číslem. Číslo registru je v příkazu vždy pevně zadáno, není tedy možné k určení registru použít obsah jiného registru.

Programy budeme zakreslovat do schémat, přičemž zvýšení hodnoty registru R_i budeme značit obdélníkem s nápisem R_i++ , snížení hodnoty registru R_i budeme značit obdélníkem s nápisem R_i-- , test na nulu v registru R_i značíme oválem s nápisem $R_i?$. Příkaz na výpočet funkce F se vstupním registrem R_i a výstupním registrem R_j značíme obdélníkem s nápisem $R_j \leftarrow F(R_i)$ (vstupní a výstupní registr musí být navzájem různé).

Začátek programu označujeme písmenem Z v kroužku a konec programu písmenem K v kroužku (program může mít i více konců, ale jen jeden začátek). Jednotlivé příkazy jsou pospojovány šipkami, které určují tok řízení programu. Z obdélníku vychází vždy jediná šipka. Z oválu vycházejí dvě šipky, přičemž jedna z nich je označena nulou (po ní pokračuje výpočet tehdy, když byla v testovaném registru nula, v opačném případě výpočet sleduje druhou šipku). Jestliže v programu za sebou následuje několik příkazů zvýšení, snížení a výpočtu hodnoty funkce, můžeme je napsat pod sebe do jednoho obdélníku.

Příklad. Nechť $F(x)$ je předem daná, ale neznámá funkce. Sestrojte

registrový stroj, který dostane na vstupu jedno číslo n (uložené v registru R_1) a který spočítá součet $F(0) + F(1) + \dots + F(n-1)$ (uloží ho do registru R_0).



ŘEŠENÍ. Stroj řešící zadanou úlohu vidíme na obrázku. Tento stroj pracuje v cyklu. Při každém průchodu cyklem sníží hodnotu uloženou v registru R_1 o jedničku, vypočítá hodnotu funkce $F(R_1)$ a přičte ji k registru R_0 . Budeme potřebovat několik pomocných registrů. Registr R_3 bude sloužit jako vstupní registr pro výpočet funkce F . Do tohoto registru zkopírujeme obsah registru R_1 tak, že nejprve „přesypeme“ obsah registru R_1 do registrů R_3 a R_4 a potom zpět obsah R_4 do R_1 . Nyní můžeme použít příkaz na výpočet funkce F . Výsledek výpočtu funkce se zapíše do registru R_2 . Na dokončení jednoho průchodu cyklem nyní stačí „přisypat“ obsah R_2 do registru R_0 a vynulovat registr R_3 .

Soutěžní úlohy:

- a) Nechť $F(x)$ je nějaká předem daná, ale neznámá rostoucí funkce. (Funkce F je rostoucí, jestliže pro všechna x platí $F(x+1) > F(x)$.) Sestrojte registrový stroj s jedním vstupem y , který bude počítat funkci $G(y)$ inverzní k funkci F . Přesněji řečeno, výsledek výpočtu registrového stroje $G(y) = x$, kde x je nejmenší takové nezáporné celé číslo, pro které platí $F(x) \geq y$.

Příklad. Předpokládejme, že $F(x) = x^2 + 6$. Potom pro vstupní hodnotu $y = 5$ by měl stroj vypočítat výsledek 0 (protože $F(0) \geq 5$).

a 0 je nejmenší nezáporné celé číslo). Pro vstup $y = 10$ je správným výstupem 2, pro vstup $y = 11$ je správná výstupní hodnota 3.

- b) Sestrojte Minského registrový stroj s jedním vstupem n , který zjistí, zda je zápis čísla n ve dvojkové soustavě palindromem. Palindrom je posloupnost cifer, která je stejná, když se čte zepředu i zezadu (není povoleno dopisovat nuly na začátek zápisu čísla). Stroj bude na konci výpočtu obsahovat v registru R_0 jedničku v případě, že dvojkový zápis čísla n je palindrom, nebo nulu v případě, že palindromem není.

Příklad. Zápisy čísel $0_{10} = 0_2$ nebo $21_{10} = 1010_2$ ve dvojkové soustavě jsou palindromy, zápisy čísel $10_{10} = 1010_2$ a $11_{10} = 1011_2$ palindromy nejsou.

P – III – 1

Jednosměrky

V centru města je N důležitých křižovek očíslovaných od 1 do N , které jsou pospojovány M cestami. Cestou rozumíme asfaltový koberec spojující dvě konkrétní křižovatky. Mezi libovolnými dvěma křižovatkami může vést nejvýše jedna cesta. Všechny cesty jsou obousměrné a na každou křižovátku se lze po těchto cestách dostat ze všech ostatních křižovek, tzn. pro libovolnou dvojici křižovek i a j , $i \neq j$, existují křižovatky $a_1, a_2, \dots, a_k$ takové, že $a_1 = i$, $a_k = j$ a pro $i = 1, 2, \dots, k-1$ vede mezi křižovatkami a_i a a_{i+1} cesta. Pro účely této úlohy budeme křižovatkou označovat i místo, do něhož vede jen jediná cesta (případně dvě cesty).

V rámci zvyšování dopravní bezpečnosti se městská rada rozhodla co nejvíce cest „zjednosměrnit“. To znamená, že pokud mezi křižovatkami i a j existuje cesta, po „zjednosměrnění od i k j “ (označujeme $i \rightarrow j$) bude po této cestě povoleno jet z křižovatky i na křižovátku j , ale ne naopak. Městská rada má jedinou podmínku: z každé křižovatky musí být možné dojet na všechny ostatní křižovatky při dodržení příkázaného směru jízdy.

Soutěžní úloha. Je dán počet křižovek N a počet cest M . Dále je dáno takových M dvojic křižovek $\{i, j\}$, $i \neq j$, že mezi křižovatkami i a j vede cesta. Napište program, který vypíše všechny „zjednosměrněné“ cesty a jejich povolený směr jízdy tak, aby byl počet zbývajících obousměrných cest minimální. Jestliže existuje více možných řešení, vypíše jedno libovolné z nich.

Příklad: vstup: $N = 8, M = 10$ *výstup:*

$\{1, 2\}$	$2 \rightarrow 3$
$\{2, 3\}$	$3 \rightarrow 8$
$\{2, 8\}$	$8 \rightarrow 2$
$\{3, 4\}$	$4 \rightarrow 5$
$\{3, 8\}$	$5 \rightarrow 6$
$\{4, 5\}$	$6 \rightarrow 7$
$\{4, 7\}$	$7 \rightarrow 4$
$\{5, 6\}$	$5 \rightarrow 7$
$\{5, 7\}$	(Jiným možným řešením je
$\{6, 7\}$	přesměrovat $\{5, 7\}$ na $7 \rightarrow 5$.)

P – III – 2

Volby

V jisté zemi právě skončily volby. Každý volič v nich hlasoval pro jednoho z navržených kandidátů. Za poslance místního shromáždění jsou zvoleni všichni kandidáti, pro které hlasovala více než jedna k -tina všech voličů. Všimněte si, že v takovýchto volbách je možné zvolit nejvýše $k - 1$ kandidátů (možná jich ale bude zvoleno méně). Máte k dispozici výsledky hlasování jednotlivých voličů a máte co nejrychleji zjistit, kteří kandidáti byli zvoleni do zastupitelstva.

Soutěžní úloha. Voleb se zúčastnilo N voličů očíslovaných od 1 do N a M ($M \leq N$) kandidátů očíslovaných od 1 do M (někteří kandidáti však nemuseli získat ani jeden hlas). Výsledky hlasování jsou uloženy v poli a , přičemž platí, že volič i ($1 \leq i \leq N$) hlasoval pro kandidáta číslo $a[i]$. Obsah tohoto pole nesmíte měnit. Dále máte dáno číslo $k \geq 2$ (velmi malé v porovnání s N). Napište program, který vypíše čísla všech kandidátů, pro něž hlasovalo více než N/k voličů.

Poznámka. Čísla M a N mohou být *velmi* velká. Snažte se proto, aby váš program pracoval co nejefektivněji a aby používal co nejméně paměti.

<i>Příklad 1: vstup:</i>	<i>výstup:</i>
$N = 11, k = 3, M = 7$	Zvoleni jsou kandidáti 1, 2.
$A = (1, 2, 3, 2, 1, 1, 7, 2, 3, 2, 1)$	

<i>Příklad 2: vstup:</i>	<i>výstup:</i>
$N = 8, k = 4, M = 4$	Nebyl zvolen žádný kandidát.
$A = (1, 2, 3, 3, 4, 2, 4, 1)$	

Minského registrové stroje

Definice. (Oproti domácímu kolu se studijní text liší příkladem sestavení Minského stroje z bloků.)

Minského registrový stroj je jednoduché výpočetní zařízení. K dispozici má několik registrů označených $R_0, R_1, R_2, \dots$, přičemž v každém registru může být uloženo jedno libovolně velké nezáporné celé číslo.

Minského registrový stroj může mít jeden nebo více vstupů, jejichž hodnoty jsou na začátku výpočtu uloženy v registrech $R_1, \dots, R_k$, kde k je počet těchto vstupů. Ostatní registry jsou na začátku výpočtu inicializovány na hodnotu nula. Po skončení výpočtu Minského registrového stroje je výsledkem hodnota uložená v registru R_0 .

Každý Minského registrový stroj je řízen pevně daným programem. V programu se mohou vyskytovat tyto příkazy:

- ▷ *Zvýšení* hodnoty v daném registru o 1.
- ▷ *Snížení* hodnoty v daném registru o 1 (pokud je v registru nula, hodnota se nezmění).
- ▷ *Test na nulu* zjistí, zda je v daném registru nula.

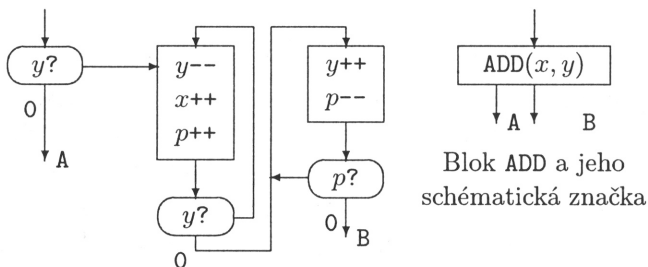
Registr je určen svým číslem. Číslo registru je v příkazu vždy pevně zadáno, není tedy možné k určení registru použít obsah jiného registru.

Programy budeme zakreslovat do schémat, přičemž zvýšení hodnoty registru R_i budeme značit obdélníkem s nápisem R_i++ , snížení hodnoty registru R_i budeme značit obdélníkem s nápisem R_i-- , test na nulu v registru R_i značíme oválem s nápisem $R_i?$. Začátek programu označujeme písmenem Z v kroužku a konec programu písmenem K v kroužku (program může mít i více konců, ale jen jeden začátek). Jednotlivé příkazy jsou pospojovány šipkami, které určují tok řízení programu. Z obdélníku vychází vždy jediná šipka. Z oválu vycházejí dvě šipky, přičemž jedna z nich je označena nulou (po ní pokračuje výpočet tehdy, když byla v testovaném registru nula, v opačném případě výpočet sleduje druhou šipku). Jestliže v programu za sebou následuje několik příkazů zvýšení nebo snížení, můžeme je napsat pod sebe do jednoho obdélníku.

Použití bloků. Při kreslení složitějších schémat se nám může stát, že některé části potřebujeme použít vícekrát. Abychom se vyhnuli opakování kreslení stejných skupin příkazů, budeme je seskupovat do bloků. *Blok* je skupina příkazů s jedním určeným počátečním příkazem a s jednou nebo více výstupními šipkami. Každý blok musíme nejprve definovat, tj. nakreslit příslušnou skupinu příkazů. Definovaný blok potom

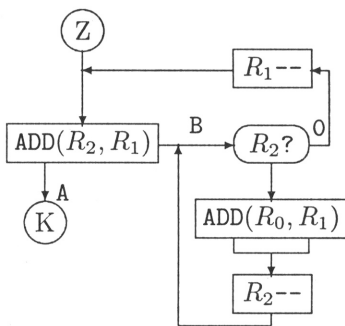
můžeme použít na libovolném místě při kreslení schématu registrového stroje (případně při definování jiných bloků). Blok značíme obdélníkem s vepsaným jménem bloku, z tohoto obdélníku vychází příslušný počet šipek. V definici bloku můžeme některé registry označit proměnnými. Za tyto proměnné se dosadí konkrétní registry až na místě, kde se blok použije (v tom případě napíšeme do obdélníku za jméno bloku do závorek registry, které se mají dosadit za jednotlivé proměnné). Proměnným, za které nedosadíme žádný registr, se nakonec přiřadí nepoužité registry (každá kopie proměnné má vlastní registr). Použití bloků nejlépe objasní příklad.

Úloha: Sestrojte stroj, který bude mít na vstupu číslo n (uložené v registru R_1) a který vypočítá součet druhých mocnin čísel od 0 do n (výsledný součet uloží do registru R_0).



ŘEŠENÍ. Nejprve nadefinujeme blok $\text{ADD}(x, y)$, který k registru x přičte obsah registru y (obsah registru y se přitom zachová). Tento blok bude používat jeden pomocný registr p , kterému není explicitně přiřazen žádný konkrétní registr. Bude mít dvě výstupní šipky: jestliže je v registru y nula, použije se šipka A, v opačném případě použijeme šipku B.

Náš registrový stroj bude pracovat následovně: K obsahu registru R_0 (který je na začátku nulový) budeme postupně přičítat $n^2, (n-1)^2, \dots, 1^2$. V registru R_1 bude vždy uloženo číslo x , jehož druhou mocninu právě přičítáme. Jestliže $x = 0$, končíme (větev A bloku $\text{ADD}(R_2, R_1)$). V opačném případě pomocí $\text{ADD}(R_2, R_1)$



zkopírujeme obsah R_1 do R_2 (obsah R_2 byl předtím nulový) a pomocí cyklu a dalšího bloku *ADD* přičítáme do R_0 x -krát číslo x . (Všimněte si, že v R_2 je po proběhnutí tohoto cyklu opět nula.) Tento postup opakuje, přičemž snižujeme obsah R_1 o jedničku.

Soutěžní úloha. Konečná množina nezáporných celých čísel $M = \{c_1, c_2, \dots, c_n\}$ se dá jednoznačně zakódovat do jednoho nezáporného celého čísla m takto: $m = \sum_{i=1}^n 2^{c_i}$. (V množině se může každý prvek nacházet nejvýše jednou, tzn. $c_i \neq c_j$ pro $i \neq j$.) Jestliže si představíme zápis čísla m ve dvojkové soustavě, potom číslo c patří do množiny M právě tehdy, když c -tý bit zápisu m je jednička. Bity číslujeme zprava doleva, tj. nejpravější bit má číslo 0, jeho levý soused číslo 1, atd.

Sestrojte Minského registrový stroj, který dostane na vstupu kód m nějaké množiny M a číslo s a zjistí, zda lze číslo s vyjádřit jako součet některých prvků množiny M , tj. zda existuje taková množina $M' \subset M$, že $\sum_{c \in M'} c = s$. Číslo m je na začátku výpočtu uloženo v registru R_1 , číslo s v registru R_2 . Po skončení výpočtu musí registr R_0 obsahovat jedničku v případě, že s je součtem některých prvků množiny M a nulu v opačném případě. Součet nula prvků má definitoricky hodnotu nula.

Příklad. Pro hodnoty $m = 203$ (11001011 v dvojkové soustavě, tzn. $M = \{0, 1, 3, 6, 7\}$), $s = 10$ je správný výsledek 1 ($10 = 3 + 7$). Pro $m = 203$, $s = 12$ je výsledek 0 (protože 12 nelze získat součtem čísel vybraných z množiny M). Pro m libovolné, $s = 0$ je správný výsledek 1.

P – III – 4

Pramen

Program: pramen.pas / pramen.cpp
Vstup: pramen.in
Výstup: pramen.out

V jistém lázeňském městečku mají N minerálních pramenů očíslovaných $1, 2, \dots, N$. Prameny jsou navzájem pospojovány pěšinami, přičemž pěšina vede mezi každou dvojicí pramenů. Turisté obvykle chtějí ochutnat vodu z každého pramenu, ale protože pěšiny nejsou nijak označené, občas se stane, že některého zbloudilého turistu najdou až ve vedlejší dolině. Správce lázní se proto rozhodl umístit ke každému pramenu právě jednu směrovku ukazující na další pramen, a to tak, že turista začínající svou

procházku u libovolného pramenu a pokračující ve své cestě podle směrovek obejde postupně všechny prameny.

Správce tedy nechal vyrobít N směrovek s čísly pramenů $1, \dots, N$. Dělníci ale příliš nepřemýšleli a rozmístili směrovky k pramenům úplně náhodně. Tak se stalo, že u každého pramene sice byla umístěna právě jedna směrovka ukazující k nějakému prameni, ale mohlo se stát, že pokud vyšel turista od některého pramene, tak se k některým jiným pramenům vůbec nedostal. Mohlo se dokonce stát, že směrovka u některého pramene ukazovala nazpět na ten samý pramen, u kterého byla umístěna.

Správce ví, že na dělníky se může spolehnout jen v této jednoduché operaci: vzájemně zaměnit směrovky u dvou pevně daných pramenů p a q , tzn. jestliže směrovka u pramene p ukazovala na pramen i a směrovka u pramene q ukazovala na pramen j , tak po této výměně bude u pramene p směrovka ukazovat na pramen j a u pramene q bude směrovka ukazovat na pramen i .

Napište program pro správce lázní, který zjistí, jaké výměny směrovek mají dělníci provést, aby bylo možné podle směrovek obejít všechny prameny a aby přitom počet provedených výměn byl nejmenší možný.

Vstupní soubor: Vstupní soubor `pramen.in` obsahuje na prvním řádku počet pramenů N ($1 \leq N \leq 30\,000$). Následuje N čísel (oddělených mezerami nebo konci řádků), přičemž i -tým z nich je číslo pramene, na který ukazuje směrovka umístěná u i -tého pramenu.

Výstupní soubor: První řádek výstupního souboru `pramen.out` obsahuje minimální počet výměn M , které je třeba provést. Následuje M řádků popisujících jednotlivé výměny: $(i + 1)$ -tý řádek výstupního souboru ($1 \leq i \leq N$) obsahuje dvě čísla p, q oddělená jednou mezerou, označující dvojici pramenů, u nichž je třeba vyměnit směrovky. Výměny se provádějí v uvedeném pořadí.

Příklad.

<code>pramen.in</code>	7	<code>pramen.out</code>	2
	5 1 4 6 2 3 7		1 3
			7 1

Poznámka. Uvedený výstupní soubor je jedním z možných správných výstupních souborů.

Kina

Program: kina.pas / kina.cpp

Vstup: kina.in

Výstup: kina.out

V jisté zemi právě začal Mezinárodní Filmový Festival Umělecké Kinematografie (MFF UK). V zemi je n ($1 \leq n \leq 50$) měst označených čísly $1, 2, \dots, n$. V každém městě je jedno kino. O každém kině je známo, který film z nabídky filmů MFF UK očíslovaných $1, 2, \dots, t$ ($1 \leq t \leq n$) v něm promítají. Konkrétně, v kině ve městě číslo i promítají každý večer stejný film číslo f_i . Města jsou navzájem pospojována m obousměrnými cestami. Pro každou cestu c ($1 \leq c \leq m$) známe čísla měst z_c a k_c ($z_c \neq k_c$), která tato cesta spojuje, a také známe její délku l_c ($0 \leq l_c \leq 1\,000$). Mezi každou dvojicí měst vede nejvýše jedna cesta.

Předpokládejme, že máte rozpis k večerů ($0 \leq k \leq 1\,000$), přičemž pro každý večer i ($1 \leq i \leq k$) je stanoveno číslo filmu p_i , který byste chtěli v ten večer vidět. Některé filmy se v rozpisu mohou vyskytnout i vícekrát. Abyste mohli večer sledovat naplánovaný film, musíte se během dne dopravit do některého města (nezáleží na tom do kterého), v němž tento film uvádějí. Protože neradi cestujete, chtěli byste, aby celková vámi procestovaná vzdálenost byla co nejmenší. První den ráno se nacházíte ve městě číslo 1. Při přejíždění z města do města je možné projíždět přes libovolná jiná města. Z každého města se lze dostat do každého a přeprava se stihne vždy za jeden den. Nezáleží na tom, ve kterém městě zůstanete po shlédnutí posledního filmu.

Napište program, který pro každé město přečte číslo promítaného filmu, dále načte popis jednotlivých cest a rozpis filmů, které chcete v jednotlivých dnech vidět, a vypíše, ve kterém městě máte jít který večer do kina, aby vámi procestovaná dráha byla minimální. Dále vypíše délku této minimální trasy.

Vstupní soubor: Vstupní soubor kina.in obsahuje na prvním řádku tři čísla n, m, k . Druhý řádek obsahuje n čísel $f_1, f_2, \dots, f_n$. Následuje m řádků popisujících cesty, každý z nich obsahuje trojici čísel z_c, k_c, l_c . Poslední řádek obsahuje k čísel $p_1, p_2, \dots, p_k$. Každá dvojice po sobě následujících čísel v řádku je oddělena jednou mezerou. Můžete předpokládat, že každý film p_i ($1 \leq i \leq k$), který chcete vidět, se promítá alespoň v jednom městě.

Výstupní soubor: Výstupní soubor kina.out obsahuje na prvním řádku délku celkově procestované dráhy mezi městy. Druhý řádek popisuje jednu optimální trasu. Obsahuje k čísel (každá dvě po sobě jdoucí čísla jsou oddělena mezerou), i -té číslo v pořadí označuje město, v němž byste podle zvolené trasy měli vidět film p_i .

Poznámka. Jestliže výstupní soubor obsahuje správnou minimální délku, ale buď neobsahuje žádný popis optimální trasy, nebo obsahuje popis trasy, který je chybný, bude řešení hodnoceno jako částečně správné a získá pro daný vstup poloviční počet bodů.

Příklad:

kina.in	6 7 7
	2 1 2 3 1 4
	1 2 13
	2 3 7
	3 4 5
	4 1 4
	1 5 8
	5 3 10
	2 6 0
	1 2 1 4 3 2 1
kina.out	49
	5 3 2 6 4 3 2

P – I – 1

Pro výrobní operaci i označíme jako h_i nejkratší čas od spuštění výroby, v němž může být operace i dokončena. Protože v továrně lze vykonávat najednou libovolný počet výrobních operací, můžeme výrobu uspořádat tak, že každá operace i bude skutečně dokončena v čase h_i . Na zhotovení celého výrobku potřebujeme dokončit všechny stanovené výrobní operace, takže výrobek je možné dokončit nejdříve v čase $\max\{h_1, h_2, \dots, h_n\}$.

Zbývá nám spočítat čas h_i . Jestliže výrobní operace i nemá žádné předcházející operace, můžeme ji okamžitě spustit. Hodnota h_i je v takovém případě přímo rovna délce trvání operace i . Pokud operace i má nějaké předcházející operace $p_1, p_2, \dots, p_k$, je třeba vykonat nejprve tyto předcházející operace. Výrobní operace i může začít nejdříve v čase, který je maximem z časů $h_{p_1}, h_{p_2}, \dots, h_{p_k}$. Když víme, kdy operace i začne a známe délku jejího trvání, umíme triviálně spočítat, kdy skončí.

Hodnoty h_i budeme počítat rekurzivně. Program obsahuje rekurzivní funkci `urci_cas(i)`, která spočítá hodnotu h_i . Pokud operace i nemá předchůdce, bude to jednoduše čas jejího trvání. Jestliže operace i má nějaké předchůdce, spustí se funkce `urci_cas` pro každého předchůdce operace i , z takto získaných časů se vezme maximum a k němu se připočítá čas trvání operace i . Navíc, jakmile vypočítáme hodnotu h_i pro nějaké i , uložíme si ji do pomocného pole. Když později zavoláme funkci `urci_cas` pro stejné i znovu, tato funkce už nebude znovu počítat, ale jednoduše vrátí už vypočítanou hodnotu z pole.

Nejprve dokážeme, že popsaný program vždy skončí. Předpokládejme, že bychom počítali nějakou hodnotu h_i a při jejím výpočtu bychom potřebovali vypočítat nějakou hodnotu h_{j_1} pro předcházející operaci j_1 . Při výpočtu h_{j_1} bychom potřebovali vypočítat nějakou hodnotu h_{j_2} a tak bychom se vnořovali hlouběji a hlouběji, až bychom zjistili, že k výpočtu nějakého h_{j_k} potřebujeme znát h_i . To by způsobilo nekonečné volání rekurze. Takovýto případ však může nastat jedině tehdy, jestliže není možné vykonat operaci i , protože operace i se může provést až po skončení j_1 a j_1 lze provést až po skončení j_2 atd., a j_k je možné vykonat až po skončení i . Dostali jsme tedy cyklus, takže i není možné provést vůbec. V zadání je však uvedeno, že všechny operace lze provést, takže tento případ nemůže nastat a náš algoritmus vždy skončí.

Na závěr zhodnotíme výpočtovou složitost algoritmu. Předpokládejme, že pro každou operaci máme seznam předcházejících operací uložen ve spojovém seznamu. Označme n počet výrobních operací a m celkový počet předcházejících operací pro všechny operace v technologickém plánu výrobku. V první části algoritmu načítáme a ukládáme potřebné údaje. Tato část má časovou složitost $O(m + n)$.

V druhé části algoritmu zavoláme funkci *urci_cas*(i) pro každé i od 1 do n . V rámci každého takového volání se funkce může ještě dále rekurzivně volat. Funkci *urci_cas*(i) voláme pro každé i jednou z hlavní části algoritmu a pro každou operaci, pro níž je operace i předchůdcem, tuto funkci opět zavoláme. Celkem se tedy vykoná $m + n$ volání funkce *urci_cas*. Pro každou hodnotu i se ale hodnota *urci_cas*(i) počítá pouze jednu a při dalších voláních se jen najde výsledek v poli v konstantním čase. Máme tedy přesně m volání funkce *urci_cas*, která proběhnou v konstantním čase. Zbývajících n volání počítá hodnotu. Při volání, které počítá hodnotu, se projde seznam všech předcházejících operací a hledá se maximum z časů. Délka takového volání je tedy úměrná počtu předcházejících operací (pokud zanedbáme čas potřebný na vnořená rekurzivní volání). Celkový čas všech n volání, která počítají hodnotu, bude tedy $O(m + n)$. Všech $m + n$ volání bude také trvat dobu $O(m + n)$. Celková časová složitost algoritmu je proto $O(m + n)$. Paměťová složitost je rovněž $O(m + n)$.

Dvě poznámky na závěr. Pokud bychom vypočítané hodnoty ve funkci *urci_cas* neukládali do pole, ale počítali bychom je vždy znovu, dostaneme algoritmus s exponenciální časovou složitostí. Všimněte si, že úlohu je možné jednoduše přetransformovat do teorie grafů — operace budou představovat vrcholy grafu a hrany (orientované) povedou vždy od operace k jejímu předchůdci. Úkolem je potom nalézt orientovanou cestu s největším součtem časů ve vrcholech. Algoritmus, který jsme uvedli, je v grafové terminologii pouze jednoduchou modifikací prohledávání do hloubky.

```
program P_I_1;
```

```
type ppostupy = ^postupy;
    postupy = record {seznam operací}
        postup: integer;
        dalsi: ppostupy;
    end;
```

```

var predpostupy:array[1..100] of ppostupy;
    {seznam předcházejících operací pro každou výrobní operaci}
cas,hotovo:array[1..100] of integer;
    {cas[i] je čas, kolik trvá operace i,
     hotovo[i] je nejnižší čas, kdy může být operace i hotova}
n:integer;          {počet operací}

procedure nacti;
var t:text;
    i,j:integer;
    novy:ppostupy;
begin
    assign(t,'TOVARNA.IN');
    reset(t);
    readln(t,n);          {načtení počtu operací}
    for i:=1 to n do begin
        read(t,cas[i]);    {načtení času jednotlivých operací}
        predpostupy[i]:=nil; {seznam předcházejících operací}
        read(t,j);
        while j<>-1 do begin
            new(novy);      {vložit operaci j do seznamu}
            novy^.postup:=j;
            novy^.dalsi:=predpostupy[i];
            predpostupy[i]:=novy;
            read(t,j);
        end;
    end;
    close(t);
end;

function urci_cas(postup:integer):integer;
{vrátí nejmenší čas, v němž může být operace hotova}
var max,kdy:integer;
    pom:ppostupy;
begin
    if hotovo[postup]=-1 then begin
        {pokud jsme čas ještě nepočítali, vypočítáme ho
         a uložíme do pole „{hotovo}, jinak neděláme nic}
        max:=0;
        pom:=predpostupy[postup];
        while pom<>nil do begin
            kdy:=urci_cas(pom^.postup);
            if kdy>max then max:=kdy;
            pom:=pom^.dalsi;
        end;
    end;

```

```

    {max je čas, kdy jsou hotové všechny předcházející operace}
    hotovo[postup] := cas[postup] + max;
end;
urci_cas := hotovo[postup];
end;

procedure vypocitej;
var i, max, kdy: integer;
    t: text;
begin
    for i:=1 to n do          {inicializace}
        hotovo[i] := -1;
    max:=0;                    {zjistíme, kdy skončí poslední operace}
    for i:=1 to n do begin
        kdy:=urci_cas(i);
        if kdy>max then max:=kdy;
    end;
    assign(t, 'TOVARNA.OUT');
    rewrite(t);
    writeln(t, max);
    close(t);
end;

begin
    nacti;
    vypocitej;
end.

```

P – I – 2

Je známo více algoritmů na konstrukci řetězce obsahujícího všechny permutace znaků dané abecedy jako podřetězce. Nejprve uvedeme příklad jednoduchého algoritmu, který pro N -prvkovou abecedu sestrojí řetězec délky $N^2 - N + 1$. Potom ukážeme komplikovanější algoritmus, který sestrojí řetězec délky pouze $N^2 - 2N + 4$ (pro $N \geq 3$). Není známo, zda existuje i nějaký kratší řetězec (existuje dolní odhad počtu znaků takového řetězce, tento odhad je však nižší než $N^2 - 2N + 4$).

Nejprve si ukážeme jednodušší konstrukci řetězce. Mějme N -prvkovou abecedu $\{a_1, a_2, \dots, a_N\}$. Řetězec bude tvořen $N - 1$ úseky tvaru $a_1 \times a_2 \dots a_N$ a ukončen bude znakem a_1 . Například pro abecedu $\{a, b, c\}$ sestrojíme řetězec $abcabca$. Tento řetězec obsahuje celkem $(N - 1) \cdot N + 1 = N^2 - N + 1$ znaků. Pro $N = 15$ tedy dostáváme řetězec délky 211.

Zbývá dokázat, že takto sestrojený řetězec skutečně obsahuje každou permutaci znaků abecedy jako vybraný podřetězec. Rozdělíme si tedy řetězec na $N - 1$ úseků tvaru $a_2a_3 \dots a_N$. Mezi každými dvěma těmito úseky, stejně jako za posledním a před prvním z nich, se nachází znak a_1 . Vezmeme nyní libovolnou permutaci znaků abecedy. Nejprve z této permutace vynecháme znak a_1 . Zbytek permutace je jistě vybraným podřetězcem našeho řetězce, neboť z každého úseku tvaru $a_2a_3 \dots a_N$ můžeme vybrat právě jeden znak tak, abychom z j -tého úseku vybrali j -tý znak permutace různý od a_1 .

Nechť je nyní znak a_1 umístěn v permutaci na i -tém místě, tzn. před ním leží $i - 1$ znaků. Potom z řetězce vybereme v pořadí i -tý výskyt znaku a_1 . Před ním se nachází $i - 1$ úseků tvaru $a_2a_3 \dots a_N$, a tedy a_1 bude i ve vybraném podřetězci na i -tém místě. Dokázali jsme, že pro libovolnou permutaci znaků abecedy můžeme nalézt takový vybraný podřetězec našeho řetězce, který se této permutaci rovná.

Nyní si popíšeme jinou, složitější konstrukci. Budeme vytvářet posloupnost řetězců $T(1), T(2), T(3), \dots$, přičemž $T(N)$ obsahuje všech $N!$ permutací N znaků jako podřetězce. Ukážeme si způsob, jak z $T(N)$ sestrojit $T(N + 1)$ (pro $N \geq 3$).

Řetězce $T(1), T(2)$ a $T(3)$ zvolíme pevně: $T(1) = a_1$, $T(2) = a_1a_2a_1$ a $T(3) = a_1a_3a_2a_1a_3a_1a_2$. Jsou to nejkratší možné řetězce pro $N = 1, 2, 3$ (to lze snadno dokázat ověřením všech možností).

Mějme nyní abecedu $\{a_1, a_2, \dots, a_N\}$. K řetězci $T(N)$ vytvoříme posloupnost *základních bodů*. První základní bod řetězce $T(N)$ je index prvního výskytu znaku a_1 v $T(N)$. Pro $i > 1$ je i -tým základním bodem index prvního výskytu znaku a_i za $(i - 1)$ -ním základním bodem v řetězci. Například v $T(3) = a_1a_3a_2a_1a_3a_1a_2$ je posloupnost základních bodů $(1, 3, 5)$.

Algoritmus pro konstrukci řetězce $T(N)$ z řetězce $T(N - 1)$:

1. Pro každé $i = 2, 3, \dots, N - 1$ vsunout těsně před i -tý základní bod řetězce $T(N - 1)$ znak a_N .
2. Na konec takto vzniklého řetězce připojit úsek

$$a_2, a_3, \dots, a_{N-3}, a_N, a_1, a_{N-1}.$$

Podle prvního bodu algoritmu jsme vložili $N - 2$ znaků a podle druhého bodu jsme vložili $N - 1$ znaků. Celkově se tedy řetězec prodloužil o $2N - 3$ znaků. Všechny základní body řetězce $T(N - 1)$ budou i základními body řetězce $T(N)$ (jen se posunou kvůli vsouvání znaků a_N

podle bodu 1) a N -tým základním bodem se stane znak a_N přidáný v bodě 2. Například pro $N = 4$ dostáváme pomocí tohoto algoritmu $T(4) = a_1a_3a_4a_2a_1a_4a_3a_1a_2a_4a_1a_3$.

Řetězec $T(N)$ sestrojený naším algoritmem (pro $N \geq 3$) má délku $N^2 - 2N + 4$. Toto tvrzení dokážeme matematickou indukcí. Délka řetězce $T(3)$ je 7, což je rovno $3^2 - 2 \cdot 3 + 4$. Předpokládejme nyní, že délka $T(N-1)$ je $(N-1)^2 - 2(N-1) + 4$ a dokážeme, že potom délka řetězce $T(N)$ je $N^2 - 2N + 4$. Řetězec $T(N)$ vznikl přidáním $2N - 3$ znaků do řetězce $T(N-1)$, jeho délka je $(N-1)^2 - 2(N-1) + 4 + 2N - 3 = N^2 - 2N + 4$. Dokázali jsme tedy, že délka řetězce $T(N)$ je skutečně $N^2 - 2N + 4$.

Nakonec je třeba dokázat, že řetězec $T(N)$ obsahuje všech $N!$ permutací znaků abecedy jako podřetězce. Tento důkaz je poměrně zdlouhavý, takže uvedeme jen základní myšlenku. Řetězec $T(N)$ rozdělíme na úseky. První úsek začíná prvním základním bodem a končí druhým základním bodem. Pro $2 \leq i \leq N-1$ bude i -tý úsek začínat znakem a_1 , který je hned za i -tým základním bodem, a končit znakem a_i , který se vyskytuje poprvé za $(i+1)$ -ním základním bodem.

Matematickou indukcí (vzhledem k N) lze dokázat, že každý z takto vytvořených úseků obsahuje znaky $a_1, a_2, \dots, a_N$ a začíná znakem a_1 . Řetězec obsahuje $N-1$ takovýchto úseků a každý z nich obsahuje právě jeden základní bod s výjimkou prvního úseku, který obsahuje dva základní body. Další vlastností dvou sousedních úseků i -tého a $(i+1)$ -ního je jejich vzájemné překrytí ve dvou znacích, a to a_1 a a_i .

Nechť má naše permutace znak a_1 na i -tém místě. Uvažujme nejprve případ že $i \leq N-1$. V permutaci vybereme první výskyt znaku a_1 z i -tého úseku řetězce $T(N)$. Z i -tého úseku je možné použít ještě jeden znak, neboť se v něm vyskytují všechny znaky, a to až za vybraným znakem a_1 . Před i -tým úsekem je $i-1$ úseků a za ním je $N-i-1$ úseků. Při výběru jednoho znaku z každého úseku tedy dostaneme všechny permutace se znakem a_1 v i -té pozici. Překrytí sousedních úseků nezpůsobí problémy, protože při výběru znaku a_1 z i -tého úseku je možné všechny ostatní znaky a_1 ignorovat a další společné znaky se stávají hraničními prvky úseků. Hraniční prvek je považován za prvek patřící jen do jednoho úseku.

Zbývá dokázat tvrzení v případě, že a_1 je v permutaci na N -tém místě. V tomto případě řetězec $T(N)$ rozdělíme na úseky se základními body jako hraničními body úseků. Počet úseků je $N-1$ a každý z nich obsahuje znaky $a_1, a_2, \dots, a_N$. Jestliže na poslední místo vybereme poslední výskyt znaku a_1 v řetězci $T(N)$, tento znak nepatří do žádného z výše uvedených

úseků. Řetězec $T(N)$ tedy obsahuje všechny permutace se znakem a_1 na posledním místě.

Uvedený algoritmus pro $N = 15$ dává následující řetězec délky 199:
acdefghijklmnobadefghijklmnoacabefghijklmnodacbfghijklmnoead
bcghijklmnofaebcdhijklmnogafbcdeijklmnohagbcdefjklmnoiahbcd
efgklmnoljaibcdefghlmnokajbcdefghimnolakbcdefghijnomalbcdefg
hijkonambcdefghijklloan

Na závěr uvádíme program, který řetězec $T(15)$ generuje výše popsá-
ným způsobem.

```
program P_I_2;
const N=15;                                {počet prvků abecedy}
      abeceda:string='abcdefghijklmno';    {abeceda řetězce}
var T: string;                              {hledaný řetězec}
    BB: array[1..N] of integer;            {indexy základních bodů}
    i: integer;
    ff: text;

function Dalsi(n: integer; T:string):string;
{funkce dostane řetězec všech permutací prvních n-1 prvků
 a vrátí řetězec všech permutací prvních n prvků (n>3)}
var i, j: integer;
    TN: string;
begin
    TN:='';
    i:=BB[1];
    j:=2;

    {přidání nových prvků před základní body}
    while j<n do begin
        TN:=TN+copy(T,i,BB[j]-BB[j-1]);
        i:=BB[j];
        j:=j+1;
        TN:=TN+abeceda[n];
    end;
    TN:=TN+copy(T,BB[n-1],length(T));

    {přidání prvků na konec řetězce}
    for i:=2 to n-3 do TN:=TN+abeceda[i];
    TN:=TN+abeceda[n];                    {nový základní bod}
    TN:=TN+abeceda[1]+abeceda[n-1];

    {úprava základních bodů}
```

```

BB[n]:=length(TN)-2;           {nový základní bod}
for i:=2 to n-1 do             {původní body se posunuly}
    BB[i]:=BB[i]+i-1;

Dalsi:=TN;
end; {Dalsi}

procedure VytvorT3;
{vytvoří řetězec všech permutací pro tříprvkovou abecedu}
begin
    T:=abeceda[1]+abeceda[3]+abeceda[2]+abeceda[1]
        +abeceda[3]+abeceda[1]+abeceda[2];
    BB[1]:=1; BB[2]:=3; BB[3]:=5;
end;

begin
    assign(ff,'P-1-2.TXT');
    rewrite(ff);
    VytvorT3;
    for i:=4 to N do
        T:=Dalsi(i,T);
        writeln(ff,T);
    close(ff);
end.

```

P – I – 3

Tuto úlohu si můžeme představit tak, že máme dáno pole n navzájem různých čísel $a[1], a[2], \dots, a[n]$ (preference politických stran) a máme za úkol nalézt nejmenší a největší prvek tohoto pole, přičemž ale k poli můžeme přistupovat pouze prostřednictvím funkce $pruzkum(i, j)$, která nám říká, zda je $a[i] < a[j]$, nebo $a[j] < a[i]$.

Ukážeme si nejprve řešení pro sudé n . Rozdělíme všechny prvky pole do dvojic a v každé dvojici prvky porovnáme (pomocí funkce $pruzkum$). Prvky se nám rozdělí na dvě podmnožiny — do množiny X dáme ty, které byly při porovnávání v dvojici větší, a do množiny Y ty, které byly při porovnávání menší. Je zřejmé, že žádný prvek z množiny X nebude nejmenším prvkem v poli, neboť existuje aspoň jeden menší prvek (ten, který s ním byl ve dvojici). Proto při hledání minima stačí hledat mezi prvky množiny Y . Podobně žádný prvek z Y nebude největším prvkem pole, a proto stačí maximum hledat mezi prvky množiny X .

Nejmenší z prvků množiny Y najdeme jednoduše. V proměnné $\min$ si budeme pamatovat nejmenší dosud nalezený prvek. Na začátku to bude libovolný prvek množiny Y . Potom budeme nejmenší dosud nalezený prvek porovnávat vždy s dalším a dalším prvkem množiny Y . Pokaždé, když bude porovnávaný prvek menší než prvek uložený v proměnné $\min$, stane se on dosud nejmenším nalezeným prvkem (uloží se do proměnné $\min$). Podobně budeme hledat i největší prvek v množině X .

V případě, že n je liché, budeme postupovat stejným způsobem, jenom do dvojic rozdělíme jen prvních $n - 1$ prvků a poslední prvek nakonec přidáme do množiny X i do množiny Y . V některých případech můžeme ještě jedno porovnávání ušetřit — pokud se tento poslední prvek stane nejmenším prvkem množiny Y (a tudíž i nejmenším prvkem celého pole), pak ho můžeme z množiny X vynechat, neboť jistě nebude současně největším prvkem. Příklad $n = 1$ ošetříme zvlášť. V tomto případě není třeba nic porovnávat — jediná strana má současně nejvyšší i nejnižší preference.

Nyní spočítáme, kolik porovnání v nejhorším případě vykonáme pro dané n . Je-li n sudé, vznikne nám $\frac{1}{2}n$ dvojic. Pro každou dvojici provedeme jedno porovnání, abychom zjistili, který prvek je větší. Množiny X a Y budou mít každá $\frac{1}{2}n$ prvků. Pro nalezení minima (nebo maxima) v množině s x prvky použijeme $x - 1$ porovnání (do proměnné $\min$ uložíme nejprve první prvek, potom tuto proměnnou postupně porovnáme se všemi ostatními $x - 1$ prvky). K nalezení minima v množině Y tedy potřebujeme $\frac{1}{2}n - 1$ porovnání a k nalezení maxima v množině X dalších $\frac{1}{2}n - 1$ porovnání. Celkový počet porovnání tedy bude

$$\frac{n}{2} + \left(\frac{n}{2} - 1\right) + \left(\frac{n}{2} - 1\right) = \frac{3n - 4}{2}.$$

Pro liché n budeme mít $\frac{1}{2}(n - 1)$ dvojic, takže na začátku vykonáme $\frac{1}{2}(n - 1)$ porovnání ve dvojicích. Množiny X a Y však budou mít $\frac{1}{2}(n - 1) + 1$ prvků, takže k nalezení minima nebo maxima z takovéto množiny potřebujeme $\frac{1}{2}(n - 1)$ porovnání. Celkový počet provedených porovnání proto bude

$$\frac{n - 1}{2} + \frac{n - 1}{2} + \frac{n - 1}{2} = \frac{3n - 3}{2}.$$

Došli jsme k závěru, že pro n sudé vykonáme nejvýše $\frac{1}{2}(3n - 4)$ porovnání a pro n liché nejvýše $\frac{1}{2}(3n - 3)$ porovnání. Tento výsledek lze zapsat i jedním vzorcem s použitím horní celé části a dolní celé části

takto:

$$2 \left\lceil \frac{n}{2} \right\rceil + \left\lfloor \frac{n}{2} \right\rfloor - 2.$$

Na závěr pár slov o implementaci popsaného algoritmu. Množiny X a Y nebudeme vytvářet na začátku výpočtu celé, ale průběžně. Nejprve vezmeme první dva prvky v poli, porovnáme je a menší z nich se stane počáteční hodnotou proměnné min , zatímco větší bude počáteční hodnotou proměnné max . Potom budeme brát vždy další a další dvojici, porovnáme vždy její prvky navzájem, následně menší z nich porovnáme s proměnnou min (a pokud bude třeba, tak obsah min změníme) a větší porovnáme s proměnnou max . Pro lichá n je třeba nakonec zvlášť zpracovat poslední prvek. Takto naprogramovaný algoritmus bude mít lineární časovou složitost (tzn. $O(n)$) a konstantní paměťovou složitost.

```
program P_I_3;
var n,i,min,max,a,b,pom:integer;

procedure pruzkum(i,j:integer; var min, max:integer);
{provede průzkum pro i a j
stranu s menšími preferencemi vrátí v min, s většími v max}
begin
  writeln('Proved průzkum ',i,',',j);
  write('> ');
  readln(max);
  if max=i then min:=j
    else min:=i;
end;

begin
  writeln('Zadej počet stran:');
  write('> '); readln(n);
  if n=1 then begin {speciální případ}
    min:=1; max:=1;
  end
  else begin {inicializace - porovnáme první pár}
    pruzkum(1,2,min,max);
  end;
  i:=3;
  while i+1<=n do begin {ostatní páry}
    pruzkum(i,i+1,a,b); {porovnáme navzájem}
    pruzkum(min,a,min,pom); {menší porovnáme s minimem}
    pruzkum(max,b,pom,max); {větší s maximem}
    i:=i+2;
```

```

end;
if i=n then begin           {poslední prvek pro liché n}
    pruzkum(min,n,min,pom); {porovnáme s min i s max,
                             pokud je třeba}

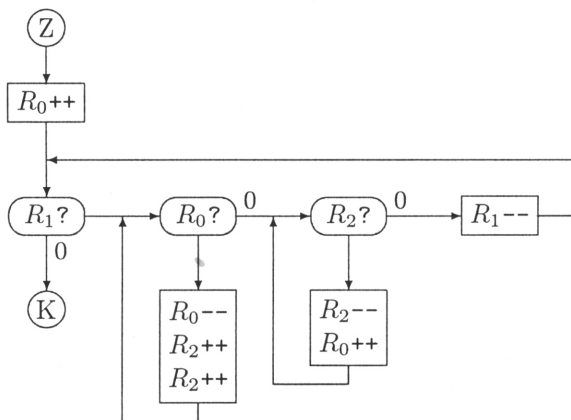
    if min<>n then
        pruzkum(max,n,pom,max);
end;
writeln('Nejvyšší preference má strana ',max,', nejnižší ',min);
end.

```

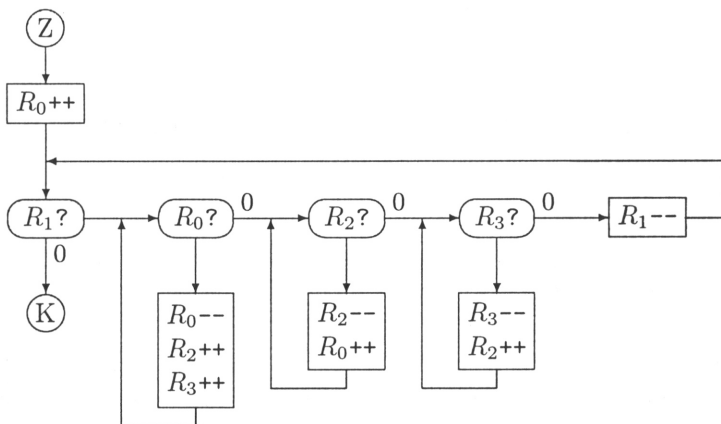
P – I – 4

a) Úkolem bylo sestavit stroj, který pro vstup n vypočítá číslo 2^n . Číslo n máme uloženo v registru R_1 . Na začátku výpočtu uložíme do R_0 číslo 1 (tj. 2^0). Potom budeme postupovat tak, že v každém kroku snížíme R_1 o 1 a registr R_0 vynásobíme 2. To provádíme tak dlouho, dokud v R_1 není 0. Tehdy máme v R_0 uloženo hledané číslo 2^n .

Zbývá už jen popsat, jak násobíme registr R_0 dvěma. Jednou možností by bylo uložit do některého dalšího registru hodnotu 2 a potom použít algoritmus násobení uvedený v příkladu ve studijním textu. My však využijeme trochu zjednodušený algoritmus, který lze použít pouze pro násobení konstantou. V cyklu budeme po jedné snižovat hodnotu R_0 , dokud neklesne na nulu, a při každém snížení R_0 dvakrát zvýšíme o 1 hodnotu registru R_2 . Po skončení cyklu máme v R_2 dvojnásobek hodnoty, kterou jsme měli původně v R_0 . Zbývá už jen přesunout hodnotu z R_2 zpět do R_0 , což provedeme dalším cyklem, který vždy jednou sníží R_2 a zvýší R_0 , dokud nebude v R_2 nula.



b) Úkolem je vypočítat n -té Fibonacciho číslo F_n . Podle definice $F_0 = F_1 = 1$ a $F_n = F_{n-1} + F_{n-2}$ pro $n \geq 2$. Jestliže si však zavedeme pomocný člen této posloupnosti $F_{-1} = 0$, bude platit, že $F_n = F_{n-1} + F_{n-2}$ i pro $n = 1$, což sníží počet případů, které je třeba v programu pro Minského stroj speciálně ošetřit.



Náš stroj bude pracovat tak, že v cyklu bude snižovat registr R_1 a počítat další člen Fibonacciho posloupnosti, dokud nebude v R_1 nula. Po k průchodech tohoto cyklu bude registr R_0 obsahovat F_k a registr R_2 bude obsahovat F_{k-1} . Celkem se provede n průchodů, takže stroj správně vypočítá hodnotu F_n .

Před prvním prováděním cyklu (tj. po nula průchodech) potřebujeme mít v registru R_0 číslo $F_0 = 1$ a v registru R_2 číslo $F_{-1} = 0$. Toho dosáhneme snadno příkazem R_0++ .

Zbývá nám prozkoumat, co potřebujeme provádět v těle cyklu. V registru R_0 máme číslo F_k a v registru R_2 číslo F_{k-1} . Chceme dosáhnout toho, aby v registru R_0 bylo $F_{k+1} = F_k + F_{k-1}$, tj. součet registrů R_0 a R_2 , a v registru R_2 aby bylo F_k , tj. číslo, které bylo původně v R_0 . Budeme postupovat tak, že hodnotu v R_0 přičítáme k registrům R_2 a R_3 (a to tak, že budeme v cyklu R_0 snižovat, dokud neklesne na nulu, a pokaždé při tom zvýšíme R_2 a R_3). Po skončení této operace máme v R_0 nulu, v R_2 máme F_{k+1} a v R_3 máme F_k . Potřebujeme nyní ještě obsahy registrů přemístit tak, abychom do R_0 dostali obsah R_2 a do R_2 obsah R_3 . To provedeme opět obvyklým způsobem (dvěma „přesýpacími“ cykly).

Jednotlivé počítače nazvime vrcholmi. Z vrchola i do vrchola j nech vedie hrana práve vtedy, keď je počítač i pripojený k počítaču j . Vznikne nám tak orientovaný graf G . Našou úlohou je vybrať množinu vrcholov M s minimálnym počtom prvkov takú, že pre ľubovoľný vrchol v existuje vrchol $u \in M$ taký, že z u do v vedie cesta (pripúšťame cestu dĺžky 0, t.j. $u = v$).

Množinu vrcholov U , pre ktorú platí, že

1. pre ľubovoľné dva vrcholy $u, v \in U, u \neq v$ existuje cesta z u do v vedúca len cez vrcholy patriace do U
2. zo žiadneho vrchola $w \in G$ nepatriaceho do U nevedie hrana do žiadneho vrchola $u \in U$

nazvime *maximálny silne súvislý komponent* (MSSK) grafu G (poznajme, že vrchol, do ktorého nevedie žiadna hrana tvorí maximálny silne súvislý komponent). Každé dva rôzne MSSK sú disjunktné. Keby MSSK U a MSSK V ($U \neq V$) neboli disjunktné, potom existuje u , ktorý patrí do oboch a vrchol v , ktorý patrí do U a nepatrí do V (alebo naopak). Z vrchola v vedie cesta do vrchola u . Na nej niekde musia za sebou nasledovať vrchol v' , ktorý do V nepatrí a vrchol u' , ktorý už do V patrí — množina V nemá vlastnosť 2 MSSK — spor.

Lahko vidno, že z každého MSSK grafu G musí nejaký vrchol patriť do množiny M . Zároveň stačí, aby z každého MSSK bol v množine M jeden ľubovoľný vrchol.

Budeme ofarbovať vrcholy grafu rôznymi farbami. Začnime z ľubovoľného neofarbeného vrchola v a ofarbime farbou f všetky neofarbené vrcholy (vrátane v), do ktorých existuje cesta z v vedúca cez doteraz neofarbené vrcholy. Farbenie realizujeme prehľadávaním grafu do hĺbky. Vrchol v vyhlásime za *reprezentanta* farby f . Potom si zvolíme ďalšiu farbu f a iný neofarbený vrchol v a opakujeme, kým sa neminú všetky neofarbené vrcholy. Do každého vrchola zrejme existuje cesta z niektorého z reprezentantov. Niektorí reprezentanti sú však zbytoční: ak do vrchola r_1 , reprezentujúceho farbu f_1 vedie cesta z vrchola r_2 reprezentujúceho farbu f_2 , z r_2 sa dá dostať do všetkých vrcholov ofarbených farbou f_1 , a preto r_1 je zbytočný.

Ako rýchlo zistiť, ktorí reprezentanti sú zbytoční? Nech r_i je zbytočný reprezentant. Potom existuje reprezentant r_j taký, že z r_j vedie do r_i cesta prechádzajúca len cez vrcholy ofarbené farbami f_j a f_i , pričom farby na ceste sa nestriedajú, t.j. cesta vedie najprv cez vrcholy farby

f_j a potom cez vrcholy farby f_i . Posledný vrchol na tejto ceste ofarbený farbou f_j označme v , prvý vrchol ofarbený f_i nech je u . Vrcholy farby f_j boli zrejme ofarbované neskôr ako vrcholy farby f_i . Preto ak by sme v okamihu ofarbovania vrchola v vedeli, že z vrchola u sa dá dostať do vrchola r_i , veľmi ľahko by sme prišli na zbytočnosť r_i . Preto si pre každú farbu f_i a pre každý vrchol u ofarbený touto farbou spočítame, či sa z vrchola u dá dostať do reprezentanta farby f_i t.j. vrchola r_i . Táto práca sa dá tiež zveriť rekurzívnej ofarbovacej procedúre. Na záver treba skontrolovať, pre každú hranu, ktorá vedie medzi vrcholmi rôznych farieb, či sa z jej koncového vrchola dá dostať do reprezentanta farby koncového vrchola. Ak áno, označíme reprezentanta tejto farby ako zbytočného. Táto kontrola sa dá tiež robiť počas ofarbovania.

Procedúra **Prehladaj**(v : integer) dostane ako argument číslo vrchola, z ktorého má prehľadávať. Označme $N(v)$ množinu vrcholov, do ktorých vedie z v hrana. Pred spustením procedúry **Prehladaj** z vrchola r_i si poznačíme, že z vrchola r_i sa dá dostať do reprezentanta r_i . Táto procedúra

- ▷ ofarbí vrchol v aktuálnou farbou f_i ;
- ▷ rekurzívne sa zavolá pre všetky vrcholy $u \in N(v)$, ktoré ešte nie sú ofarbené;
- ▷ ak existuje vrchol $u \in N(v)$ farby f_i , z ktorého sa dá dostať do reprezentanta r_i , poznačíme si, že aj z v sa dá dostať do r_i ;
- ▷ ak existuje vrchol $u \in N(v)$ ofarbený farbou $f_j \neq f_i$ a pritom z u existuje cesta do reprezentanta r_j , poznačíme si, že reprezentant r_j je zbytočný.

Ostáva ukázať, že množina všetkých reprezentantov, ktorí nie sú zbytoční, tvorí našu hľadanú množinu M . Ľahko vidno, že do každého zbytočného reprezentanta r vedie cesta z nejakého reprezentanta, ktorý nie je zbytočný. Nech to tak nie je. Potom existuje reprezentant r_1 taký, že z r_1 vedie do r cesta. Ak by aj ten bol zbytočný, existuje r_2 taký, že z neho vedie cesta do r_1 atď. Buď po nejakom čase prideme k reprezentantovi, do ktorého už nič nevedie, alebo sa nám začnú reprezentanti opakovat', teda nejakí dvaja, r_i a r_j sa v postupnosti vyskytnú aspoň dvakrát. Jeden z nich musel byť ofarbovaný skôr, nech je to r_i . Potom ale z r_i vedie do r_j cesta, a preto by mal byť r_j ofarbený farbou f_i (alebo inou, použitou skôr ako f_i) — spor. Z množiny nezbytočných reprezentantov sa teda dá dostať do ľubovoľného reprezentanta, a teda aj do ľubovoľného vrchola. Zároveň žiadni dvaja reprezentanti nemôžu ležať v rovnakom MSSK

(lebo inak by museli byť ofarbení rovnakou farbou), teda ich je naozaj minimálny možný počet.

Časová a pamäťová zložitosť: Prehľadávanie každého vrchola je úmerné počtu hrán, ktoré z neho vedú. Žiadny vrchol sa neprehľadáva viac než raz, preto celková časová zložitosť algoritmu je $O(M + N)$, kde M je celkový počet hrán v grafe. Pamäťová zložitosť je tiež $O(M + N)$, pretože si potrebujeme zapamätať celý graf.

P – II – 2

Riešenie tohoto príkladu používa metódu dynamického programovania. Označme $A_i = a[1], a[2], \dots, a[i]$ postupnosť utvorenú z prvých i členov postupnosti a , analogicky $B_j = b[1], \dots, b[j]$. Budeme riešiť všeobecnejšiu úlohu: Pre každé i, j , ($0 \leq i \leq M$, $0 \leq j \leq N$) vypočítame, aký je maximálny súčet vybranej podpostupnosti postupností A_i a B_j . Tieto maximálne súčty si budeme zapisovať do tabuľky $p[0..M, 0..N]$, kde $p[i, j]$ je súčet maximálnej vybranej podpostupnosti postupností A_i a B_j . Hľadaný maximálny súčet bude teda hodnota $p[M, N]$.

Tabuľku p budeme vyplňovať po riadkoch s využitím predpočítanej informácie v predošlom riadku. Riadok $p[0]$ obsahuje samé nuly, pretože neexistuje vybraná podpostupnosť prázdnej postupnosti. Riadok $p[i]$ (pre $i > 0$) vyplníme podľa riadku $p[i - 1]$ takto: Políčko $p[i, 0]$ je zrejme nula. Políčko $p[i, j]$ (pre $j > 0$) vieme vyplniť pomocou hodnôt $p[i - 1, j]$, $p[i, j - 1]$ a $p[i - 1, j - 1]$. Ak sa čísla $a[i]$ a $b[j]$ nezhodujú, každá vybraná podpostupnosť postupností A_i a B_j je zároveň vybranou podpostupnosťou postupností A_{i-1} a B_j alebo A_i a B_{j-1} . Teda v tomto prípade je $p[i, j]$ rovné maximu z čísel $p[i - 1, j]$ a $p[i, j - 1]$. Ak $a[i] = b[j]$, každá vybraná podpostupnosť postupností A_i a B_j je vybranou podpostupnosťou postupností A_{i-1} a B_j alebo A_i a B_{j-1} , alebo vybranou podpostupnosťou postupností A_{i-1} a B_{j-1} s pridaným členom $a[i] = b[j]$. Preto $p[i, j]$ je rovné maximu z čísel $p[i - 1, j]$, $p[i, j - 1]$ a $p[i - 1, j - 1] + a[i]$.

Navrhnutý algoritmus má časovú zložitosť $O(M \cdot N)$. Pamäťová zložitosť je tiež $O(M \cdot N)$. Keďže každý riadok tabuľky p závisí iba na predchádzajúcom riadku, stačí si pamätať len posledné dva riadky (pri počítaní riadku $p[i]$ si pamätáme predchádzajúci riadok $p[i - 1]$. V programe p1 značí riadok $p[i - 1]$ a p2 riadok $p[i]$.), pamäťová zložitosť je $O(M + N)$.

Úlohu budeme riešiť v dvoch prechodoch. V prvom prechode nájdeme kandidáta — prvok, ktorý ako jediný môže mať nadpolovičnú väčšinu. V druhom prechode len overíme, či sa tento prvok nachádza v poli a viac ako $\frac{1}{2}N$ -krát.

Kandidáta budeme hľadať nasledovným spôsobom: pre každý prvok k , ktorý sa v poli a aspoň raz vyskytuje, si budeme počítat jeho silu s_k . Na začiatku položíme silu všetkých prvkov rovnú 0. Silu prvkov budeme meniť takým spôsobom, aby v každom okamihu bola nenulová pre nanajvýš jeden prvok. Tento prvok nazvime kandidátom a označme ho K . Ak majú všetky prvky silu nulovú, kandidátom je buď prvok $a[1]$ (pred začiatkom výpočtu), alebo prvok, ktorý bol kandidátom v predchádzajúcom kroku.

Pri spracovávaní prvku $a[i]$ môžu nastať tieto situácie:

1. $K = a[i]$, t.j. ďalší spracovávaný hlas patrí kandidátovi. Zvýšime s_K o 1.
2. $K \neq a[i]$, $s_K > 0$. spracovávaný hlas nepatrí kandidátovi, preto znížime s_K o 1.
3. $K \neq a[i]$, $s_K = 0$. Zvýšime silu $s_{a[i]}$ prvku $a[i]$ o 1. Tým sa prvok $a[i]$ stane novým kandidátom K .

Tento postup opakujeme, kým nespracujeme všetky prvky poľa.

Je zrejmé, že si netreba pamätať silu všetkých prvkov, stačí si pamätať silu kandidáta a to, ktorý prvok je kandidátom. Na to nám stačia dve premenné typu integer.

Lemma. *Nech sa nejaký prvok K vyskytuje v poli a M -krát, kde $M > \frac{1}{2}N$. Potom po spracovaní všetkých prvkov poľa bude K kandidátom so silou $s_K \geq 2M - N > 0$.*

Označme počet zvýšení sily kandidáta K ako k_+ , počet znížení jeho sily k_- , počet zvýšení sily ľubovoľného iného kandidáta l_+ a počet znížení sily iných kandidátov l_- . Zníženie sily kandidáta K , ako aj zvýšenie sily iného kandidáta je spôsobené jedine výskytom prvku rôzneho od K . Takýchto operácií teda bude najviac $N - M$. Každý výskyt prvku K spôsobí buď zvýšenie sily K , alebo zníženie sily iného kandidáta (prvky rôzne od K si však môžu znižovať silu aj navzájom), preto týchto operácií bude najmenej M .

$$N - M \geq k_- + l_+,$$

$$k_+ + l_- \geq M,$$

$$l_+ - l_- \geq 0.$$

Posledná nerovnosť vyplýva z toho, že počet znížení u žiadneho prvku nepresiahne počet zvýšení. Po sčítaní nerovností a úprave dostaneme

$$k_+ - k_- \geq 2M - N > 0,$$

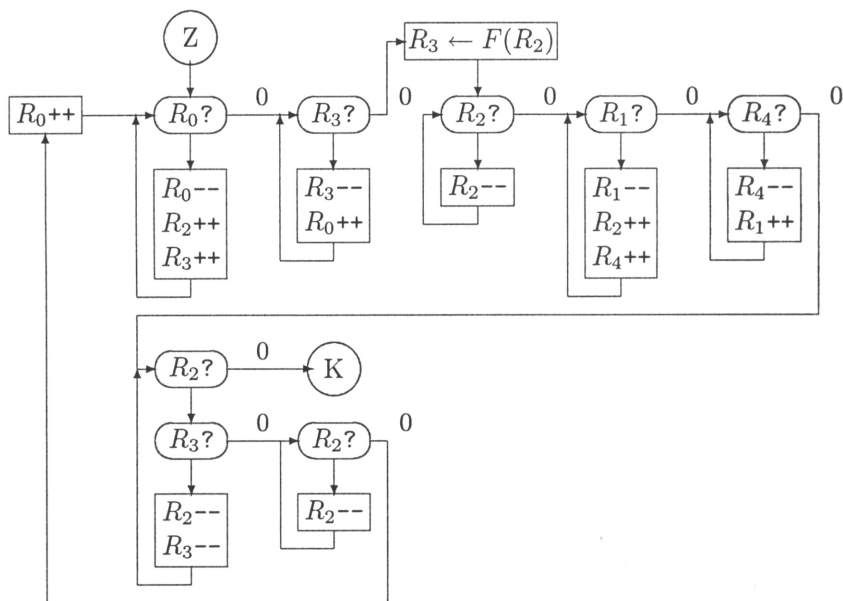
a teda po skončení algoritmu má prvok K kladnú silu. To je možné len tak, že bude na konci kandidátom.

Časová a pamäťová zložitosť: algoritmus vyžaduje dva prechody poľom, každý z nich je v čase $O(N)$. Pamäťová zložitosť je konštantná.

P – II – 4

Časť A. Stroj, riešiaci túto úlohu bude postupne skúšať ako x čísla $0, 1, 2, \dots$. Vypočíta funkčnú hodnotu $F(x)$ v každom z týchto čísel a porovná ju s hodnotou y v registri R_1 . Ak je väčšia alebo rovná, máme riešenie, ak nie, zvýšime hodnotu x o 1 a pokračujeme. V registri R_0 budeme uchovávať hodnotu x , v registri R_1 bude hodnota y . Funkčnú hodnotu $F(x)$ uložíme do registra R_3 . Registre R_2 a R_4 slúžia ako pomocné registre. Jeden cyklus stroja sa bude skladať z týchto operácií:

- ▷ skopírovanie obsahu registra R_0 do registra R_2 , ktorý bude slúžiť ako vstupný register pre výpočet funkcie F . Najprv presunieme obsah registra R_0 do registrov R_2 a R_3 (za každé zníženie registra R_0 raz zvýšime každý z registrov R_2 a R_3), potom presunieme naspäť obsah R_3 do registra R_0 .
- ▷ Použijeme inštrukciu na výpočet funkcie F . R_2 je vstupný register, výsledok sa uloží do výstupného registra R_3 . Po vykonaní tejto inštrukcie je obsah registra R_2 nedefinovaný, preto ho musíme vynulovať.
- ▷ Prekopírujeme obsah registra R_1 do registra R_2 za pomoci registra R_4 rovnakou technikou ako v prvom kroku.
- ▷ Porovnáme veľkosti čísel uložených v registroch R_2 a R_3 . Postupne budeme od oboch registrov odpočítavať jednotku, až kým sa nám jeden z registrov nevynuluje. Ak sa prvý vynuluje register R_2 (alebo sa vynulujú oba registre naraz), znamená to, že $y \leq F(x)$. Keďže sme hodnoty x skúšali od najmenšieho, je to najmenšie x s touto vlastnosťou. Hodnota x je prezieravo uložená v registri R_0 , takže môžeme skončiť. Ak sa naopak prvý vynuluje register R_3 , znamená to, že $F(x) < y$, a preto musíme pokračovať v cykle. Vynulujeme register R_2 a vrátime sa na začiatok.



Časť B. Činnosť stroja je založená na jednoduchej myšlienke: vstupné číslo n si skopírujeme do pomocného registra, vypočítame číslo, ktoré vznikne obrátením binárneho zápisu n (označme ho n_2^R). Potom porovnáme obe tieto čísla. Ak sú rovnaké, odpoveď stroja bude 1, v opačnom prípade bude odpoveď 0.

Podme sa na činnosť stroja pozrieť trochu bližšie. Najprv obsah registra R_1 , obsahujúceho vstupné číslo n prekopírujeme za pomoci registra R_4 do registra R_5 . Potom budeme v cykle v registri R_3 vyrábať číslo, ktoré vznikne otočením binárneho zápisu čísla n . Nech $n = 2^k b_k + 2^{k-1} b_{k-1} + \dots + 2^0 b_0$ je binárny zápis čísla n . Po i prechodoch cyklu ($0 \leq i \leq k+1$) bude platiť:

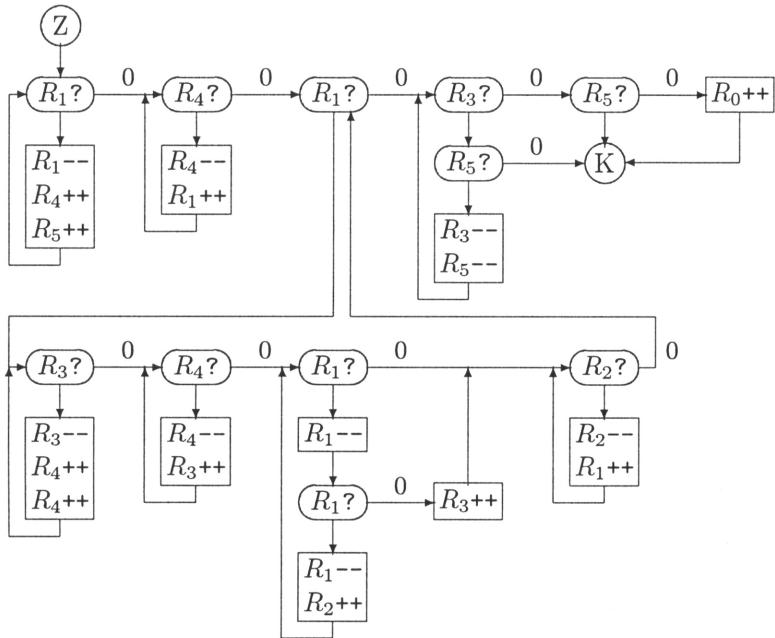
$$R_1 = 2^{k-i} b_k + 2^{k-i-1} b_{k-1} + \dots + 2^0 b_i,$$

$$R_3 = 2^{i-1} b_0 + 2^{i-1} b_1 + \dots + 2^0 b_{i-1}.$$

Na začiatku teda $R_1 = n$, $R_3 = 0$. V jednom prechode cyklom najprv vynásobíme obsah R_3 dvoma, potom vydělíme obsah R_1 dvoma. Obe tieto operácie sa dajú realizovať s jedným pomocným registrom tak, že na každé zníženie obsahu R_3 dvakrát zvýšime obsah pomocného registra, resp. na každé dve zníženia R_1 raz zvýšime obsah pomocného registra. Potom stačí len presypať obsah pomocného registra naspäť do R_3 ,

resp. R_1 . Ak nám pri delení vznikne zvyšok, vieme, že posledná cifra zápisu R_1 bola 1, a preto nastavíme poslednú cifru aj číslu v registri R_3 (t.j. pripočítame k R_3 jednotku).

Keď je v registri R_1 nula, zrejme platí $i > k$, a teda v R_3 je číslo n_2^R . Posledná vec, ktorú treba urobiť, je porovnať obsah registrov R_3 a R_5 . Budeme postupne znižovať obsah oboch registrov o 1, až kým sa jeden z nich nebude nulový. V prípade, že je aj druhý nulový, zvýšime obsah registra R_0 (lebo n je palindróm). V opačnom prípade zanecháme v registri R_0 nulu a skončíme.



P – III – 1

Vytvoríme graf G , ktorého vrcholy sú križovatky a medzi vrcholmi i a j vedie hrana práve vtedy, ak sú križovatky i a j spojené cestou. Hranu, ktorej odobratie spôsobí rozpadnutie grafu na dve časti (komponenty súvislosti), nazývame *most*. Ukážeme, že mosty sú práve tie hrany, ktoré musia zostať obojsmerné. Zrejme hranu e z vrchola u do v , ktorá je mostom, nemôžeme orientovať (ak sme ju zorientovali povedzme z u do v ,

nedalo by sa dostať z v do u , pretože e je most). Z algoritmu vyplynie, že všetky ostatné hrany môžu byť „zjednosmernené“.

Algoritmus je založený na myšlienke prehľadávania grafu do hĺbky. Prehľadávanie do hĺbky je rekurzívna procedúra s jediným parametrom — vrcholom v . Tento vrchol označíme za už prehľadaný a rekurzívne voláme tú istú procedúru pre všetkých ešte neprehľadaných susedov vrchola v . Volajme týchto susedov *potomkami* vrchola v . Nazvime *hlavnou* každú hranu, ktorá vedie z predka do niektorého z jeho potomkov, a hrany, ktoré nie sú hlavné, volajme *spätné*.

Pre naše účely priradíme navyše pri prehľadávaní každému vrcholu v poradové číslo c_v označujúce, kolkátý v celkovom poradí bol vrchol v prvýkrát objavený. Zároveň každú ešte neorientovanú (obojsmernú) hranu orientujeme („zjednosmerníme“) smerom od vrchola v . Orientáciu už orientovaných hrán nemeníme.

Ukážeme, ako sa dá tento algoritmus použiť na nájdenie mostov v grafe. Aby sme zistili, či je hrana z u do v most, potrebujeme overiť, či sa dá dostať z v do u po hranách rôznych od hrany (u, v) . Ak sa dá, hrana (u, v) mostom byť nemôže. Naopak, ak sa nedá, hrana (u, v) je most. Keďže každý most je hlavnou hranou, predpokladajme, že v je potomok u , teda prehľadávacia procedúra pre v bola spustená v prehľadávacej procedúre pre u . Nech w je vrchol s najmenším číslom c_w taký, do ktorého sa dá dostať z vrchola v po orientovaných hranách.

Ak je hrana (u, v) most, pre každý vrchol w' objavený volaním prehľadávacej procedúry z v platí $c_{w'} \geq c_v$. Zároveň žiaden z vrcholov, do ktorých sa vieme prehľadávaním z v dostať (nepoužívame hlavné, t.j. už orientované, hrany), nemohol byť v okamihu volania procedúry pre v objavený. Teda $c_w \geq c_v$.

Naopak, predpokladajme, že $c_v \leq c_w$. Označme P množinu prehľadaných vrcholov tesne pred spustením procedúry pre v a Q množinu vrcholov, ktoré objavíme touto procedúrou. Okrem hrany (u, v) nemôže viesť žiadna hlavná hrana z vrchola z P do vrchola z Q , ani naopak. (Ak by viedla z P do Q , vrchol v Q by bol už prehľadaný pred volaním procedúry, čo je spor. Ak by viedla z Q do P , táto hrana by nemohla byť hlavná, pretože vedie do už objaveného vrchola — opäť spor.) Ak $c_v \leq c_w$, neexistuje žiadna spätná hrana z Q do P (inak by platilo $c_v > c_w$), a teda neexistuje žiadna hrana medzi P a Q okrem (u, v) . Z toho vyplýva, že hrana (u, v) je most.

Zostáva nám ukázať, že hrany, ktoré nie sú mostami, sú orientované vhodne, t.j. že je možné prejsť z každého vrchola do ľubovoľného iného

vrchola dodržiujúc orientáciu hrán. Predpokladajme, že graf G nemá mosty. Nech u a v sú také vrcholy, že $c_u = 1$ a $c_v = 2$. Z predošlého vieme, že pre hlavnú hranu (u, v) , ktorá nie je mostom, platí $c_v > c_w$, teda $c_w = 1$. Teda existuje orientovaný cyklus (postupnosť vrcholov $v_1, \dots, v_k$ taká, že $v_1 = v_k$ a pre každé $i = 1, \dots, k - 1$ vedie z vrchola v_i do v_{i+1} orientovaná hrana) prechádzajúci vrcholmi u, v . Označme S množinu vrcholov z cyklu. Pre množinu S platí, že sa vieme z každého do každého z jej vrcholov dostať po orientovaných hranách. Ak S obsahuje všetky vrcholy, ukázali sme, že orientácia grafu vyhovuje. V opačnom prípade nech x je vrchol mimo S taký, že existuje vrchol y v S , že z y vedie do x orientovaná hrana. Z x sa dá dostať po orientovaných hranách do niektorého vrcholu z S , pretože inak by bola hrana (y, x) most. Takže sa dá dostať aj z x do u , aj z u do x , a teda ak vrchol x pridáme do S , zostane zachovaná vlastnosť, že z každého do každého vrchola v S sa dá dostať. Induktívne môžeme pridať do S všetky vrcholy, z čoho vyplýva, že navrhnutá orientácia G je vhodná.

Analogickú argumentáciu možno použiť aj keď sa v grafe G mosty nachádzajú. Nech G_1 je (neorientovaný) graf, ktorý vznikne z G po odobraní všetkých mostov. Označme $C_1, \dots, C_l$ komponenty súvislosti G_1 (z každého vrchola v C_i sa dá dostať do každého vrchola z C_i po hranách z G_1 , pre $i = 1, \dots, l$). Existuje aspoň jeden komponent C_i , do ktorého viedol jediný most. (Ak si zostrojíme graf, v ktorom každému komponentu zodpovedá jeden vrchol a dva vrcholy sú spojené hranou práve vtedy, keď medzi zodpovedajúcimi komponentami v G vedie most, tento graf je súvislý a neobsahuje kružnice, musí to byť teda strom. Každý strom má aspoň jeden list.) Pre tento komponent možno použiť argumenty z predchádzajúceho odstavca, C_i z grafu vynechať a induktívne pokračovať v dôkaze pre ostatné C_j .

Implementácia. Pre každý vrchol v je v poli **kriz** uložený zoznam jeho susedných vrcholov. Teda každá hrana je v tomto poli uložená dvojnásobne a tieto dve kópie na seba navzájom ukazujú pomocou smerníkov **dual**. Atribút **aktiv** hovorí, či sa dá v tom smere po danej hrane prechádzať (využíva sa pri orientovaní, keď povoľujeme len jeden z dvoch možných smerov hrany). Pole **sus[v]** uchováva počet susedov vrchola v , pole **c[v]** obsahuje c_v a v poli **naj[v]** je uložené číslo c_w . Najskôr pomocou procedúry **prehladaj** podľa vyššie uvedeného algoritmu orientujeme všetky hrany grafu, pričom mosty úplne vymažeme (obidvom hranám nastavíme **aktiv** na **false**). Potom vypíšeme všetky aktívne hrany.

Pamäťová zložitosť algoritmu je $O(M + N) = O(M)$. Časová zložitosť

je zhodná so zložitostou prehľadávania do hĺbky, teda tiež $O(M)$ (po každej hrane prejdeme práve raz).

P – III – 2

Úlohu budeme riešiť analogicky ako v krajskom kole — v prvom prechode nájdeme kandidátov a v druhom overíme pre každého z nich, či sa nachádza v poli a viac ako N/k -krát.

Kandidátov je zjavne najviac $k - 1$. Budeme ich hľadať nasledovne: pre každý prvok p , ktorý sa v poli a aspoň raz vyskytuje, si budeme počítat jeho silu s_p . Na začiatku položíme silu všetkých prvkov rovnú 0. Silu prvkov budeme meniť takým spôsobom, aby v každom okamihu bola nenulová pre nanajviš $k - 1$ prvkov. Tieto prvky nazveme kandidátmi a označme ich $K_1, \dots, K_{k-1}$.

Pri spracovávaní prvku $a[i]$ môžu nastať tieto situácie:

1. $\exists j: K_j = a[i]$, t.j. ďalší spracovávaný hlas patrí niektorému kandidátovi. Zvýšime s_{K_j} o 1.
2. $\forall j: K_j \neq a[i]$, $\forall u: s_{K_u} > 0$. Spracovávaný hlas nepatrí žiadnemu kandidátovi, preto znížime každému kandidátovi silu o 1.
3. $\forall j: K_j \neq a[i]$, $\exists u: s_{K_u} = 0$. Zvýšime silu $s_{a[i]}$ prvku $a[i]$ o 1. Preto sa prvok $a[i]$ stane novým kandidátom K_u .

Tento postup opakujeme, kým nespracujeme všetky prvky poľa.

Je zrejmé, že si netreba pamätať silu všetkých prvkov, stačí si pamätať sily $k - 1$ kandidátov a to, ktoré prvky sú kandidátmi. Na to nám stačia dve polia veľkosti $O(k)$.

Lemma. *Nech sa nejaký prvok P vyskytuje v poli a M -krát, kde $M > N/k$. Potom po spracovaní všetkých prvkov poľa bude P kandidátom so silou $s_P > 0$.*

DÔKAZ. Nazvime operácie 1 a 3 zvýšením a operáciu 2 znížením. Dokážme najskôr, že zníženie je najviac N/k . Sporom. Všimnime si, že sila každého prvku je nezáporné celé číslo, teda súčet síl všetkých prvkov je na konci určite nezáporný. Ak by bolo zníženie viac ako N/k , (t.j. aspoň $\lfloor N/k \rfloor + 1$) znamenalo by to, že sa celkový súčet síl znížil aspoň o $(\lfloor N/k \rfloor + 1) \cdot (k - 1)$, zatiaľ čo sa zvýšil najviac o $N - (\lfloor N/k \rfloor + 1)$. To ale znamená, že súčet síl prvkov na konci je

$$S \leq N - \left(\left\lfloor \frac{N}{k} \right\rfloor + 1 \right) - \left(\left\lfloor \frac{N}{k} \right\rfloor + 1 \right) (k - 1) = N - k \left(\left\lfloor \frac{N}{k} \right\rfloor + 1 \right) < 0,$$

čo je spor, preto je naozaj zníženie najviac N/k .

Všimnime si teraz prvok P . Nech jeho výskyt A -krát spôsobil zníženie, B -krát zvýšenie. Opäť sporom. Ukážeme, že $s_P > 0$. Ak by mal prvok P na konci silu 0, znamenalo by to, že bolo aspoň $A + B$ znížení — A -krát ho spôsobil prvok P , B iných znížení muselo prvku P znížiť silu na 0. Počet znížení je teda aspoň $A + B = M > N/k$, čo je spor s vyššie dokázaným tvrdením, že zvýšenie je najviac N/k . Preto má prvok P na konci nenulovú silu. To je možné len tak, že bude na konci kandidátom.

Časová a pamäťová zložitosť: algoritmus vyžaduje dva prechody poľom, každý z nich je v čase $O(kN)$. Pamäťová zložitosť je $O(k)$.

P – III – 3

Stroj riešiaci túto úlohu je už pomerne zložitý a veľmi ťažko by sa kreslil naraz, bez toho, aby sme ho rozložili na menšie celky. Predtým ako ho budeme konštruovať, si povedzme, ako by sa takýto problém riešil na normálnom počítači.

Prvé riešenie. Jeden z prístupov by bol backtrackom skúšať všetky možné podmnožiny množiny M , pre každú vypočítať súčet a overiť či sa nerovná danému číslu s . Skončili by sme, ak by sme našli podmnožinu s vyhovujúcim súčtom, alebo keby sme vyskúšali všetky podmnožiny množiny M . Klasické backtrackové riešenie však používa zásobník, ktorý by sme pomocou registrového stroja simulovali len s veľkou námahou. Pekný trik, ako vyskúšať všetky podmnožiny množiny M je takýto: Množina M má kód m . Postupne budeme skúšať všetky také množiny N , ktorých kód n je menší alebo rovný m . Pre každú takúto množinu vypočítame kód prieniku $M \cap N$, čo je vlastne logický súčin (AND) po bitoch čísel m a n . Niektoré podmnožiny síce vygenerujeme viackrát, ale to vôbec nevadí. Pre každý prienik (t.j. pre jeho kód m AND n) potom spočítame súčet jeho prvkov a skontrolujeme, či sa náhodou nerovná hľadanému súčtu s .

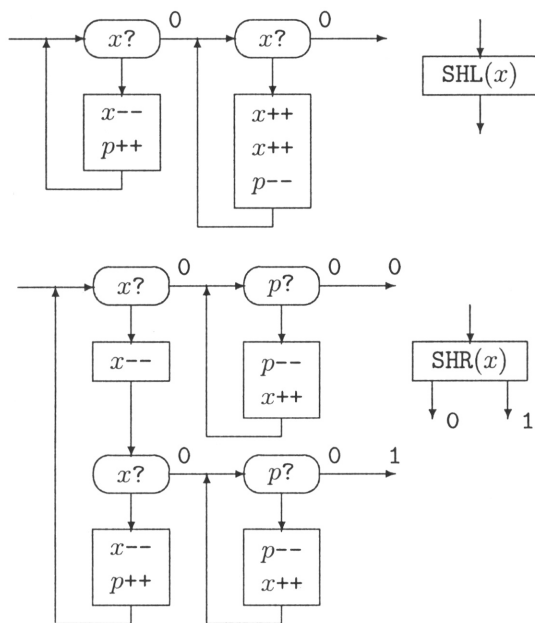
Blok pre bitový logický súčin skonštruujeme podobne ako blok pre logický súčet (OR, viď. ďalej). Na výpočet súčtu prvkov v množine môžeme použiť blok $\text{SHR}(x)$, ktorý zisťuje hodnotu najnižšieho bitu čísla v registri x a zároveň register x celočíselne vydolí dvoma (viď. ďalej) a blok $\text{ADD}(x, y)$, definovaný v príklade zo zadania.

Druhé riešenie. Ako vzorové uvádzame iné riešenie, ktoré využíva myšlienku dynamického programovania. Postupne budeme budovať množiny súčtov, ktoré sa dajú vytvoriť len z k najmenších prvkov množiny M . Označme tieto množiny $S_0, S_1, \dots, S_k$ a ich kódy $s_1, s_2, \dots, s_k$. Jediné

číslo, ktoré sa dá utvoriť súčtom nula prvkov, je číslo 0. Preto $S_0 = \{0\}$ a $s_0 = 1$. Predstavme si, že už máme vytvorenú množinu S_i , a nech $(i + 1)$ -vý najmenší prvok v množine M je p . Ku každému prvku z množiny S_i pripočítame číslo p . Dostaneme tak množinu S'_i , $S'_i = \{c + p : c \in S_i\}$. Keďže každý súčet z prvých $i + 1$ prvkov sa dá dosiahnuť buď s použitím alebo bez použitia prvku p , množina S_{i+1} dostaneme zjednotením množín S_i a S'_i . Kód s'_i množiny S'_i dostaneme veľmi jednoducho: $s'_i = s_i \cdot 2^p$. Zjednotenie množín zase dosiahneme bitovým logickým súčtom ich kódov, t.j. $s_{i+1} = s_i \text{OR} s'_i$. Pred tým, ako do detailov rozoberieme činnosť nášho stroja, si definujeme a popíšeme niekoľko blokov.

Všetky popisované bloky pre správnu činnosť predpokladajú, že všetky použité pomocné registre sú pred vstupom do bloku vynulované. Pred výstupom z bloku sa tieto registre opäť vynulujú.

Popis blokov.

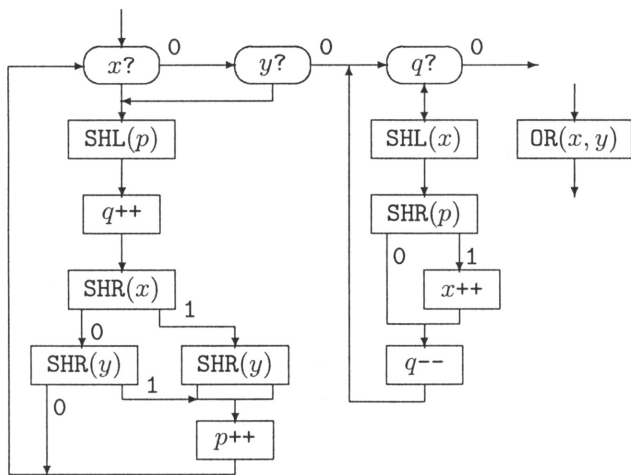


Blok $\text{ADD}(x, y)$ bol popísaný a definovaný v zadaniach.

Blok $\text{SHL}(x)$ vynásobí číslo v registri x dvoma. Potrebuje jeden pomocný register p , do ktorého „presype“ obsah registra x , potom obsah p presýpa do x , pričom za každé zníženie registra p dvakrát zvýši obsah registra x .

Blok $\text{SHR}(x)$ celočíselne delí číslo v registri x dvoma. Má dva výstupy označené 0 a 1, pričom výstup 0 sa použije, ak bolo číslo v registri x pri vstupe do bloku párne a výstup 1, ak bolo nepárne. Pracuje podobne ako blok SHL s tým, že najprv za každé dve zníženia registra x raz zvýšime pomocný register a potom presýpame obsah pomocného registra naspäť do x .

Blok $\text{OR}(x, y)$ je najzložitejší. Jeho funkciou je priradiť do registra x bitový OR čísel uložených v registroch x a y a zároveň vynulovať register y . Budeme to robiť podobne ako vo vzorovom riešení krajského kola. Pomocou blokov SHR odoberieme posledný bit zo zápisu oboch čísel x , y . Ak je aspoň jeden z odobratých bitov jednotkový, pridáme na koniec zápisu čísla v pomocnom registri p jednotku (t.j. p vynásobíme dvoma pomocou bloku SHL a potom k nemu pripočítame jednotku), v opačnom prípade pridáme na koniec p nulu (t.j. vynásobíme ho dvoma). Takto sa nám bude v registri p postupne objavovať bitový súčet čísel x a y , avšak s bitmi zapísanými v obrátenom poradí. Na koniec teda musíme obsah registra p otočený zapísať do registra x , čo spravíme opäť odoberaním posledného bitu zápisu p a jeho pridávaním na koniec zápisu x . Aby sme vedeli, koľkokrát máme túto operáciu spraviť, počítame si počet platných bitov registra p v pomocnom registri q .

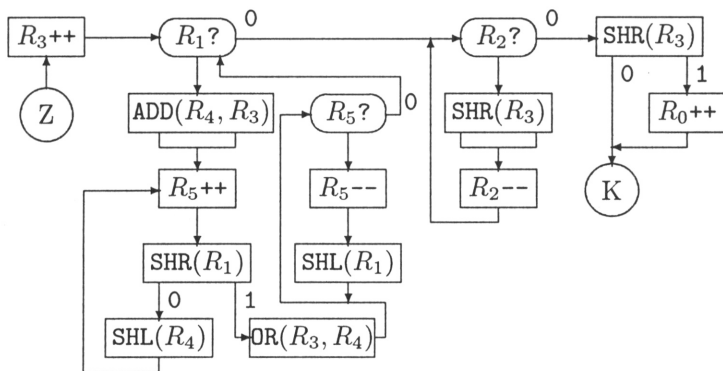


Nakoniec nám zostáva popísať samotný stroj. Skladá sa z dvoch hlavných cyklov. V prvom cykle sa v registri R_3 postupne počítajú kódy mno-

žín S_i , po skončení i -teho cyklu R_3 obsahuje kód s_i . Zároveň sa v každom prechode odoberie z množiny M jej najmenší prvok.

Odoberanie prvku sa robí v dvoch menších cykloch spolu s výpočtom kódu množiny S'_{i-1} , ktorý bude uložený v registri R_4 . Na začiatku si kód s_{i-1} skopírujeme z registra R_3 aj do registra R_4 . V prvom cykle postupne delíme obsah registra R_1 (t.j. kód množiny M) dvoma a obsah registra R_4 naopak násobíme dvoma, pričom si počítame počet prechodov cyklu v registri R_5 . Keď sa nám nepodari vydeliť obsah R_1 dvoma bezo zvyšku, narazili sme na najmenší prvok. V tomto okamihu máme v R_4 kód S'_{i-1} , ktorý pomocou bloku OR pridáme k obsahu R_3 , čím nám v tomto registri vznikne kód množiny S_i . Na záver v druhom cykle po sebe upraceme, t.j. vynásobíme obsah R_1 takou mocninou dvojky, akou sme ho v prvom cykle vydělili. Všimnite si, že týmto postupom sa nám automaticky vynuloval najnižší nenulový bit registra R_1 , t.j. odobrali sme najmenší prvok množiny M .

Keď z M odoberieme posledný prvok, jej kód uložený v registri R_1 sa vynuluje a riadenie prejde do druhého hlavného cyklu. V tomto cykle overíme, či číslo s uložené v registri R_2 , patrí do množiny, ktorej kód je uložený v registri R_3 . Hodnotu R_3 stačí s krát vydeliť dvoma a potom zistiť, či posledný bit registra je 1.



P – III – 4

Smerovník pri prameni číslo i nech ukazuje na prameň číslo $s[i]$, pre $1 \leq i \leq N$. Hovoríme, že pramene $p_1, p_2, \dots, p_k$ tvoria *cyklus*, ak od

prameňa p_1 ukazuje smerovník k prameňu p_2 , od p_2 k p_3 a tak ďalej, až od prameňa p_k k prameňu p_1 .

Ak vyštartujeme z ľubovoľného prameňa p_1 a sledujúc smerovníky prechádzame postupne pramene $p_2, p_3, \dots$, po najviac N krokoch sa nám stane, že prideme k prameňu p_{k+1} , pri ktorom sme už boli, t.j. $p_{k+1} = p_j$ pre nejaké $j \leq k$. Keďže smerovník ukazujúci na prameň j bol vyrobený iba jeden, musí byť buď $p_k = p_{j-1}$, alebo $j = 1$. Ak je k najmenšie také, že $p_{k+1} = p_j$, $j \leq k$, potom $j = 1$, a teda pramene $p_1, \dots, p_k$ tvoria cyklus. Takýmto spôsobom vieme pre každý prameň určiť cyklus, do ktorého patrí.

Vzorový program využíva fakt, že ak vymeníme smerovníky pri dvoch prameňoch, ktoré patria do rovnakého cyklu, tento cyklus sa nám rozdelí na dva, t.j. počet cyklov sa zvýši o 1. Ak naopak vymeníme smerovníky pri prameňoch z rôznych cyklov, tieto dva cykly sa spoja do jedného.

Označíme si pramene, ktoré patria do toho istého cyklu ako prameň 1. Postupne budeme hľadať ďalšie cykly, ktoré budeme pripájať k cyklu obsahujúcemu prameň 1. Vždy, keď nájdeme nový cyklus, označíme si všetky pramene, ktoré doň patria a zároveň spravíme výmenu smerovníkov, ktorou sa tento cyklus pripojí k cyklu obsahujúcemu prvý prameň. Všimnime si, že novovzniknutý cyklus obsahuje práve doposiaľ označené pramene. Ďalší cyklus potom hľadáme medzi neoznačenými prameňmi.

Implementácia. Na označovanie prameňov používame pole s . Prameň j považujeme za označený, ak $s[j] \leq 0$. Nový cyklus začíname hľadať od takého neoznačeného prameňa i , že všetky pramene s číslom menším ako i sú označené. Prameň i ako reprezentanta nového cyklu označíme $s[i] = -1$, ostatné pramene cyklu označíme nulou. Keď pri prechádzaní cyklom opäť natrafíme na prameň i , cyklus sme uzatvorili. Rovnakým postupom hľadáme ďalší cyklus, pričom vieme, že každý prameň na tomto cykle má číslo väčšie ako i .

Na záver vymeníme jeden smerovník z každého cyklu so smerovníkom pri niektorom prameni z cyklu obsahujúceho prameň 1, t.j. postupne vymieňame smerovník pri prameni 1 so smerovníkmi pri prameňoch označených -1 (reprezentanti cyklov). Potrebný počet výmen je o jednotku menší ako počet cyklov.

Časová a pamäťová zložitosť. Pamäťová zložitosť je zrejme $O(N)$. Časová zložitosť algoritmu je tiež $O(N)$, pretože s každým prvkom poľa s pracujeme maximálne trikrát (dvakrát pri hľadaní cyklov a na záver robíme ešte jeden prechod poľom s).

Úlohu zo zadania si trochu zjednodušíme a venujme sa len podstate úlohy. Je jasné, že ak sa z akéhokoľvek dôvodu budeme presúvať medzi dvoma mestami, tak optimálne bude, ak sa medzi nimi budeme presúvať najkratšou možnou cestou. Nech teda $h_{i,j}$ ($1 \leq i, j \leq N$) označuje dĺžku najkratšej cesty medzi mestom i a mestom j . Riešenie podúlohy, ako zistiť dĺžky najkratších ciest je uvedené o pár odstavcov nižšie.

Takto zredukovanú úlohu budeme riešiť metódou dynamického programovania. Označme $E_{i,j}$ ($1 \leq i \leq N$, $0 \leq j \leq K$) dĺžku najkratšej trasy končiacej v deň j v meste i , takej, že sme videli filmy $p_1, p_2, \dots, p_j$ a pritom sme posledný film p_j videli v meste i . Ak hodnota $E_{i,j}$ neexistuje (t.j. neexistuje trasa popísaných vlastností), položíme $E_{i,j} = \infty$.

Hodnoty $E_{i,j}$ budeme postupne počítat z iných skôr vypočítaných hodnôt $E_{i,j}$ a z hodnôt $h_{i,j}$. Začneme zrejme takto: $E_{1,0} = 0$ a $E_{i,0} = \infty$ ($2 \leq i \leq N$).

Počítajme hodnotu $E_{i,j}$ pre $1 \leq j \leq N$. Máme dve možnosti: V meste i nehrajú film p_j . Trasa požadovaných vlastností končiaca v meste i neexistuje, preto $E_{i,j} = \infty$.

Druhá možnosť je, že film p_j v meste i hrajú. Rozoberme túto možnosť. Na to, aby sme videli filmy $p_1, p_2, \dots, p_j$ sme museli vidieť filmy $p_1, p_2, \dots, p_{j-1}$. Film p_{j-1} sme mohli vidieť v nejakom meste s . Do mesta s sme sa pritom zrejme dostali najkratšou trasou, končiacou v tomto meste. Dĺžka tejto trasy je $E_{s,j-1}$. Z mesta s do mesta i sme takisto museli ísť najkratšou cestou. Teda dĺžka takejto trasy bude $E_{s,j-1} + h_{s,i}$. Prostým vyskúšaním všetkých možných miest s dostaneme dĺžku najkratšej cesty $E_{i,j}$:

$$E_{i,j} = \min\{E_{s,j-1} + h_{s,i} : 1 \leq s \leq N\}.$$

Najkratšia trasa, potrebná na zhliadnutie všetkých K filmov končí v nejakom meste i . Stačí nám teda vybrať z dĺžok trás, ktoré končia v jednotlivých mestách tú najmenšiu. Pre dĺžku optimálnej trasy E teda platí

$$E := \min\{E_{i,K} : 1 \leq i \leq N\}.$$

Ostáva nám ešte zistiť, cez ktoré mestá vlastne optimálna trasa vedie. Označme $D_{i,j}$ mesto, z ktorého sme prišli v deň j do mesta i , ak by sme išli po optimálnej trase končiacej v meste i v deň j . Hodnotu $D_{i,j}$ budeme počítat súbežne s hodnotou $E_{i,j}$. $D_{i,j}$ bude vlastne to mesto s , pre ktoré bude $E_{s,j-1} + h_{s,i}$ minimálne.

Mesto, v ktorom končí hľadaná optimálna trasa (t.j. také, pre ktoré je $E_{i,K}$ minimálne) označme x_K . Potom predchádzajúce mesto na optimálnej trase bude $x_{K-1} = D_{x_K, K}$. Vo všeobecnosti mesto na optimálnej trase, z ktorého sme prišli do mesta x_i bude mesto $x_{i-1} = D_{x_i, i}$.

Nakoniec venujme pár slov otázke, ako vzdialenosti $h_{i,j}$ ($1 \leq i, j \leq \leq N$) medzi jednotlivými mestami počítať. Použijeme štandardný Floyd-Warshallov algoritmus. Vstupom tohto algoritmu je matica $h_{i,j}$ ($1 \leq \leq i, j \leq N$), obsahujúca dĺžky ciest, spájajúcich jednotlivé dvojice miest (pre cestu vedúcu medzi mestami i a j s dĺžkou l položíme $h_{i,j} = h_{j,i} = l$, ak medzi i a j nevedie žiadna cesta, položíme $h_{i,j} = h_{j,i} = \infty$). Výstupom algoritmu je matica h , v ktorej $h_{i,j}$ je minimálna vzdialenosť, ktorú musíme precestovať, aby sme sa dostali z mesta i do mesta j .

Algoritmus bude pracovať v N cykloch. Po vykonaní k -teho cyklu ($0 \leq k \leq N$) bude platiť, že $h_{i,j}$ je dĺžka najkratšej trasy z mesta i do mesta j , ktorá prechádza len cez mestá s číslom menším alebo rovným k . Na začiatku (t.j. po vykonaní 0 cyklov) je v $h_{i,j}$ uložená dĺžka priamej trasy bez prechádzania cez iné vrcholy. Pri vykonávaní k -teho cyklu, dĺžka trasy $h_{i,j}$ môže byť buď rovnaká ako v predchádzajúcom cykle (ak nevyužijeme možnosť viesť trasu z mesta i do mesta j cez mesto k), alebo rovná $h_{i,k} + h_{k,j}$ (ak trasu z i do j vedieme cez mesto k). Vždy si samozrejme vyberieme kratšiu možnosť.

Implementácia. Dĺžky ciest načítavame priamo do poľa h , v ktorom aj počítame vzdialenosti medzi mestami F-W algoritmom. Na výpočet si nepotrebujeme pamätať všetky hodnoty $E_{i,j}$, stačí si pamätať iba dva stĺpce pre j a $j + 1$. To robíme v poli `optim`, pričom `optim[0]` obsahuje hodnoty $E_{i,j}$ pre j párne a `optim[1]` pre j nepárne. Ak však chceme zrekonštruovať optimálnu trasu, potrebujeme si pamätať aspoň hodnoty $D_{i,j}$. Na tie nám však stačí typ `byte`. Hodnoty $D_{i,j}$ máme v poli `pred`, ktoré je vo vzorovom programe alokované dynamicky, aj keď obmedzenia v zadaní dovoľovali použiť statické pole.

Časová a pamäťová zložitosť. Časová zložitosť celého algoritmu bude $O(N^3 + KN^2)$, z toho $O(N^3)$ je Floyd-Warshallov algoritmus. Pamäťová zložitosť bude $O(N^2 + KN)$, kde $O(N^2)$ pamäti zaberá matica h a $O(NK)$ zaberajú matice E a D .

Poznámka. Existuje algoritmus, ktorý nepotrebuje úvodné predvypočítanie vzdialeností F-W algoritmom a ktorý na výpočet každého stĺpca matice E používa modifikáciu Dijkstrovho algoritmu. Tento algoritmus má časovú zložitosť $O(K(M \log N))$, resp. $O(KN^2)$ (podľa implementácie Dijkstrovho algoritmu) a pamäťovú zložitosť $O(M + NK)$.

Přípravné soustředění před 41. MMO

V průběhu 49. ročníku se konalo výběrové soustředění pro přípravu na mezinárodní matematickou olympiádu bezprostředně po skončeném celostátním kole kategorie A, a to od 17. do 21. dubna 2000 v Jevíčku. Na soustředění byli pozváni nejlepší řešitelé III. kola kategorie A s přihlédnutím k výsledkům II. kola. Soustředění bylo zaměřeno na přípravu reprezentantů a ke konečné nominaci šestičlenného družstva.

Úspěšnost jednotlivých studentů ukazuje následující tabulka:

Jan Houštěk	7/7, G Jirsíkova 244, Pelhřimov	91,5
Jan Herman	3/4, G tř. Kpt. Jaroše 14, Brno	76,5
Jan Kynčl	5/6, G Kostelní 259, Jilemnice	86,5
Jan Pipek	7/8, G Parlérova 2/118, Praha 6	61
Jaroslav Hájek	2/4, G 17. listopadu 526, Bílovec	76
Ondřej Suchý	6/7, G Mikulášské nám. 23, Plzeň	63
Josef Křišťan	6/7, G Mikulášské nám. 23, Plzeň	58
Rudolf Stolař	3/4, G tř. Kpt. Jaroše 14, Brno	63

Na základě uvedených výsledků, v nichž jsou započítány i výsledky oblastního a celostátního kola, bylo prvních šest vybráno do reprezentativního družstva a sedmý byl určen jako náhradník. Toto družstvo nás reprezentovalo i na již tradičním střetnutí s družstvem Slovenska.

Jednotlivé semináře vedli a úlohy připravili:

dr. *Martin Panák* (17.4.),

dr. *Jaroslav Švrček* (18.4.),

doc. *Jaromír Šimša* (19.4.),

dr. *Pavel Calábek* (20.4.),

dr. *Karel Horák* (21.4.).

Úlohy zadané na přípravném soustředění

1. Najděte všechna přirozená čísla n taková, že $2^n - 1$ je násobkem čísla 3 a $\frac{1}{3}(2^n - 1)$ je dělitelem $4m^2 + 1$ pro nějaké m .

2. Mějme přirozená s, t a necht' (x, y) je uspořádaná dvojice celých čísel. Změnou dostaneme z dvojice (x, y) dvojici $(x + t, y - s)$. Dvojici (x, y) nazveme *dobrou*, jestliže po nějakém počtu změn (i nulovém) dostaneme dvojici soudělných čísel.

a) Rozhodněte, jestli je (s, t) „dobrá“ dvojice.

b) Ukažte, že pro libovolné s a t existují celá x, y taková, že dvojice (x, y) není dobrá.

3. Necht' $X = \{1, 2, \dots, n\}$, $n \geq 2$ a necht' $A_1, A_2, \dots, A_k$ jsou podmnožiny množiny X takové, že pro libovolné indexy $1 \leq i_1, i_2, i_3, i_4 \leq k$ platí

$$|A_{i_1} \cup A_{i_2} \cup A_{i_3} \cup A_{i_4}| \leq n - 2.$$

Dokažte, že $k \leq 2^{n-2}$.

4. Necht' M, N jsou takové vnitřní body trojúhelníku ABC , pro něž platí $|\sphericalangle MAB| = |\sphericalangle NAC|$ a $|\sphericalangle MBA| = |\sphericalangle NBC|$. Dokažte, že platí

$$\frac{|AM| \cdot |AN|}{|AB| \cdot |AC|} + \frac{|BM| \cdot |BN|}{|BA| \cdot |BC|} + \frac{|CM| \cdot |CN|}{|CA| \cdot |CB|} = 1.$$

5. Funkce $f: \langle 0; 1 \rangle \rightarrow \mathbb{R}$ taková, že platí $f(0) = f(1) = 0$, vyhovuje pro všechny dvojice u, v různých čísel z intervalu $\langle 0, 1 \rangle$ podmínce

$$|f(u) - f(v)| \leq |u - v|.$$

Dokažte, že pro libovolná $a, b \in \langle 0, 1 \rangle$ platí nerovnost

$$|f(a) - f(b)| \leq \frac{1}{2}.$$

6. Každý bod prostoru je obarven bílou nebo černou barvou. Dokažte, že (v prostoru) lze vybrat 5 bodů stejné barvy tak, že jeden z nich je těžištěm čtyřstěnu, jehož vrcholy tvoří zbylé 4 vybrané body.

7. Úhlopříčky konvexního čtyřúhelníku $ABCD$ se protínají v bodě R . Body P, Q necht' jsou po řadě středy stran AB, CD . Jestliže je čtyřúhelník $ABCD$ tětíkový, pak přímky, které jsou kolmé k BD, AC, AD a procházejí po řadě body P, Q, R , se protínají v jednom bodě. Dokažte.

8. Nechť A je lichá a B sudá číslice. Dokažte, že pro každé přirozené číslo n existuje přirozené číslo, které je dělitelné číslem 2^n a které nemá ve svém dekadickém zápise číslice různé od A a B .

9. Najděte nejmenší celé číslo c s vlastností: Splňuje-li nekonstantní mnohočlen $P(x)$ s celočíselnými koeficienty podmínku $P(1) = P(2) = 0$, pak aspoň jeden z jeho koeficientů není větší než c .

10. Osa úhlu BAC protne stranu BC trojúhelníku ABC v bodě D . Na polopřímkách AB a AC jsou vybrány po řadě body M a N tak, že $|\sphericalangle MDA| = |\sphericalangle ABC|$ a $|\sphericalangle NDA| = |\sphericalangle ACB|$. Označme P průsečík přímků AD a MN . Dokažte rovnost $|AD|^3 = |AB| \cdot |AC| \cdot |AP|$.

11. Součin kladných čísel $a_1, a_2, \dots, a_n$ je roven 1. Dokažte nerovnost

$$\frac{1}{n-1+a_1} + \frac{1}{n-1+a_2} + \dots + \frac{1}{n-1+a_n} \leq 1.$$

12. Vrcholy A, B, C ostroúhlého trojúhelníku ABC leží po řadě na stranách B_1C_1, C_1A_1 a A_1B_1 trojúhelníku $A_1B_1C_1$, přičemž platí podobnost $\triangle A_1B_1C_1 \sim \triangle ABC$. Dokažte, že průsečíky výšek obou trojúhelníků mají stejnou vzdálenost od středu kružnice opsané trojúhelníku ABC .

13. Osy AA_1, BB_1, CC_1 úhlů trojúhelníku ABC se protínají v bodě M (A_1, B_1, C_1 jsou body stran trojúhelníku ABC).

Dokažte: Jestliže jsou si poloměry kružnic vepsaných do trojúhelníků MB_1A a MC_1A rovny a poloměry kružnic vepsaných do trojúhelníků MC_1B a MA_1B jsou si také rovny, pak je trojúhelník ABC rovnostranný.

14. Určete všechny reálné polynomy f vyhovující rovnici

$$f(x^2) = f(x)f(x-1).$$

15. Z Wilsonovy věty je známo, že posloupnost $\left(\frac{n!+1}{n+1}\right)_{n=0}^{\infty}$ obsahuje nekonečně mnoho celých čísel.

- Určete všechna přirozená čísla a s vlastností, že posloupnost $\left(\frac{n!+a}{n+a}\right)$ obsahuje nekonečně mnoho celých čísel.
- Určete všechny celočíselné členy této posloupnosti pro $a=2$ a $a=5$.

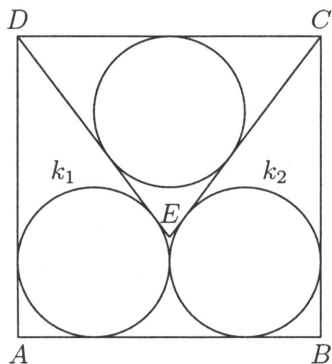
16. Rozhodněte, zda existuje rostoucí funkce $f: \mathbb{N} \rightarrow \mathbb{N}$ s následujícími vlastnostmi:

$$\begin{aligned} f(1) &= 2, \\ f(f(n)) &= f(n) + n. \end{aligned}$$

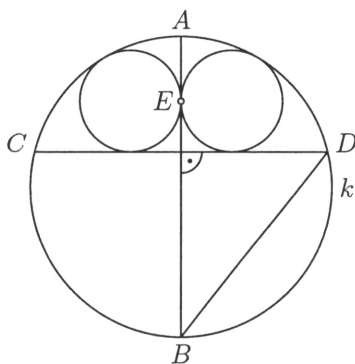
17. V rovině je dán trojúhelník $A_1A_2A_3$ a libovolný bod P_0 . Položme $A_{k+3} = A_k$ pro $k \geq 1$ a sestrojme posloupnost bodů $P_0, P_1, P_2, \dots$ tak, že pro každé $k \geq 1$ bude bod P_k obrazem bodu P_{k-1} při otočení o úhel 120° (v záporném smyslu) okolo středu A_k . Jestliže $P_{1998} = P_0$, je trojúhelník $A_1A_2A_3$ rovnostranný. Dokažte.

18. Je dán pravidelný n -úhelník, $n \geq 5$; označme A a B dva jeho sousední vrcholy, O jeho střed. V rovině n -úhelníku se pohybuje trojúhelník XYZ shodný s trojúhelníkem OAB , a to tak, že nejprve $X = O, Y = A, Z = B$ a potom Y a Z probíhají oba celý obvod n -úhelníku, přičemž X leží stále uvnitř n -úhelníku. Určete množinu všech možných poloh vrcholu X .

19. Dvě shodné kružnice k_1, k_2 mají vnější dotyk a současně se zevnitř dotýkají po řadě stran AD a AB , resp. AB a BC daného čtverce $ABCD$ (obr. 58). Uvažujme tečny DE, CE po řadě ke kružnicím k_1, k_2 . Dokažte, že kružnice vepsaná trojúhelníku CDE je shodná s kružnicemi k_1 a k_2 .



Obr. 58



Obr. 59

20. Nechť AB je průměr a CD tětiva dané kružnice k , přičemž $AB \perp CD$. Uvažujme dvojici shodných kružnic k_1, k_2 , které se vně dotýkají v bodě E ležícím na průměru AB (obr. 59). Dokažte, že trojúhelník BDE je rovnoramenný.

6. česko-slovenské střetnutí

MODRA-PIESOK 7.–10. ČERVNA 2000

Dvě šestice úspěšných finalistů matematické olympiády kategorie A vybraných pro reprezentaci České a Slovenské republiky na MMO se už po šesté utkaly ve vzájemném střetnutí, jehož organizace se v tomto roce ujala *Slovenská komisia matematickej olympiády*. Spolu se svými vedoucími byli reprezentanti pozváni do rekreační Zochovy chaty bratislavské Juventy v Modre-Piesok nedaleko Bratislavy. Tam po dva dny řešili dvě trojice úloh za podmínek obdobných třetímu kolu kategorie A (4,5 hodiny na tři úlohy). Z našich reprezentantů se bohužel omluvil Jan Houštěk, kterého v tomto klání nahradil Josef Křišťan. Slovenské mužstvo shodou okolností vystoupilo rovněž s jedním náhradníkem, Balasze Keszegha nahradil Tomáš Kulich.

Pořadí	Jméno	Země	body	Součet
1.	Vladimír Zajac	SVK	772777	37
2.	Miroslava Sotáková	SVK	771672	30
3.	Tomáš Jurík	SVK	722671	25
4.	Katarína Quittnerová	SVK	710741	20
5.	<i>Jan Kynčl</i>	CZE	703711	19
6.	Peter Májek	SVK	710711	17
7.	<i>Jaroslav Hájek</i>	CZE	711412	16
8.	<i>Josef Křišťan</i>	CZE	604121	14
9.	<i>Jan Herman</i>	CZE	700600	13
10.	Tomáš Kulich	SVK	700010	8
11.–12.	<i>Ondřej Suchý</i>	CZE	000200	2
11.–12.	<i>Rudolf Stolař</i>	CZE	200000	2

Jak je vidět z připojené tabulky, nedopadli tentokrát naši reprezentanti ve společném měření sil před 41. MMO moc dobře. O nic lépe si nevedli ani v tradičním volejbalovém střetnutí (prohráli 0 : 3).

Naš tým doprovázeli RNDr. *Jaroslav Švrček*, CSc., a Mgr. *Pavel Calábek*, Dr., slovenský tým vedli *Eugen Kováč* a *Juraj Földes*.

Texty soutěžních úloh

1. Dokažte, že pokud kladná reálná čísla a, b, c splňují nerovnost

$$5abc > a^3 + b^3 + c^3,$$

pak existuje trojúhelník s délkami stran a, b, c . (MO Bělorusko, 1999)

2. Je dán trojúhelník ABC a jemu vepsaná kružnice k . Kružnice k_a, k_b, k_c protínají ortogonálně kružnici k a úsečky BC, CA, AB jsou (v tomto pořadí) jejich tětivy. Kružnice k_a, k_b se znovu protínají v bodě C' , kružnice k_c, k_a v bodě B' a kružnice k_b, k_c v bodě A' . Dokažte, že poloměr kružnice opsané trojúhelníku $A'B'C'$ je polovinou poloměru kružnice k .

Poznámka. Říkáme, že dvě kružnice se protínají *ortogonálně*, jestliže jsou jejich tečny v každém společném bodě navzájem kolmé.

(jury MMO 1999)

3. Dokažte, že přirozené číslo n je mocninou 2, právě když existuje celé číslo m takové, že $2^n - 1$ je dělitelem čísla $m^2 + 9$. (jury MMO 1998)

4. Nechť P je mnohočlen s celočíselnými koeficienty. Dokažte, že mnohočlen Q , kde

$$Q(x) = P(x^4)P(x^3)P(x^2)P(x) + 1,$$

nemá celočíselný kořen.

(E. Kováč)

5. Je dán rovnoramenný lichoběžník $ABCD$ se základnami AB a CD . Kružnice vepsaná trojúhelníku BCD se dotýká strany CD v bodě E . Nechť F je takový bod na ose úhlu DAC , že přímky EF a CD jsou navzájem kolmé. Kružnice opsaná trojúhelníku ACF protíná přímku CD v bodech C a G . Dokažte, že trojúhelník AFG je rovnoramenný.

(USAMO, 1999)

6. Každé celé číslo je obarveno jednou z barev červená, modrá, zelená a bílá. Nechť x a y jsou lichá čísla taková, že $|x| \neq |y|$. Dokažte, že existují nějaká dvě celá čísla stejné barvy, jejichž rozdíl nabývá jednu z hodnot $x, y, x + y$ anebo $x - y$.

(jury MMO 1999)

Řešení úloh

1. Tvrzení dokážeme sporem. Nechť pro nějaká kladná reálná čísla a, b, c platí daná nerovnost, a přitom trojúhelník s délkami stran a, b, c neexistuje. To znamená, že pro čísla a, b, c neplatí aspoň jedna z trojúhelníkových

nerovností. Bez újmy na obecnosti (daná nerovnost je symetrická) můžeme předpokládat, že $c \geq a + b$, neboli $c = a + b + x$, kde $x \geq 0$. Po dosažení dostáváme

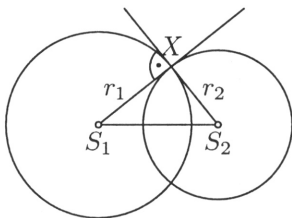
$$5ab(a + b + x) > a^3 + b^3 + (a + b)^3 + 3(a + b)^2x + 3(a + b)x^2 + x^3$$

a po úpravě

$$2a^2b + 2ab^2 > 2a^3 + 2b^3 + abx + 3(a^2 + b^2)x + 3(a + b)x^2 + x^3.$$

Protože poslední čtyři členy součtu na pravé straně jsou nezáporné, plyne odtud $2a^2b + 2ab^2 > 2a^3 + 2b^3$, což je ekvivalentní nerovnosti $(a + b) \times (a - b)^2 < 0$, která pro kladná čísla a, b zjevně neplatí. Tím jsme dospěli ke sporu, a tvrzení úlohy je tak dokázáno.

2. Především si uvědomme, že kružnice $k_1(S_1; r_1)$ a $k_2(S_2; r_2)$ se ortogonálně protínají, právě když $r_1^2 + r_2^2 = |S_1S_2|^2$. To snadno plyne z Pythagorovy věty pro trojúhelník S_1S_2X (obr. 60).



Obr. 60

Označme nyní V střed kružnice k vepsané trojúhelníku ABC , r její poloměr D, E, F po řadě její dotykové body se stranami BC, CA, AB a P, Q, R středy úseček EF, FD, DE (obr. 61). Ukážeme, že body Q a R leží na kružnici k_a .

Z Eukleidovy věty o odvěsně pro trojúhelníky VBD a VCD plyne, že $|VQ| \cdot |VB| = r^2$ a $|VR| \cdot |VC| = r^2$. To znamená, že body B, C, R, Q leží na nějaké kružnici se středem O_a a poloměrem r_a . Pro mocnost bodu V k této kružnici platí $|VQ| \cdot |VB| = r^2 = |VO_a|^2 - r_a^2$, což znamená, že kružnice opsaná čtyřúhelníku $BCRQ$ protíná kružnici k ortogonálně. Body B a C však může procházet nejvýše jedna kružnice, která kružnici k protíná ortogonálně. Kružnice opsaná čtyřúhelníku $BCRQ$ je tedy kružnice k_a .

Poznámka. To, že body B, C, R, Q leží na kružnici, lze snadno odvodit i z rovnosti obvodových úhlů v tětíivových čtyřúhelnících $VFBD$ a $VQDR$ (jsou tětíivové podle Thaletovy věty): $|\sphericalangle VBD| = |\sphericalangle VBF| = |\sphericalangle QDV| = |\sphericalangle QRV|$, takže i čtyřúhelník $BCRQ$ je tětíivový (obr. 62).

Pokud n není mocninou dvojky, je dělitelné nějakým lichým číslem $l \geq 3$, a protože $2^l - 1$ vždy dělí $2^n - 1$, dělí i $m^2 + 9$. Číslo $2^l - 1$ dává při dělení čtyřmi zbytek -1 , stejný zbytek musí mít i některý z jeho prvočíselných dělitelů, takže existuje prvočíslo p takové, že $p \equiv -1 \pmod{4}$ a zároveň $p \mid m^2 + 9$. Z *malé Fermatovy věty* tak plyne

Pro $p \neq 3$ můžeme na pravé straně podle stejného tvrzení dosadit $3^{p-1} \equiv 1$ a zároveň $(-1)^{\frac{1}{2}(p-1)} \equiv -1$, protože prvočíslo p jsme našli tak, že $p-1$

není dělitelné čtyřmi, takže exponent $\frac{1}{2}(p-1)$ je lichý. Vidíme tedy, že poslední kongruence vede ke sporu. Pokud ovšem $p=3$, je 2^l-1 dělitelné třemi, takže $2^l \equiv 1 \pmod{3}$, a zároveň pro liché l platí $2^l \equiv (-1)^l \equiv -1 \pmod{3}$, což je opět spor. Tím je první implikace dokázána.

Předpokládejme nyní, že $n=2^k$. Pro $k=0$ i $k=1$ tvrzení zřejmě platí, předpokládejme tedy, že $k \geq 2$. Potom

$$2^n - 1 = (2+1)(2^2+1)(2^{2^2}+1)\dots(2^{2^{k-1}}+1). \quad (1)$$

Uvědomme si, že pro $\alpha < \beta$ jsou čísla $2^{2^\alpha}+1$ a $2^{2^\beta}+1$ nesoudělná. Pro jejich libovolný společný dělitel d , který je nutně lichý, můžeme totiž modulo d psát $-1 \equiv 2^{2^\beta} \equiv 2^{2^\beta \cdot 2^{\beta-\alpha}} \equiv (2^{2^\beta})^{2^{\beta-\alpha}} \equiv 1$, což znamená, že je $d=1$. Podle tzv. *čínské zbytkové věty* (čísla $2^{2^l}+1$ jsou navzájem nesoudělná) tak existuje přirozené číslo c , které je řešením soustavy kongruencí

$$c \equiv 2^{2^l} \pmod{2^{2^{l+1}}+1}, \quad l \in \{0, 1, \dots, k-2\}.$$

Pro takové c pak dostáváme $c^2+1 \equiv 0 \pmod{2^{2^{l+1}}+1}$ ($l \in \{0, 1, \dots, k-2\}$), takže podle (1) 2^n-1 dělí $9(c^2+1) = (3c)^2+9$. Stačí tedy vzít $m=c^2$.

4. Všimněme si, že pro každé celé číslo n platí $n^3 \equiv n \pmod{3}$. To snadno plyne z rozkladu $n^3 - n = n(n^2 - 1) = (n-1)n(n+1)$. Je tedy také $n^4 \equiv n^2 \pmod{3}$ a pro mnohočlen P s celočíselnými koeficienty tak platí $P(n^3) \equiv P(n) \pmod{3}$ a $P(n^4) \equiv P(n^2) \pmod{3}$. Pro mnohočlen Q , který má rovněž celočíselné koeficienty, tak z posledních dvou kongruencí plyne

$$Q(n) \equiv (P(n)P(n^2))^2 + 1 \pmod{3}.$$

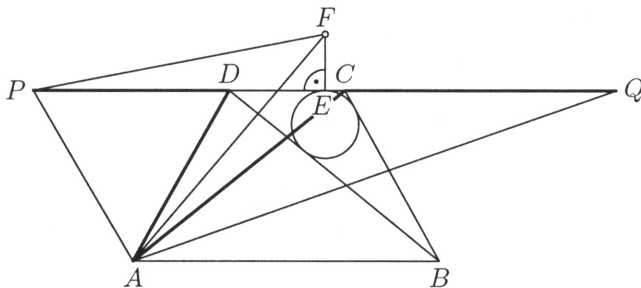
Protože pro každé celé číslo m platí buď $m^2 \equiv 0 \pmod{3}$, nebo $m^2 \equiv 1 \pmod{3}$, vidíme, že kongruence $Q(x) \equiv 0 \pmod{3}$ nemá žádné řešení v oboru celých čísel, tím spíše nemůže mít celočíselné řešení rovnice $Q(x) = 0$.

5. Ukážeme, že $|FA| = |FG|$. Budeme postupovat odzadu. Na polopřímce opačné k DC vezměme bod P tak, že $|DP| = |DA|$, a podobně na polopřímce opačné k CD vezměme bod Q tak, že $|CQ| = |CA|$. Pomocí známých vztahů pro úseky stran trojúhelníku BCD vzhledem k jeho

vepsané kružnici dostáváme (obr. 63; zároveň využíváme toho, že daný lichoběžník $ABCD$ je rovnoramenný)

$$\begin{aligned}
 |PE| &= |PD| + |DE| = |DA| + \frac{|BD| + |CD| - |BC|}{2} = \\
 &= \frac{|BD| + |CD| + |BC|}{2}, \\
 |QE| &= |QC| + |CE| = |AC| + \frac{|BC| + |CD| - |BD|}{2} = \\
 &= \frac{|BC| + |CD| + |BD|}{2},
 \end{aligned}$$

neboli $|PE| = |QE|$. To znamená, že trojúhelník PQE je rovnoramenný a přímka EF je osou jeho strany PQ .



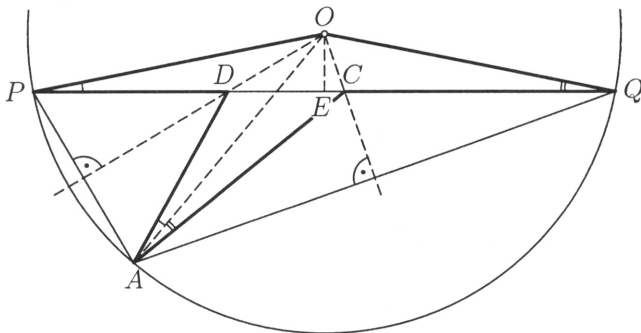
Obr. 63

Uvažme nyní kružnici opsanou trojúhelníku AQP . Její střed O leží na EF , a protože trojúhelníky APD a AQC jsou rovnoramenné, jsou DO a CO osami úseček AP a AQ (obr. 64). Odtud plyne, že

$$|\sphericalangle DAO| = |\sphericalangle DPO| = |\sphericalangle CQO| = |\sphericalangle CAO|,$$

bod O tedy leží na ose úhlu CAD , což znamená, že $O \equiv F$. Z rovnosti úhlů $|\sphericalangle CAF| = |\sphericalangle CPF|$ zároveň vidíme, že body C, A, F, P leží na kružnici, což znamená, že bod P je druhým průsečíkem přímky CD s kružnicí opsanou trojúhelníku CAF , je tedy $P \equiv G$ a $|FA| = |FP| = |FG|$, což jsme měli dokázat.

6. Zřejmě stačí najít celé číslo a takové, že mezi čtyřmi celými čísly $a, a + x, a + y, a + x + y$ najdeme dvě čísla stejné barvy. To budou ta čísla, jejichž existenci chceme dokázat.



Obr. 64

Předpokládejme naopak, že existuje obarvení celých čísel takové, že pro každé celé číslo a mají čísla $a, a + x, a + y, a + x + y$ vesměs různou barvu, tj. že existuje funkce $f: \mathbb{Z} \rightarrow \{C, M, Z, B\}$ taková, že pro každé celé číslo a platí

$$f(\{a, a + x, a + y, a + x + y\}) = \{C, M, Z, B\} \quad (1)$$

(písmeny C, M, Z, B jsme označili uvažované barvy). Ukážeme dále, že takový předpoklad vede ke sporu.

Uvažujme na množině $\mathbb{Z} \times \mathbb{Z}$ (množině všech mřížových bodů v rovině) obarvení $g: \mathbb{Z} \times \mathbb{Z} \rightarrow \{C, M, Z, B\}$ určené pro všechna $i, j \in \mathbb{Z}$ vztahem

$$g(i, j) = f(ix + jy).$$

Při takovém obarvení g mají ovšem *vrcholy libovolného jednotkového čtverce navzájem různé barvy*: vrcholům

$$(i, j), (i + 1, j), (i, j + 1), (i + 1, j + 1) \in \mathbb{Z} \times \mathbb{Z}$$

totiž odpovídají celá čísla

$$(ix + jy), (ix + jy) + x, (ix + jy) + y, (ix + jy) + x + y,$$

jež mají podle (1) vesměs různou barvu.

Tato vlastnost jednotkových čtverců už je podstatným omezením pro existenci takového obarvení. Protože sousední mřížové body (ať v řádku či v sloupci) nemohou mít stejnou barvu, obsahuje každý řádek (či sloupec) buď jen dvě barvy, které se pravidelně střídají, anebo se po sobě vyskytnou tři vrcholy navzájem různých barev.

Předpokládejme například, že v nějakém sloupci najdeme tři po sobě jdoucí vrcholy s navzájem různými barvami; bez újmy na obecnosti nechť

B

to jsou barvy M . Uvážíme-li vrcholy sousedních jednotkových čtverců,

C

zjistíme, že jsou už jednoznačně určeny především barvy obou vrcholů sousedících s prostředním vrcholem a pak i barvy zbývajících vrcholů obou sousedních trojic vrcholů:

$$\begin{array}{ccccc} B & C B C & B C B C B & & \\ Z M Z, & Z M Z, & M Z M Z M, & \dots & \\ C & B C B & C B C B C & & \end{array}$$

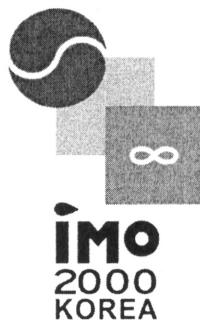
Barvy vrcholů v každém ze tří odpovídajících řádků se tudíž musejí pravidelně střídát.

Pokud se v některém řádku (či sloupci) pravidelně střídají dvě barvy (například $\dots C M C M \dots$), musejí se díky uvedené vlastnosti jednotkových čtverců v dalším řádku pravidelně střídát obě zbývající barvy:

$$\begin{array}{cc} C M C M C & \text{nebo} & C M C M C \\ Z B Z B Z & & B Z B Z B \end{array}$$

A v dalším řádku se zase budou pravidelně střídát první dvě barvy C a M atd. Vidíme, že se pak v každém řádku pravidelně střídá vždy jedna z dvojic barev C a M , resp. B a Z , a to tak, že všechny liché řádky obsahují jednu dvojici barev a všechny sudé řádky druhou dvojici.

Můžeme tedy předpokládat, že např. řádky mají tu vlastnost, že se v nich pravidelně střídají dvě barvy. Označme C a M barvy bodů $(0, 0)$ a $(1, 0)$, takže $g(0, 0) = C$ a $g(1, 0) = M$. Potom $g(y, 0) = M$ (protože y je liché). Protože i x je liché, musí být $g(0, x) \in \{B, Z\}$. Z rovnosti $g(y, 0) = f(xy) = g(0, x)$ však dostáváme spor. Tím je existence takového obarvení vyvrácena a tvrzení úlohy dokázáno.



V pořadí již 41. ročník této mezinárodní soutěže se konal 13.–25. července 2000 v Korejské republice. Soutěž proběhla v Taejonu, na půdě jedné z nejprestižnějších univerzit v Asii, kterou je poměrně mladá vysoká škola KAIST (Korea Advanced Institut of Science and Technology), založená v r. 1971. Letošního ročníku MMO se zúčastnili 462 soutěžících z 82 zemí celého světa. Logo olympiády svými dvěma čtverci symbolizuje uplynulá dvě tisíciletí, taeguk (rozdělený kruh) je tradičním symbolem Koreje (najdeme ho i na korejské státní vlajce). Použité barvy (modrá, zelená a červená, jejichž kombinací dostaneme všechny barvy spektra) pak symbolizují krásu a sílu matematiky, základu veškeré vědy a technologie.

Patronát nad organizací a průběhem 41. ročníku MMO převzal ministerský předseda Korejské republiky *Han Dong Lee*, který se osobně zúčastnil slavnostního zahájení soutěže. Korejští organizátoři připravili všem účastníkům dobré podmínky pro samotnou soutěž a také zajímavý doprovodný kulturní a společenský program. Soutěžící měli možnost navštívit korejský skanzen v Yonginu poblíž Suwonu, archeologická naleziště v Kyóngju a také pavilony světové výstavy EXPO, která se v Taejonu konala v roce 1993. Ve školském kampusu KAISTu byla po celou dobu soutěžícím k dispozici všechna sportoviště, plavecký bazén a moderně vybavená počítačová učebna.

Výběr soutěžících za Českou republiku byl proveden v Jevíčku na závěrečném soutěžním soustředění osmi nejúspěšnějších účastníků celostátního kola. Vybraní soutěžící se pak ještě zúčastnili utkání ve slovenské Modre mezi Českou republikou a Slovenskem, kde soutěžili reprezentanti obou zemí za podmínek podobných soutěži na MMO. Naše družstvo tvořila následující šestice olympioniků: *Jaroslav Hájek*, z Gymnázia M. Koperníka v Bílovci, *Jan Herman* a *Rudolf Stolař* z Gymnázia na tř. Kpt. Jaroše v Brně, *Jan Houštěk* z Gymnázia v Pelhřimově, *Jan Kynčl* z Gymnázia v Jilemnici, a *Ondřej Suchý* z Gymnázia na Mikulášském

nám. v Plzni. Vedoucím naší výpravy byl dr. *Karel Horák* z Matematického ústavu Akademie věd v Praze, pedagogickým vedoucím družstva byl dr. *Jaroslav Švrček* z Univerzity Palackého v Olomouci. Vedoucí delegace přicestoval do hlavního města Koreje Soulu kvůli výběru úloh již 13. července, ostatní čeští účastníci pak o tři dny později.

Dva dny po přeletu soutěžících, tedy 18. července se konalo slavnostní zahájení. Vlastní soutěž proběhla ve dnech 19. a 20. července v pavilonu univerzitní knihovny KAISTu. Jako obvykle žáci řešili v každém soutěžním dni po třech úlohách. Na každou trojici úloh měli vyhrazeny vždy 4,5 hodiny čistého času a za každou úlohu mohli získat maximálně 7 bodů.

Naše mladé družstvo nezklamalo. Svědčí o tom především zisk jedné stříbrné medaile *Janem Houškem* a dále tři bronzových medailí oproti jediné bronzové medaili z předešlé 40. MMO v Rumunsku. Jejich výsledky jsou uvedeny v následující tabulce:

Umístění		Body za úlohu						Body	Cena
		1	2	3	4	5	6		
395.–416.	Jaroslav Hájek, 2. roč. GMK Bílovec,	0	0	0	2	0	0	2	
205.–213.	Jan Herman, 3. roč. gymnázia, Brno, tř. Kpt. Jaroše	7	2	0	2	0	0	11	III.
90.–99.	Jan Houštěk, 7. roč. gymnázia, Pelhřimov	7	2	1	7	4	0	21	II.
139.–149.	Jan Kynčl, 5. roč. gymnázia, Jilemnice	7	1	4	2	0	2	16	III.
351.–368.	Rudolf Stolař, 3. roč. gymnázia, Brno, tř. Kpt. Jaroše	0	1	0	3	0	0	4	
205.–213.	Ondřej Suchý, 6. roč. gymnázia, Plzeň, Mikulášské nám.	7	0	0	2	0	2	11	III.
Celkem		28	6	5	18	4	4	65	

O náročnosti soutěžních úloh svědčí i nízké hranice pro zisk medailí: na třetí cenu stačilo 11 bodů, druhá cena se udělovala za 21–30 bodů

a první za alespoň 31 bodů. Řešitelů, kteří si z Taejonu odvezli zlatou medaili, bylo celkem 28. Mezi nimi byli ale čtyři, kteří získali plný počet 42 bodů: *Alexandr Usnič* z Běloruska, *Zhiwei Yun* z Číny a dva soutěžící *Alexej Pojarkov* a *Alexandr Gajfullin* z Ruska.

Neoficiální pořadí všech zúčastněných zemí s počtem získaných cen a celkovým bodovým ziskem (případná čísla v závorce upozorňují na nižší počet reprezentantů):

	I	II	III	body		I	II	III	body
ČLR	6	0	0	218	<i>Česká republika</i>	0	1	3	65
Rusko	5	1	0	215	Makedonie	0	1	2	63
USA	2	4	0	184	Kolumbie	0	0	2	61
Korea	3	3	0	172	Kuba	0	0	2	61
Bulharsko	2	3	1	169	Lotyšsko	0	0	3	60
Vietnam	3	2	1	169	Nizozemsko	0	0	2	60
Bělorusko	2	2	2	165	Brazílie	0	0	3	58
Tchaj-wan	2	3	1	164	Francie	0	0	3	58
Maďarsko	1	5	0	156	Itálie	0	0	3	57
Írán	1	4	1	155	Indonézie	0	0	2	54
Izrael	2	1	3	139	Finsko	0	0	3	52
Rumunsko	0	4	2	139	Belgie	0	0	2	51
Ukrajina	2	2	0	135	Lucembursko (4)	0	0	2	51
Indie	0	5	1	132	Maroko	0	0	1	48
Japonsko	1	2	3	125	Řecko	0	0	1	46
Austrálie	1	3	1	122	Norsko	0	0	1	45
Kanada	1	2	1	112	Estonsko	0	0	1	42
<i>Slovensko</i>	0	2	3	111	Trinidad a Tobago	0	0	0	40
Turecko	0	3	1	111	Island	0	0	0	37
Arménie	0	2	3	108	Dánsko	0	0	1	36
Německo	1	1	2	108	Litva	0	0	1	34
Velká Británie	0	2	4	96	Nový Zéland	0	0	0	34
Jugoslávie	0	1	3	93	Ázerbájdžán	0	0	0	32
Kazachstán	0	1	4	91	Kypr	0	0	0	32
Argentina	0	1	4	88	Malajsie (3)	0	0	2	32
Moldavsko (5)	0	2	3	84	Peru (4)	0	0	0	32
JAR	0	0	4	81	Španělsko	0	0	0	29
Hongkong	0	1	2	80	Irsko	0	0	0	28
Bosna a Hercegovina	0	0	4	78	Filipíny (4)	0	0	0	23
Thajsko	0	1	3	78	Uruguay (3)	0	0	0	23
Švédsko	0	2	0	77	Portugalsko	0	0	0	21
Mexiko	0	1	3	75	Srí Lanka (3)	0	0	0	21
Polsko	0	1	2	75	Ekvádor	0	0	0	19
Chorvatsko	0	0	4	73	Albánie (5)	0	0	0	17
Slovinsko	0	1	1	73	Kirgizie (4)	0	0	1	16
Gruzie	0	1	0	72	Macao	0	0	0	16
Singapur	0	1	2	71	Kuvajt (4)	0	0	0	12
Uzbekistán	0	0	2	70	Guatemala	0	0	0	11
Rakousko	0	2	1	68	Venezuela (2)	0	0	0	11
Mongolsko	0	0	4	67	Brunei (2)	0	0	0	8
Švýcarsko (4)	0	1	2	67	Portoriko	0	0	0	8

Jak je patrné z tabulky zúčastněných států, na prvních místech neo-

ficiálního pořadí jednotlivých zemí podle celkového bodového zisku se už tradičně vyskytují Čína, Rusko, Spojené státy, Korea, Bulharsko a Vietnam. Naše umístění v druhé polovině této tabulky nás rozhodně nectí.

Slavnostní zakončení 41. MMO se konalo v mezinárodním kulturním centru státní univerzity Chungnam v Taejonu. Při této příležitosti pozvali zástupci USA všechny zúčastněné delegace k účasti na dalším ročníku MMO.

Texty soutěžních úloh

(v závorce je uvedena země, která úlohu navrhl)

1. Jsou dány dvě kružnice k_1 a k_2 , které se protínají v bodech M a N . Označme l jejich společnou tečnu takovou, že bod M je blíže l než bod N . Bod dotyku přímky l s k_1 označme A a bod dotyku s k_2 označme B . Přímka vedená bodem M rovnoběžně s l protíná kružnici k_1 v dalším bodě C a kružnici k_2 v dalším bodě D . Označme dále E průsečík přímek CA a DB , P průsečík přímek AN a CD a Q průsečík přímek BN a CD . Dokažte, že $|EP| = |EQ|$.
(Maďarsko)

2. Nechť a, b, c jsou kladná reálná čísla taková, že $abc = 1$. Dokažte, že

$$\left(a - 1 + \frac{1}{b}\right)\left(b - 1 + \frac{1}{c}\right)\left(c - 1 + \frac{1}{a}\right) \leq 1.$$

(Rusko)

3. Nechť $n \geq 2$ je přirozené číslo. Na počátku je na vodorovné přímce n blech, ne všechny v témž bodě. Pro kladné reálné číslo λ definujeme *tah* následujícím způsobem:

vybereme libovolné dvě blechy v bodech A a B , přičemž bod A je nalevo od bodu B ;

blechu z bodu A necháme skočit do bodu C dané přímky napravo od B , přičemž $|BC|/|AB| = \lambda$.

Určete všechny hodnoty λ takové, že pro libovolný bod M na dané přímce a pro libovolnou počáteční polohu n blech existuje konečná posloupnost tahů, která přesune všechny blechy na místa napravo od bodu M .

(USA)

4. Kouzelník má sto karet očíslovaných od 1 do 100. Vloží je do tří krabic — červené, bílé a modré — tak, aby v každé byla aspoň jedna karta. Jeden z diváků si vybere dvě ze tří krabic, z každé vytáhne jednu kartu a oznámí součet čísel na vybraných kartách. Na základě tohoto součtu

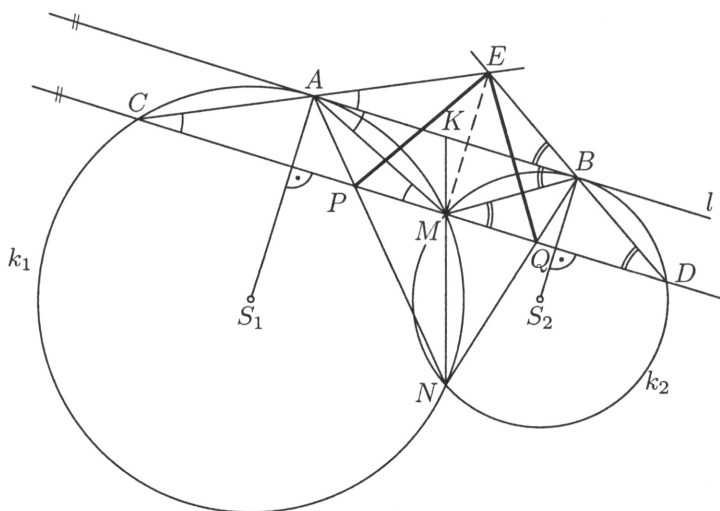
určí kouzelník krabici, z které divák kartu nevytáhl. Kolika způsoby lze rozdělit všechny karty do krabic, aby kouzelníkův trik vždy fungoval? (Rozdělení považujeme za různá, pokud se aspoň jedna karta objeví v jiné krabici.) (Rusko)

5. Rozhodněte, zda existuje kladné celé číslo n takové, že n je dělitelné přesně 2000 různými prvočísly a $2^n + 1$ je dělitelné n . (Bělorusko)

6. Nechtě AH_1 , BH_2 , CH_3 jsou výšky ostroúhlého trojúhelníku ABC . Kružnice vepsaná trojúhelníku ABC se po řadě dotýká stran BC , CA , AB v bodech T_1 , T_2 , T_3 . Označme po řadě l_1 , l_2 , l_3 přímky souměrně sružené s přímkami H_2H_3 , H_3H_1 , H_1H_2 podle os T_2T_3 , T_3T_1 , T_1T_2 . Dokažte, že přímky l_1 , l_2 , l_3 určují trojúhelník, jehož vrcholy leží na kružnici vepsané trojúhelníku ABC . (Rusko)

Řešení úloh

1. Označme K průsečík přímky MN s tečnou l obou kružnic (obr. 65).



Obr. 65

Protože bod K leží na chordále obou kružnic, má vůči nim stejnou mocnost, tj.

$$|AK|^2 = |KM| \cdot |KN| = |KB|^2.$$

A tak i čtyřúhelníky $PNDE$ a $CNQE$ jsou tětíkové. Navíc zřejmě platí

$$|\sphericalangle ANC| = |\sphericalangle AMC| \quad \text{a} \quad |\sphericalangle BND| = |\sphericalangle BMD|,$$

takže

$$|\sphericalangle EQC| = |\sphericalangle ENC| = |\sphericalangle END| = |\sphericalangle EPD|.$$

Rovnost $|\sphericalangle EQC| = |\sphericalangle EPD|$ v trojúhelníku PQE ovšem znamená, že $|EP| = |EQ|$.

2. Danou nerovnost můžeme homogenizovat vhodnou změnou proměnných: Vezměme kladná čísla x , y a z tak, aby

$$a = \frac{x}{y}, \quad b = \frac{y}{z}, \quad c = \frac{z}{x}$$

(jedna možná volba je např. $x = 1$, $y = 1/a$ a $z = 1/(ab)$). Místo původní nerovnosti tak dostaneme

$$(x - y + z)(y - z + x)(z - x + y) \leq xyz.$$

Protože každá dvě z čísel

$$u = x - y + z, \quad v = y - z + x, \quad w = z - x + y$$

mají kladný součet, je nejvýše jedno z nich záporné. V takovém případě je ovšem $uvw \leq 0 < xyz$ a nerovnost triviálně platí.

Předpokládejme tedy, že $u \geq 0$, $v \geq 0$, $w \geq 0$. Potom podle nerovnosti mezi aritmetickým a geometrickým průměrem platí

$$\sqrt{uv} = \sqrt{(x - y + z)(y - z + x)} \leq \frac{(x - y + z) + (y - z + x)}{2} = x.$$

Podobně platí i $\sqrt{vw} \leq y$ a $\sqrt{wu} \leq z$, takže $uvw \leq xyz$, jak jsme chtěli dokázat.

Jiné řešení. Protože

$$b - 1 + \frac{1}{c} = b \left(1 - \frac{1}{b} + a \right),$$

je

$$\left(a - 1 + \frac{1}{b} \right) \left(b - 1 + \frac{1}{c} \right) = b \left(a^2 - \left(1 - \frac{1}{b} \right)^2 \right) \leq ba^2. \quad (1)$$

Analogicky platí

$$\left(b - 1 + \frac{1}{c}\right)\left(c - 1 + \frac{1}{a}\right) \leq cb^2 \text{ a } \left(c - 1 + \frac{1}{a}\right)\left(a - 1 + \frac{1}{c}\right) \leq ac^2. \quad (2)$$

Jestliže jsou levé strany všech tří nerovností (1) a (2) nezáporné, můžeme nerovnosti (1) a (2) vynásobit a dostaneme, že kvadrát levé strany je nejvýše roven $ba^2cb^2ac^2 = 1$, čímž je nerovnost dokázána.

Připustíme, že je některý z činitelů záporný, např. $a - 1 + 1/b < 0$ (to můžeme dokonce předpokládat bez újmy na obecnosti, protože daná nerovnost je cyklická). Pak ovšem $a < 1 - 1/b < 1$ a $1/b < 1 - a < 1$, tj. $b > 1$, takže $c - 1 + 1/a > 0$ a $b - 1 + 1/c > 0$ a daná nerovnost je triviálně splněna. Tím je důkaz hotov.

Jiné řešení. Díky dané podmínce $abc = 1$ platí

$$\begin{aligned}\frac{1}{a}\left(a - 1 + \frac{1}{b}\right) + c\left(b - 1 + \frac{1}{c}\right) &= 2, \\ \frac{1}{b}\left(b - 1 + \frac{1}{c}\right) + a\left(c - 1 + \frac{1}{a}\right) &= 2, \\ \frac{1}{c}\left(c - 1 + \frac{1}{a}\right) + b\left(a - 1 + \frac{1}{b}\right) &= 2.\end{aligned}$$

Speciálně odtud plyne, že nejvýše jedno z čísel

$$u = a - 1 + \frac{1}{b}, \quad v = b - 1 + \frac{1}{c}, \quad w = c - 1 + \frac{1}{a}$$

je záporné. Pokud ano, pak daná nerovnost triviálně platí, jak už jsme ukázali v předchozích řešeních. Pokud $u \geq 0$, $v \geq 0$, $w \geq 0$, máme podle nerovnosti mezi aritmetickým a geometrickým průměrem

$$\begin{aligned}2 = \frac{1}{a}u + cv &\geq 2\sqrt{\frac{c}{a}uv}, \quad 2 = \frac{1}{b}v + aw \geq 2\sqrt{\frac{a}{b}vw}, \\ 2 = \frac{1}{c}w + bu &\geq 2\sqrt{\frac{b}{c}wu}.\end{aligned}$$

Je tedy $uv \leq a/c$, $vw \leq b/a$ a $wu \leq c/b$, takže $(uvw)^2 \leq 1$. Protože $uvw \geq 0$, je důkaz hotov.

Jiné řešení (řešení *Jaroslava Hájka*, na které přišel po soutěži). Daná nerovnost je invariantní vůči cyklické permutaci čísel a , b , c , proto můžeme předpokládat, že je např. $a \leq 1$ a $c \geq 1$, takže platí

$$(a - 1)(c - 1) \leq 0.$$

To lze zapsat jako $ac - a + 1 \leq c$, zároveň však je $ac - a + 1 = a(c - 1) + 1 \geq 1 > 0$, tedy

$$0 < ac - a + 1 \leq c.$$

Protože $1 - (bc - c)^2 \leq 1$, plyne z předchozí nerovnosti nerovnost

$$(1 - (bc - c)^2)(ac - a + 1) \leq c,$$

neboli

$$\begin{aligned} (1 - bc + c)(1 + bc - c)(ac - a + 1) &\leq c, \\ (1 - bc + c)\left(\frac{1}{c} + b - 1\right)(ac - a + 1) &\leq 1. \end{aligned}$$

Vydělíme-li teď levou stranu poslední nerovnosti číslem $abc = 1$ tak, že první závorku vydělíme bc a třetí číslem a , dostaneme dokazovanou nerovnost.

Poznámka. V každém z uvedených řešení lze snadno nahlédnout, že rovnost platí, právě když $a = b = c = 1$.

3. (S využitím myšlenek *Jana Kynčla*.) Označme $A_1, A_2, \dots, A_n$ pozice jednotlivých blech zleva doprava v některém okamžiku a uvažujme součet $s = |A_1 A_n| + |A_2 A_n| + \dots + |A_{n-1} A_n|$. Podívejme se, co se stane, provedeme-li tah pro blechy v bodech A_k, A_n ($1 \leq k < n$): dostaneme tak novou posloupnost bodů $A'_1, A'_2, \dots, A'_n$, pro kterou platí $A'_1 = A_1, A'_2 = A_2, \dots, A'_{k-1} = A_{k-1}, A'_k = A_{k+1}, \dots, A'_{n-1} = A_n, A'_n = A_n + \lambda(A_n - A_k)$, takže $|A'_n A'_i| = |A_n A_i| + \lambda|A_k A_n|$ pro $i < k$ a $|A'_n A'_i| = |A_n A_{i+1}| + \lambda|A_k A_n|$ pro $k \leq i < n$. Nové poloze blech odpovídá součet s' , přičemž

$$\begin{aligned} s' - s &= |A_k A_n| + (n - 1)\lambda|A_k A_n| = \\ &= ((n - 1)\lambda + 1)|A_k A_n|. \end{aligned} \tag{1}$$

Je tedy zřejmé, že pro $(n - 1)\lambda \geq 1$, neboli $\lambda \geq 1/(n - 1)$ se příslušný součet s nezmenší. Volíme-li tahy tak, že postupně skáče vždy první blecha zleva za poslední, přesunou se tak po $n - 1$ krocích všechny blechy za blechu, která byla poslední v počáteční pozici. Protože je vždy

$$s \leq (n - 1)|A_1 A_n|, \quad \text{tj.} \quad |A_1 A_n| \geq s(0)/(n - 1),$$

kde jsme jako $s(0)$ označili odpovídající součet s na začátku, octne se původně první blecha nejméně o $s(0)/(n - 1)$ dále vpravo. Vidíme, že

uvedeným postupem se lze se všemi blechami dostat libovolně daleko (jakmile se jedna blecha octne za nějakým bodem M , dostanou se tam v dalších krocích i všechny ostatní).

Nechť nyní $\lambda < 1/(n-1)$. Víme již, že při skoku k -té blechy ($1 \leq k \leq n-1$) přes poslední se příslušný součet s zmenší o hodnotu (1). Tím spíš se součet s zmenší skokem z k -té pozice přes l -tou blechu ($k < l < n$), i když se tím blecha dostane za poslední. Konečně skokem z k -té pozice přes l -tou někam vlevo od poslední blechy se příslušný součet s zmenší přesně o délku skoku, která je $(1+\lambda)|A_k A_l|$.

Vidíme tedy, že pokud se po nějakém skoku vůbec změní poloha poslední blechy, změní se nejvýše o $z \leq \lambda|A_k A_n|$, zatímco odpovídající součet se dle (1) sníží alespoň o

$$(1 - (n-1)\lambda)|A_k A_n| \geq (1 - (n-1)\lambda)z/\lambda = \lambda_0 z.$$

Ovšem tento součet se může celkem snížit nejvýše o hodnotu $s(0)$, kterou měl na počátku, to znamená, že poloha pravé krajní blechy se nikdy neposune dále než o $s(0)/\lambda$.

Odpověď. Úloze vyhovují všechna reálná čísla $\lambda \leq 1/(n-1)$.

4. Ukážeme, že hledaný počet je 12.

Předpokládejme, že máme karty rozmístěny do tří krabic tak, že kouzelníkův trik funguje. Jsou-li a, b, c, d čtyři různá čísla z množiny $\{1, 2, \dots, 100\}$ taková, že $a + b = c + d$ a karty s čísly a, b, c se nacházejí v různých krabicích, musí být karta s číslem d ve stejné krabici jako c . (Tuto základní úvahu využijeme i v dalších řešeních.)

Nechť existuje i takové, že karty s čísly $i, i+1$ a $i+2$ jsou vesměs v různých krabicích (označme je po řadě písmeny A, B a C). Protože $i + (i+3) = (i+1) + (i+2)$, musí být karta s číslem $i+3$ (pokud $i \leq 97$) v krabici A . Vidíme, že tři po sobě jdoucí čísla karet v různých krabicích určují, v jaké krabici bude karta s následujícím číslem: rozmístění karet do krabic se cyklicky opakuje. Stejný argument můžeme samozřejmě použít i opačným směrem. Stačí tedy přiřadit barvy krabic číslům 1, 2 a 3, což lze učinit šesti způsoby. Dvě karty pak budou ve stejné krabici, právě když jejich čísla dávají při dělení třemi stejný zbytek. Takové rozdělení zřejmě vyhovuje, protože součet dvou čísel s různými zbytky modulo 3 dává třetí možný zbytek.

Předpokládejme nyní, že žádná taková trojice po sobě jdoucích čísel ve třech různých krabicích neexistuje. Označme A krabici, která obsahuje kartu s číslem 1, a necht' i je nejmenší číslo, které v krabici A není,

necht' je v krabici označené B . Nejmenší číslo ve zbývajících třech krabicích C označme k . Z právě uvedených předpokladů plyne, že je $i + 1 < k$.

Předpokládejme, že $k < 100$. Protože $i + k = (i - 1) + (k + 1)$, musí být $k + 1 \in A$. Na druhou stranu z rovnosti $i + (k + 1) = (i + 1) + k$ plyne, že $i + 1 \in C$, a to odporuje předpokladu, že nejmenší číslo v krabici C je k . Je tedy $k = 100$.

Protože $(i - 1) + 100 = i + 99$, je 99 v krabici B . Ukažme, že v takovém případě už musejí být v krabici B všechna čísla mezi 1 a 100: Kdyby nějaké $t > 1$ bylo v krabici A , muselo by být $t - 1$ v krabici C , to však odporuje předchozímu závěru, neboť $t + 99 = (t - 1) + 100$.

Dostali jsme tak další obarvení čísel, které rovněž vyhovuje: je-li součet vybraných čísel nejvýše 100, je zbývajících krabice C ; je-li součet 101, zbývá B , a je-li součet větší než 101, zbývá A . Počet takových obarvení, resp. uspořádání karet do krabic je opět šest. Tím je úloha vyřešena.

Jiné řešení. Jsou dvě možnosti, jak karty rozmístit do krabic: buď dáme do stejné krabice vždy všechny karty, jejichž čísla dávají stejný zbytek při dělení třemi, anebo kartu s číslem 1 dáme do jedné krabice, kartu s číslem 100 do druhé a zbývajících karty s čísly $2, \dots, 99$ do zbylé krabice. Protože krabice se liší barvou, je v obou případech šest možností pro výběr příslušných krabic. Snadno ověříme, že obě popsaná rozdělení karet mají požadované vlastnosti.

Pro $n \geq 3$ označme H_n tvrzení, že jiné rozdělení n karet do tří krabic, při němž popsaný trik funguje, neexistuje. Dokážeme H_n matematickou indukcí. Tvrzení H_3 triviálně platí (obě popsaná rozdělení dokonce splývají). Předpokládejme, že H_n platí pro nějaké $n \geq 3$, a uvažujme $n + 1$ karet (očíslovaných všemi čísly z množiny $\{1, 2, \dots, n + 1\}$).

Pokud je karta s číslem $n + 1$ sama v jedné krabici, zatímco karta s číslem 1 je v krabici ještě s jinými kartami, označme N součet největších čísel v obou zbývajících krabicích, takže $n + 2 \leq N \leq n + (n - 1) = 2n - 1$. To však znamená, že číslo N můžeme dostat i jako součet z jiné dvojice krabic: $N = (n + 1) + (N - n - 1)$. Karta s číslem 1 musí tedy být také samostatně.

Pokud je karta s číslem $n + 1$ v krabici ještě s jiným číslem, dostaneme po jejím odstranění situaci s n kartami, což nemá na fungování triku vliv. Podle indukčního předpokladu jsou možná nejvýše dvě různá rozdělení karet do krabic. Protože $(n + 1) + (n - 2) = n + (n - 1)$, musí být $n + 1$ v krabici s číslem $n - 2$. Je-li n karet rozděleno do krabic podle zbytků modulo 3, bude i karta s číslem $n + 1$ v odpovídající krabici, protože

$n + 1 \equiv n - 2 \pmod{3}$; pokud je n karet (pro $n > 3$) rozděleno do krabic druhým možným způsobem, vyjde nám, že karta s číslem $n + 1$ by musela být zároveň i v krabici s číslem 1, neboť $(n + 1) + 1 = n + 2$, což nejde. Tím je důkaz indukci hotov: ani pro $n + 1$ jiné rozdělení karet do krabic, při němž trik funguje, neexistuje.

Vzhledem k tomu, že krabice jsou rozlišeny třemi barvami, existuje pro každé z uvedených rozdělení šest permutací barev, takže pro každé $n > 3$ existuje $2 \cdot 6 = 12$ různých rozdělení karet, pro něž kouzelníkův trik funguje.

Jiné řešení. Předpokládejme, že máme karty rozmístěny do tří krabic tak, že kouzelníkův trik funguje. Karty s čísly 1 a 2 nemohou být ve stejné krabici: Jestliže je $1, 2, \dots, i - 1 \in A$, karta s číslem $i > 2$ je v krabici B a j je nejmenší číslo karty v krabici C , musí být karta s číslem $j - 1$ v krabici B , protože $(j - 1) + i = j + (i - 1)$. To však nejde, protože součet $1 + j = 2 + (j - 1)$ nedovoluje určit zbývající krabici.

Pro $1 \in A$ a $2 \in B$ označme j nejmenší číslo karty v krabici C . Pokud $j = 3$, dostáváme rozdělení karet do krabic podle zbytku jejich čísla při dělení třemi, pro které trik funguje.

Pokud $j = 100$, plyne z rovnosti $100 + 1 = 99 + 2$, že $99 \in B$, a z rovnosti $k + 99 = 100 + (k - 1)$ pak plyne, že krabice A nemůže obsahovat žádné $k > 1$. Dostáváme tak rozdělení karet do krabic, při němž v krabici B budou všechny karty s čísly $2, 3, \dots, 99$, což je rozmístění, které rovněž vyhovuje.

Kdyby však bylo $3 < j < 100$, vyjde z rovnosti $j + 2 = (j + 1) + 1$, že karta s číslem $j + 1$ je v A , a z rovnosti $j + 3 = (j + 1) + 2$ zas, že karta s číslem 3 je v krabici C . To je ve sporu s definicí čísla j .

Existují tedy jen dvě různá rozmístění karet do krabic, a protože krabice jsou rozlišeny třemi barvami, je celkem $2 \cdot 6 = 12$ různých rozdělení karet, pro něž kouzelníkův trik funguje.

Jiné řešení. Předpokládejme, že v jedné krabici jsou karty s čísly $a_1 < \dots < a_m$, v druhé s čísly $b_1 < \dots < b_n$ a v třetí s čísly $c_1 < \dots < c_p$, přičemž $m \geq 1$, $n \geq 1$, $p \geq 1$.

Uvažujme množiny

$$\begin{aligned} &\{a_1 + b_1, a_1 + b_2, \dots, a_1 + b_n, a_2 + b_n, \dots, a_m + b_n\}, \\ &\{b_1 + c_1, b_1 + c_2, \dots, b_1 + c_p, b_2 + c_p, \dots, b_n + c_p\}, \\ &\{c_1 + a_1, c_1 + a_2, \dots, c_1 + a_m, c_2 + a_m, \dots, c_p + a_m\}, \end{aligned}$$

které mají postupně $m + n - 1$, $n + p - 1$ a $p + m - 1$ různých prvků vždy uspořádaných do rostoucí posloupnosti. Aby kouzelníkův trik fungoval, musejí být uvedené tři množiny navzájem disjunktní, přitom dohromady obsahují $2(m + n + p) - 3 = 197$ různých čísel, z nichž nejmenší je aspoň $1 + 2 = 3$ a největší nejvýše $100 + 99 = 199$, takže jsou to vlastně *všechny* možné součty. Jinými slovy to znamená, že pokud trik funguje, má množina součtů $\{a_i + b_j\}$ právě $m + n - 1$ prvků, množina součtů $\{b_j + c_k\}$ právě $n + p - 1$ prvků a množina součtů $\{c_k + a_i\}$ právě $p + m - 1$ prvků. Bez újmy na obecnosti předpokládejme, že $m \geq n \geq p$.

Pokud $n \geq 2$, uvažujme pro $1 \leq i \leq m - 1$ dvě množiny

$$\{a_1 + b_1, \dots, a_i + b_1, a_{i+1} + b_1, a_{i+1} + b_2, \dots, a_m + b_2, \dots, a_m + b_n\},$$

$$\{a_1 + b_1, \dots, a_i + b_1, a_i + b_2, a_{i+1} + b_2, \dots, a_m + b_2, \dots, a_m + b_n\}.$$

Obě mají právě $m + n - 1$ různých prvků uspořádaných do rostoucí posloupnosti, takže se musejí rovnat, je tudíž

$$a_{i+1} + b_1 = a_i + b_2 \quad \text{neboli} \quad a_{i+1} - a_i = b_2 - b_1.$$

Podobně pro $p \geq 2$ dostaneme

$$c_{i+1} - c_i = b_2 - b_1, \quad 1 \leq i \leq p - 1, \quad b_{i+1} - b_i = a_2 - a_1, \quad 1 \leq i \leq n - 1.$$

To znamená, že v každé z krabic, v níž jsou aspoň dvě karty, tvoří příslušná čísla aritmetickou posloupnost s jednou společnou diferencí $d = a_2 - a_1$. Kdyby bylo $d > 3$, nemohli bychom všechny karty rozmístit do tří krabic. Pokud $d = 3$, musí být karty s čísly 1, 2 a 3 v různých krabicích a vychází, že obsah jednotlivých krabic tvoří karty s čísly 1, 4, 7, ..., 100, 2, 5, 8, ..., 98 a 3, 6, 9, ..., 99.

Je-li $d \leq 2$, nemůže se stát, že by část jedné z možných aritmetických posloupností s diferencí d byla v jedné krabici a část v druhé: jakmile je $c_i - b_j = d = a_2 - a_1$, je $a_1 + c_i = a_2 + b_j$ a trik nemůže fungovat. Vidíme, že pro $d = 2$ by musela být krabice neobsahující 1 nebo 2 prázdná a podobně i pro $d = 1$.

Zůstává ještě možnost $n = p = 1$, $m = 98$. Nechť $b_1 < c_1$ a označme $d = c_1 - b_1$. Kdyby bylo $d < 99$, pak určitě existují $a_i < a_j$ tak, že $a_j - a_i = d = c_1 - b_1$ a trik by nemohl fungovat. Musí tudíž být $c_1 - b_1 = 99$, neboli $b = 1$, $c_1 = 100$ a v první krabici jsou všechna čísla 2, 3, ..., 99. Snadno ověříme, že při takovém uspořádání karet bude kouzelníkův trik fungovat.

Našli jsme dvě principiálně odlišná rozdělení karet do nerozlišených krabic, a protože barvy krabic můžeme permutovat šesti způsoby, je počet různých rozdělení karet celkem 12.

5. Nejprve ukážeme, že pro každé přirozené a platí

$$3^a \mid 2^{3^a} + 1. \quad (1)$$

Pro $a = 1$ je $3 \mid 2^3 + 1 = 9$; předpokládejme, že pro nějaké přirozené k platí $3^k \mid 2^{3^k} + 1$. Protože

$$2^{3^{k+1}} + 1 = (2^{3^k} + 1)(2^{2 \cdot 3^k} - 2^{3^k} + 1),$$

stačí ukázat, že výraz v druhé závorce je dělitelný třemi. Je ovšem $2^{2 \cdot 3^k} - 2^{3^k} + 1 \equiv 1 - (-1) + 1 \equiv 0 \pmod{3}$, takže $3^{k+1} \mid 2^{3^{k+1}} + 1$ a dle principu matematické indukce platí (1) pro každé $a \in \mathbb{N}$.

Číslo $2^{3^a} + 1$ může ovšem mít i další dělitele, dokonce libovolný počet různých prvočinitelů. Najdeme-li tedy a tak, že $2^{3^a} + 1$ má dalších 1 999 různých prvočinitelů $3 < p_1 < \dots < p_{1999}$, bude řešením úlohy například číslo $n = 3^a p_1 p_2 \dots p_{1999}$, které má právě 2 000 různých prvočinitelů a dělí číslo $2^n + 1$, neboť dělí jeho dělitele $2^{3^a} + 1$:

$$2^{3^a} + 1 \mid (2^{3^a})^{p_1 \dots p_{1999}} + 1 = 2^n + 1.$$

Ukažme, že číslo $2^{3^a} + 1$ má pro dostatečně velké a dostatečný počet různých prvočinitelů. Vyjdeme opět z rozkladu

$$m^3 + 1 = (m + 1)(m^2 - m + 1).$$

Protože je zároveň

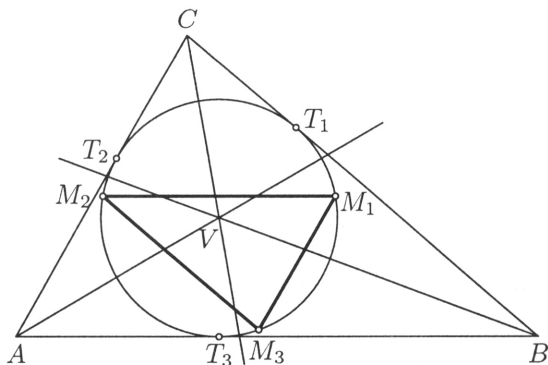
$$m^2 - m + 1 = (m + 1)(m - 2) + 3,$$

je největší společný dělitel čísel $m + 1$ a $m^2 - m + 1$ dělitelem čísla 3. Přitom je-li číslo $m + 1$ dělitelné třemi, je i $m - 2$ dělitelné třemi, takže $(m + 1)(m - 2) + 3$ je dělitelné třemi, ale ne devíti. Pro $m > 2$ tak musí mít $m^2 - m + 1$ alespoň jednoho prvočinitele $p > 3$, který nedělí $m + 1$. Z rozkladu

$$2^{3^{a+1}} + 1 = (2^{3^a})^3 + 1 = (2^{3^a} + 1)(2^{2 \cdot 3^a} - 2^{3^a} + 1)$$

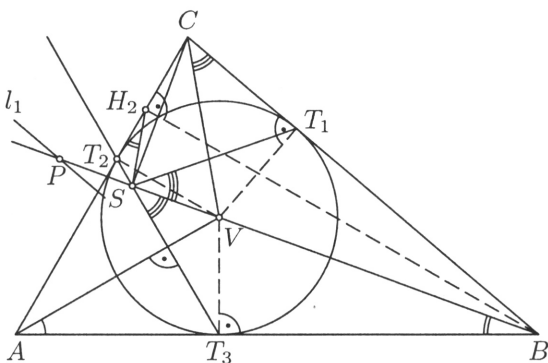
proto plyne, že číslo $2^{3^{a+1}} + 1$ má přinejmenším aspoň jednoho dalšího prvočinitele, který není dělitelem čísla $2^{3^a} + 1$. Pro $a > 1999$ tak bude mít číslo $2^{3^a} + 1$ aspoň 2 000 různých prvočinitelů. To je vše, co jsme potřebovali dokázat.

6. Označme V střed vepsané kružnice a M_1, M_2, M_3 po řadě body souměrné s body T_1, T_2, T_3 podle příslušných os úhlů AV, BV a CV . Body M_1, M_2, M_3 (obr. 67) tudíž leží na kružnici vepsané trojúhelníku ABC . Ukážeme, že to jsou vrcholy trojúhelníku tvořeného přímkami l_1, l_2, l_3 .



Obr. 67

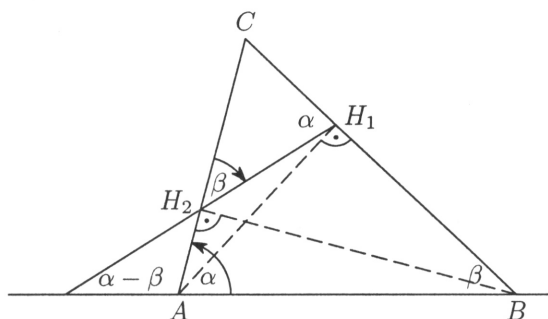
S ohledem na symetrii celé situace stačí, když dokážeme, že bod M_2 leží na přímce l_1 . Body T_2 a H_2 leží vždy v téže polorovině určené přímkou BV . Budeme se zabývat pouze případem, kdy vrchol C trojúhelníku ABC leží ve stejné polorovině (druhý případ se vyřeší analogicky). Označme S průsečík polopřímky BV se spojnici T_2T_3 a P průsečík polo-



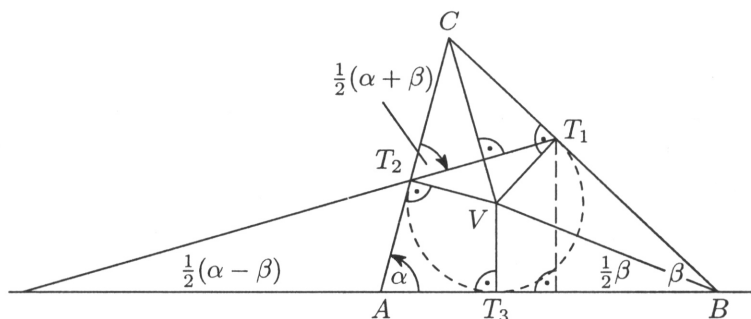
Obr. 68

přímky BV s přímkou l_1 (obr. 68). Protože $|T_1S| = |T_3S|$ a $|\sphericalangle VST_3| = |\sphericalangle AT_3T_2| - \frac{1}{2}\beta = (90^\circ - \frac{1}{2}\alpha) - \frac{1}{2}\beta = \frac{1}{2}\gamma$, je $|\sphericalangle VST_1| = |\sphericalangle VCT_1| = \frac{1}{2}\gamma$.

Podobně zjistíme, že i přímky l_1 a l_2 jsou rovnoběžné s odpovídajícími stranami trojúhelníku ABC .



Obr. 70



Obr. 71

Jiné řešení. Jak jsme ukázali na závěr předchozího řešení, jsou přímky l_1 , l_2 , l_3 jsou rovnoběžné s odpovídajícími stranami trojúhelníku ABC . To ovšem znamená, že existuje stejnolehlost h , která převádí trojúhelník $A'B'C'$ určený přímkami l_1 , l_2 , l_3 na trojúhelník ABC . Abychom dokázali, že trojúhelník $A'B'C'$ je vepsán kružnici vepsané danému trojúhelníku ABC , musíme ukázat, že obrazem této kružnice ve stejnolehlosti h je právě kružnice trojúhelníku ABC opsaná. Přitom je zřejmé, že příslušný koeficient podobnosti musí být r/R , kde r značí poloměr vepsané a R poloměr opsané kružnice trojúhelníku ABC .

Označme O střed kružnice opsané trojúhelníku ABC a uvažujme bod J , který dělí úsečku VO v uvedeném poměru, tj. $|VJ| : |JO| = r : R$, a stejnolehlost $h(J, -r/R)$. Ukážeme, že trojúhelník $A'B'C'$, který je obrazem trojúhelníku ABC ve stejnolehlosti h , je trojúhelník tvořený

přímkami l_1 , l_2 a l_3 . Vzhledem k symetrii prvků v trojúhelníku stačí ukázat, že bod T_1 leží na ose souměrnosti přímek H_1H_2 a $A'B'$, tj. že vzdálenost bodu T_1 od přímky H_1H_2 je stejná jako od přímky $A'B'$. Protože analogicky i vzdálenost bodu T_2 od obou přímek $A'B'$ a H_1H_2 je stejná, bude to znamenat, že přímka $A'B'$ je obrazem přímky H_1H_2 v souměrnosti podle přímky T_1T_2 , takže $l_3 = A'B'$.

Nejdříve zjistíme vzdálenost obou rovnoběžek AB a $A'B'$. Protože vzdálenost bodu O od strany AB je $R \cos \gamma$, plyne ze stejnolehlosti h , že vzdálenost bodu V od strany $A'B'$ je $\frac{r}{R} \cdot R \cos \gamma = r \cos \gamma$. Vzdálenost AB a $A'B'$ je tedy $r(1 + \cos \gamma) = 2r \cos^2 \frac{1}{2} \gamma$.

Vzdálenost bodu T_1 od strany AB je (obr. 71) $|T_1B| \sin \beta = 2|T_1B| \times \sin \frac{1}{2} \beta \cos \frac{1}{2} \beta = 2r \cos^2 \frac{1}{2} \beta$, protože $|T_1B| = |T_3B| = r \cotg \frac{1}{2} \beta$. Odečtením od vzájemné vzdálenosti přímek AB a $A'B'$ vychází, že vzdálenost bodu T_1 od $A'B'$ je

$$2r \left| \cos^2 \frac{\gamma}{2} - \cos^2 \frac{\beta}{2} \right| = r |\cos \gamma - \cos \beta|. \quad (1)$$

Zbývá spočítat vzdálenost bodu T_1 od přímky H_1H_2 . Z tětiového čtyřúhelníku ABH_1H_2 (obr. 70) máme $|\sphericalangle H_2H_1C| = \alpha$, takže příslušná vzdálenost je

$$|H_1T_1| \sin \alpha = ||CH_1| - |CT_1|| \sin \alpha = \sin \alpha \left| b \cos \gamma - r \cotg \frac{\gamma}{2} \right|,$$

přičemž $b = r(\cotg \frac{1}{2} \alpha + \cotg \frac{1}{2} \gamma)$, takže

$$\begin{aligned} |H_1T_1| \sin \alpha &= r \sin \alpha \left| \cos \gamma \left(\cotg \frac{\alpha}{2} + \cotg \frac{\gamma}{2} \right) - \cotg \frac{\gamma}{2} \right| = \\ &= r \sin \alpha \left| \cotg \frac{\alpha}{2} \cos \gamma - \cotg \frac{\gamma}{2} (1 - \cos \gamma) \right| = \\ &= r \left| \sin \alpha \cotg \frac{\alpha}{2} \cos \gamma - \sin \alpha \cotg \frac{\gamma}{2} \cdot 2 \sin^2 \frac{\gamma}{2} \right| = \\ &= r \left| 2 \sin \frac{\alpha}{2} \cos \frac{\alpha}{2} \cotg \frac{\alpha}{2} \cos \gamma - 2 \sin \alpha \cos \frac{\gamma}{2} \sin \frac{\gamma}{2} \right| = \\ &= r \left| 2 \cos^2 \frac{\alpha}{2} \cos \gamma - \sin \alpha \sin \gamma \right| = \\ &= r |(1 + \cos \alpha) \cos \gamma - \sin \alpha \sin \gamma| = \\ &= r |\cos \gamma + \cos(\alpha + \gamma)| = r |\cos \gamma - \cos \beta|. \end{aligned}$$

Zjistili jsme, že vzdálenost bodu T_1 od přímky H_1H_2 je stejná jako vzdálenost (1) téhož bodu od přímky $A'B'$. Tím je tvrzení úlohy dokázáno.

Středoevropská olympiáda v informatice (CEOI) je soutěží středoškolských studentů v návrhu a implementaci programů řešících zadané algoritmicky netriviální problémy. CEOI se od roku 1994 koná vždy v některé ze středoevropských zemí, členů CEOI (ČR, Chorvatsko, Maďarsko, Německo, Polsko, Rumunsko, Slovensko, Slovinsko, zcela pasivním členem zůstává Rakousko). Smyslem založení CEOI bylo podpořit přípravu reprezentací středoevropských zemí pro mezinárodní olympiádu v informatice (IOI). Kromě čtyřčlenných týmů členských zemí se na pozvání pořadatelské země mohou účastnit další týmy, jak domácí, tak zahraniční.

CEOI je soutěží jednotlivců, pořadí týmů není vyhlašováno. Soutěžící řeší ve dvou kolech vždy po třech problémech. Výsledkem musí být funkční program řešící zadaný problém pro přípustná vstupní data. Veškeré vyhodnocení je prováděno automatizovaně, bez subjektivních vlivů. O bodových hodnoceních jednotlivých úloh rozhoduje scientific committee daného ročníku soutěže.

Místem konání letošní CEOI byla rumunská Cluj-Napoca. Soutěž proběhla ve dnech 24.–31. srpna 2000. Organizací bylo pověřen Krajský školní inspektorát Cluj a Univerzita Babes-Bolyai Cluj. Sponzorsky se na soutěži podílela dlouhá řada firem především z oblasti IT1, generálním sponzorem byla společnost Computer Press Agora Tirgu Mures. Soutěžním i ubytovacím místem reprezentačních týmů byl hotel Transylvania Cluj. Letošního ročníku se kromě členských zemí účastnily na pozvání týmy Nizozemí, Moldávie a USA. Kromě toho Rumunsko využilo svého práva a přizvalo i své týmy „2“ a „3“.

Reprezentanti pro CEOI se každoročně vybírají podle výsledků celostátního kola MO kategorie P a to následovně: jako reprezentanti pro mezinárodní olympiádu v informatice (IOI) se vyberou čtyři nejlepší soutěžící, a následující čtyři na výsledkové listině se stanou reprezentanty ČR na CEOI, pokud mají před sebou ještě alespoň rok studia. Mají tedy možnost zkušenosti nabyté na CEOI uplatnit v následujícím ročníku IOI. Tak je naplněn jeden z účelů CEOI, tj. příprava pro IOI.

Reprezentační týmy jsou vedeny vedoucím a jeho zástupcem (team leader, deputy leader), kteří zajišťují schválení, resp. výběr soutěžních úloh a jejich překlad do národního jazyka. Kromě toho překládají zpět do oficiálního jazyka soutěže, tj. do angličtiny případné dotazy řešitelů a samozřejmě tvoří organizační zázemí pro soutěžní tým.

Letošního ročníku se jako reprezentanti ČR účastnili:

Martin Beránek z Gymnázia Praha 4, Ohradní

Pavel Čížek z Gymnázia Ch. Dopplera, Praha 5

Roman Krejčík z Gymnázia Ch. Dopplera, Praha 5

Miloslav Trmač z Biskupského gymnázia Brno, Barvičova

Vedoucím týmu byl jmenován RNDr. *Tomáš Pitner*, Dr., odborný asistent Fakulty informatiky Masarykovy univerzity v Brně. Roli zástupce vedoucího plnila RNDr. *Miroslava Kozubíková*, učitelka gymnázia na třídě Kpt. Jaroše 14, Brno

Výsledky české reprezentace na CEOI 2000 bohužel nebyly příliš uspokojivé. V letošním ročníku nezískal žádný z našich soutěžících medaili. Úlohy byly nadprůměrně obtížné a zaměřené do oblastí, které běžná středoškolská výuka informatiky v žádném případě nepokrývá.

12. mezinárodní olympiáda v informatice

Tento ročník mezinárodní olympiády v informatice (The International Olympiad in Informatics) se konal ve dnech 23.–30. 9. 2000 v Pekingu v Číně. Olympiády se zúčastnilo 276 soutěžících ze 70 zemí. Soutěž IOI získává v celém světě stále větší zájem a popularitu, počet účastníků se každoročně zvyšuje. Ještě před deseti lety v ní soutěžili studenti jenom z asi dvaceti zemí, pro srovnání loni bylo na IOI 250 soutěžících z 65 zemí.



Olympiáda se konala v konferenčním středisku umístěném v severní části Pekingu v těsném sousedství hotelů Grand Continental a Catic, v nichž bylo zajištěno ubytování a stravování všech účastníků. Pro zajímavost můžeme uvést, že nedaleko se nachází rozsáhlý sportovní komplex a že s těmito hotely, konferenčním střediskem a sportovišti se počítá i pro případ, že bude Číně svěřeno uspořádání letních olympijských her v roce 2008. Celá akce byla organizačně i technicky výborně připravena, vlastní soutěž i vyhodnocování proběhlo zcela bez závad a bez problémů. Pro všechny účastníky byl připraven i velmi zajímavý doprovodný program. Měli jsme možnost navštívit nejvýznamnější historické památky nacházející se v Pekingu a jeho okolí — náměstí Tian'anmen, areály historických císařských paláců Forbidden City a Summer Palace, park s chrámy Temple of Heaven a také slavnou velkou čínskou zeď.

Soutěž byla jako vždy rozdělena do dvou soutěžních dnů, v každém z nich řešili studenti tři úlohy. Řešení úloh v IOI probíhá u počítačů podobným způsobem jako v praktické části celostátního kola kategorie P naší matematické olympiády. Každý soutěžící má přidělen svůj osobní počítač a na práci v každém soutěžním dnu má k dispozici omezený čas 5 hodin. Soutěžní den je vždy zakončen testováním odevzdaných programů, při kterém se sleduje nejen správnost výpočtu, ale pomocí časových limitů také kvalita vytvořených programů (tzn. jejich časové a paměťové nároky). Pro každou úlohu bylo stanoveno maximální možné ohodnocení 100 bodů, každý soutěžící navíc obdržel v každém ze soutěžních dnů

bonus 50 bodů. Celkové pořadí bylo dáno součtem hodnocení všech šesti úloh a obou bonusů, maximálně tedy bylo možné získat 700 bodů.

Slavnostního zakončení olympiády spojeného s vyhlášením výsledků a předáním medailí nejlepším řešitelům se zúčastnila celá řada významných státních představitelů, zástupci čínských odborných informatických společností a organizací a zástupci sponzorů. Mezinárodní olympiáda v informatice je výhradně soutěží jednotlivců a žádné oficiální pořadí družstev v ní podle pravidel IOI není vyhlášováno. Celkem bylo v soutěži uděleno 23 zlatých medailí, 47 stříbrných a 69 bronzových medailí. Počet udělených medailí se stanoví v IOI podle pravidla, že medaili získá přibližně polovina soutěžících, přičemž zlaté, stříbrné a bronzové medaile se rozdělují v poměru 1 : 2 : 3. Drobné odchylky od tohoto pravidla jsou způsobeny tím, že více soutěžících může dosáhnout stejného bodového zisku (udělených medailí pak může být o něco více).

Členové reprezentačního družstva z České republiky byli vybráni na základě výsledků celostátního kola kategorie P 49. ročníku Matematické olympiády. Na IOI 2000 nás reprezentovalo družstvo ve složení *Jakub Bystroň* (absolvent Gymnázia Karviná), *Pavel Charvát* (absolvent Gymnázia Ohradní v Praze 4), *Ondřej Rucký* (absolvent Gymnázia na Mikulášském nám. v Plzni) a *Jiří Svoboda* (student Gymnázia Ch. Dopplera v Praze 5). Vzhledem k podzimnímu termínu konání soutěže tři z našich reprezentantů již ukončili své studium na gymnáziu a v současné době jsou Jakub Bystroň a Pavel Charvát posluchači 1. ročníku oboru informatika na Matematicko-fyzikální fakultě Univerzity Karlovy v Praze, Ondřej Rucký nyní studuje v 1. ročníku na Západočeské univerzitě v Plzni a pouze Jiří Svoboda je nadále studentem gymnázia. Vedením české delegace byl pověřen místopředseda Ústředního výboru Matematické olympiády pro kategorii P doc. RNDr. *Pavel Töpfer*, CSc., z Matematicko-fyzikální fakulty Univerzity Karlovy v Praze, druhým vedoucím byl *Daniel Král* z téže fakulty.

V rámci přípravy na soutěž se naši studenti zúčastnili na konci srpna 2000 týdenního přípravného soustředění v Polsku, které bylo společné pro týmy vybrané na IOI ze Slovenska, z Polska a z České republiky. Soustředění organizačně a finančně zajistili kolegové pečující o programátorskou olympiádu v Polsku, na odborném programu se podíleli rovným dílem vedoucí ze všech tří zúčastněných zemí. Naši reprezentanti nebyli letos tak úspěšní, jako tomu bylo dosud snad ve všech předchozích ročnících této soutěže. Dosažené výsledky však přesto stále patří v celosvětovém měřítku mezi nadprůměrné. Pavel Charvát, Jakub Bystroň a Jiří Svo-

boda získali bronzové medaile, na Ondřeje Ruckého tentokrát medaile nezbyla.

Výsledky našich studentů:

79.	Pavel Charvát	380 bodů	bronzová
102.	Jakub Bystron	320 bodů	bronzová
108.	Jiří Svoboda	300 bodů	bronzová
	Ondřej Rucký	210 bodů	–

Texty soutěžních úloh

1. Parkoviště

Parkoviště u Velké zdi je tvořeno dlouhou řadou parkovacích míst. Jeden konec řady budeme považovat za její levý konec a druhý za její pravý konec. Všechna parkovací místa jsou obsazena. Každý zaparkovaný vůz je určitého typu, několik různých vozů může být stejného typu. Typy vozů jsou označeny celými čísly. Několik nudících se dělníků se rozhodlo přeuspořádat zaparkované vozy podle jejich typů, a to v rostoucím pořadí zleva doprava. Dělníci chtějí použít následující postup složený z několika po sobě následujících kroků: v každém kroku vyjedou současně několika vozy z jejich parkovacích míst a zaparkují je na místa, která se takto uvolnila. Každý dělník může přeparkovat v jednom kroku nejvýše jeden vůz. V jednom kroku lze tedy přeparkovat nejvýše tolik vozů, kolik je dělníků. Aby se příliš nenadřeli, chtějí dělníci svůj záměr uskutečnit během co nejmenšího počtu kroků.

Nechť N je počet vozů a W je počet dělníků. Vaším úkolem je vytvořit program, který na vstupu obdrží popis typů zaparkovaných vozů a nalezne způsob, jak vozy přeparkovat během nejvýše $\lceil N/(W-1) \rceil$ kroků, kde $\lceil N/(W-1) \rceil$ je číslo $N/(W-1)$ zaokrouhlené nahoru na nejbližší celé číslo. Počet kroků, které jsou potřeba k přeparkování vozů, je vždy nejvýše $\lceil N/(W-1) \rceil$.

Podívejme se na následující příklad: Na parkovišti je 10 vozů; jejich typy jsou označeny čísly 1, 2, 3 a 4. Čtyři dělníci chtějí tyto vozy přeparkovat. Původní rozmístění vozů je následující:

2 3 3 4 4 2 1 1 3 1

Minimální možný počet kroků je tři a rozmístění vozů po jednotlivých krocích může být například následující:

2 1 1 4 4 2 3 3 1 — po prvním kroku,

2 1 1 2 4 3 3 3 4 1 — po druhém kroku,

1 1 1 2 2 3 3 3 4 4 — po třetím kroku.

Vstup: Vstupní soubor se jmenuje CAR.IN. Jeho první řádek obsahuje tři celá čísla. První z nich, N , je počet vozů na parkovišti ($2 \leq N \leq 20\,000$) a druhé z nich, M , je počet typů vozů ($2 \leq M \leq 50$). Jednotlivé typy vozů jsou označeny čísly 1 až M . Mezi zaparkovanými vozy se nachází od každého z těchto typů alespoň jeden. Třetí číslo, W , určuje počet dělníků, kteří chtějí vozy přeparkovat ($2 \leq W \leq M$). Na druhém řádku se nachází N čísel, která určují počáteční rozmístění vozů — i -té číslo představuje typ i -tého vozu v řadě počítáno zleva doprava.

Výstup: Výstupní soubor se jmenuje CAR.OUT. První řádek výstupního souboru obsahuje jedno celé číslo R , představující počet kroků nalezeného řešení úlohy. Následujících R řádků popisuje po řadě jednotlivé kroky. První číslo, C , na každém z těchto řádků určuje počet vozů přeparkovávaných v příslušném kroku. Za tímto číslem následuje dalších $2C$ celých čísel, která popisují pozice vozů. Parkovací místa jsou očíslována zleva doprava od 1 do N . První dvojice těchto čísel popisuje přesun jednoho z vozů: první číslo z dvojice udává pozici vozu před tímto krokem a druhé po tomto kroku. Další dvojice čísel určuje, jak bude přeparkován další z vozů, atd. Pokud existuje více možných řešení, vaším úkolem je nalézt a vypsat jedno libovolné z nich.

Příklady vstupů a výstupů:

CAR.IN

10 4 4

2 3 3 4 4 2 1 1 3 1

CAR.OUT

3

4 2 7 3 8 7 2 8 3

3 4 9 9 6 6 4

3 1 5 5 10 10 1

Hodnocení. Předpokládejme, že váš program našel způsob, jak vozy přeparkovat v R krocích. Označme si $\lceil N/(W-1) \rceil$ jako Q . Jestliže nalezený postup přesunu vozů nespňuje zadání úlohy, je příslušný test hodnocen 0 body. Jinak je ohodnocen body podle následujících pravidel:

$R \leq Q$ 100 % bodů

$R = Q + 1$ 50 % bodů

$R = Q + 2$ 20 % bodů

$R \geq Q + 3$ 0 % bodů

2. Palindromy

Palindromem nazýváme takový znakový řetězec, který je symetrický, tj. který se čte stejně zleva i zprava. Vytvořte program, jenž pro zadaný řetězec určí minimální počet znaků, které je nutné do tohoto řetězce vložit tak, aby se z něj stal palindrom.

Například vložením dvou znaků do řetězce „Ab3bd“ můžeme získat palindrom „dAb3bAd“ nebo „Adb3bdA“. Vložením méně než dvou znaků však z tohoto řetězce palindrom nelze vytvořit.

Vstup: Vstupní soubor se jmenuje PALIN.IN. První řádek vstupního souboru obsahuje jedno celé číslo N , délku vstupního řetězce, $3 \leq N \leq 5\,000$. Druhý řádek obsahuje vstupní řetězec. Řetězec se skládá z velkých písmen od »A« do »Z«, z malých písmen od »a« do »z« a z číslic od »0« do »9«. Velká a malá písmena v řetězci představují odlišné znaky.

Výstup: Výstupní soubor se jmenuje PALIN.OUT. První řádek obsahuje jedno číslo — nalezený minimální počet znaků, které je do řetězce třeba vložit.

<i>Příklady vstupů a výstupů:</i>	PALIN.IN	PALIN.OUT
	5	2
	Ab3b	

3. Střední tvrdost

V rámci jednoho z výzkumných projektů je třeba otestovat N vzorků materiálů. Vzorky jsou označeny čísly od 1 do N , kde N je liché číslo. Tvrdost každého ze vzorků lze vyjádřit celým číslem Y ($1 \leq Y \leq N$). Žádné dva vzorky nemají stejnou tvrdost. Vzorek X je vzorek se střední tvrdostí, jestliže stejný počet vzorků má tvrdost menší než vzorek X a stejný počet vzorků má tvrdost větší. Vaším úkolem je vytvořit program, který určí vzorek se střední tvrdostí.

Jediný přístroj, který je v laboratoři k dispozici, je schopný určit ze tří předložených vzorků ten, který má mezi těmito vzorky střední tvrdost.

Popis knihovny. Obdrželi jste knihovnu pojmenovanou *device*, která obsahuje následující tři rutiny:

- ▷ Rutina *GetN* musí být zavolána právě jednou, a to na začátku výpočtu; volá se bez parametrů a její návratovou hodnotou je číslo N .
- ▷ Rutina *Med3* je volána se třemi parametry, jimiž jsou čísla tří navzájem různých vzorků vložených do přístroje. Její návratová hodnota je číslo vzorku, který má střední tvrdost mezi předloženými vzorky.

▷ Rutina `Answer` musí být zavolána právě jednou, a to na konci výpočtu; jejím jediným parametrem je vaším programem určené číslo vzorku X , tj. číslo vzorku se střední tvrdostí. Zavolání této rutiny zároveň korektně ukončí výpočet programu.

Knihovna `device` vytvoří dva soubory: `MEDIAN.OUT` a `MEDIAN.LOG`. První řádek souboru `MEDIAN.OUT` obsahuje jedno číslo, které je rovno číslu vzorku předanému vaším programem rutině `Answer`. Druhý řádek obsahuje jedno celé číslo představující počet volání rutiny `Med3` vaším programem. Komunikace mezi vaším programem a knihovnou je zaznamenána v souboru `MEDIAN.LOG`.

Pokyny pro programátory v jazyce Pascal: Vložte do vašeho programu následující řádek:

```
uses device;
```

Pokyny pro programátory v jazyce C/C++: Vložte do vašeho programu následující řádek:

```
#include "device.h"
```

Dále vytvořte projekt `MEDIAN.PRJ` a vložte do tohoto projektu soubory `MEDIAN.C` (`MEDIAN.CPP`) a `DEVICE.OBJ`.

Ladění. Komunikaci programu s poskytnutou knihovnou lze ladit následujícím způsobem: Vytvoříte textový soubor `DEVICE.IN`. Tento soubor by měl obsahovat dva řádky. Na prvním z nich bude uvedeno jedno celé číslo — počet vzorků N . Druhý řádek obsahuje N celých čísel od 1 do N ve vámi zvoleném pořadí; i -té z nich představuje tvrdost i -tého vzorku.

Příklad vstupu: `DEVICE.IN`

```
5
2 5 4 3 1
```

Tento soubor `DEVICE.IN` obsahuje popis následujících pěti vzorků:

Vzorek	1	2	3	4	5
Tvrdost	2	5	4	3	1

Korektní posloupnost komunikace vašeho programu s knihovnou by mohla vypadat následovně:

1. `GetN` (v Pascalu) nebo `GetN()` (v C/C++) — návratová hodnota je 5.
2. `Med3(1,2,3)` — návratová hodnota je 3.
3. `Med3(3,4,1)` — návratová hodnota je 4.
4. `Med3(4,2,5)` — návratová hodnota je 4.
5. `Answer(4)`

Omezení specifická pro tuto úlohu

- ▷ Počet vzorků N splňuje tyto podmínky: $5 \leq N \leq 1\,499$ a N je liché.
- ▷ Číslo vzorku i splňuje $1 \leq i \leq N$.
- ▷ Tvrdost vzorku Y splňuje $1 \leq Y \leq N$; všechny vzorky mají navzájem různou tvrdost.
- ▷ Jméno knihovny v Pascalu: `device.tpu`
- ▷ Deklarace knihovních funkcí a procedur v Pascalu:

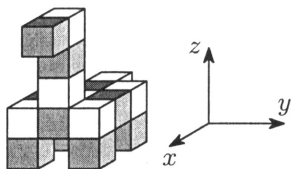
```
function GetN: integer;  
function Med3(x,y,z:integer):integer;  
procedure Answer(m:integer);
```
- ▷ Jméno knihovny v C/C++: `device.h`, `device.obj` (při kompilaci použijte `LARGE` memory model)
- ▷ Deklarace funkcí v C/C++:

```
int GetN(void);  
int Med3(int x, int y, int z);  
void Answer(int m);
```
- ▷ Váš program smí provést nejvýše 7 777 volání funkce `Med3` během jednoho testu.
- ▷ Váš program nesmí číst ani zapisovat do žádných souborů.

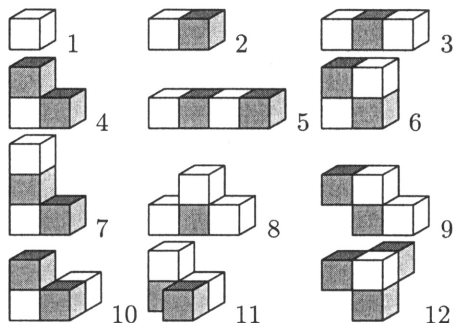
4. Stavby z kostek

Jednotková krychle je krychle o rozměrech $1 \times 1 \times 1$ taková, že všechny její rohy mají celočíselné souřadnice x , y a z . Dvě jednotkové krychle se dotýkají, pokud mají společnou jednu ze svých stěn. Dvě krychle jsou spojeny, pokud existuje posloupnost jednotkových krychlí taková, že každé dvě po sobě následující krychle se dotýkají a uvažované krychle jsou první a poslední v této posloupnosti. Stavbou z kostek rozumíme libovolnou množinu jednotkových krychlí takovou, že každé dvě jednotkové krychle v této množině jsou spojené (obr. 72). Objemem stavby z kostek nazýváme počet jednotkových krychlí, které tato stavba obsahuje. Kostkou rozumíme množinu nejvýše čtyř navzájem spojených jednotkových krychlí. Existuje 12 různých druhů kostek, které jsou znázorněny na obr. 73. Kostky kteréhokoliv druhu mohou být použity ve stavbě v libovolném množství a to jakkoliv posunuté a natočené (ne však zrcadlově převrácené). Barevné odstíny na obou obrázcích jsou použity pouze ke zvýšení jejich přehlednosti a nemají žádný další význam.

Množinu D kostek lze použít pro vytvoření stavby S , pokud lze všechny kostky z množiny D umístit a natočit tak, aby vytvořily přesně stavbu S a přitom byly navzájem disjunktní (mohou se však dotýkat).



Obr. 72. Kůň z kostek



Obr. 73. 12 různých druhů kostek

Vytvořte program, který obdrží na vstupu popis všech 12 druhů kostek a stavby S . Program poté určí nejmenší možný počet kostek, které jsou potřeba k vytvoření stavby S , a dále určí, jaké kostky budou ve stavbě použity.

Vstup: Ve vstupních souborech se na všechny jednotkové krychle odkazujeme pomocí trojice souřadnic x , y a z z toho jejich rohu, pro který je $x + y + z$ nejmenší.

Vstupní soubor, který popisuje jednotlivé druhy kostek, se jmenuje `TYPES.IN`. Obsah tohoto souboru je stejný pro všechny testy. Obsahuje popis 12 druhů kostek; všechny tyto kostky jsou znázorněny na obr. 73. Popis druhů kostek obsažený v tomto souboru je uspořádaný podle čísel druhů jednotlivých kostek. Každý druh kostek je popsán na několika řádcích. První řádek popisu jednoho druhu kostky obsahuje jediné celé číslo I , které je jeho identifikátorem ($1 \leq I \leq 12$). Druhý řádek obsahuje jediné celé číslo V , které představuje objem kostky popisovaného druhu ($1 \leq V \leq 4$). Každý z následujících V řádků pak obsahuje trojici celých čísel x , y , z , která určují polohu jednotlivých jednotkových krychlí tvořících kostku ($1 \leq x, y, z \leq 4$).

Vstupní soubor, který popisuje zadanou stavbu z kostek, se jmenuje `BLOCK.IN`. Jeho první řádek obsahuje jediné celé číslo V , jež představuje celkový objem stavby ($1 \leq V \leq 50$). Každý z následujících V řádků pak obsahuje trojici celých čísel x , y , z , která určují polohu jednotlivých jednotkových krychlí tvořících stavbu ($1 \leq x, y, z \leq 7$).

Výstup: Výstupní soubor se jmenuje `BLOCK.OUT`. První řádek obsahuje jediné celé číslo M , které představuje minimální počet kostek, které jsou potřeba pro vytvoření zadané stavby. Druhý řádek obsahuje M iden-

tifikátorů druhů kostek, ze kterých je možné stavbu sestavit. Pokud existuje více optimálních řešení, váš program vypíše právě jedno z nich.

Příklady vstupů a výstupů:

TYPES.IN			BLOCK.IN
1	6	10	18
1	4	4	2 1 1
1 1 1	1 1 1	2 1 1	4 1 1
2	1 2 1	1 2 1	2 3 1
2	1 1 2	2 2 1	4 3 1
1 1 1	1 2 2	2 1 2	2 1 2
1 2 1	7	11	3 1 2
3	4	4	4 1 2
3	1 1 1	1 1 1	1 2 2
1 1 1	1 2 1	1 2 1	2 2 2
1 2 1	1 1 2	2 2 1	3 2 2
1 3 1	1 1 3	1 1 2	4 2 2
4	8	12	2 3 2
3	4	4	3 3 2
1 1 1	1 1 1	2 2 1	4 3 2
1 2 1	1 2 1	2 1 2	4 2 3
1 1 2	1 3 1	1 2 2	4 2 4
5	1 2 2	2 2 2	4 2 5
4	9		5 2 5
1 1 1	4		
1 2 1	1 2 1		
1 3 1	1 3 1		
1 4 1	1 1 2		
	1 2 2		
			BLOCK.OUT
			5
			7 10 2 10 12

Poznámka.

1. Vstupní soubor BLOCK.IN popisuje stavbu koně z obr. 72.
2. Jiná přípustná řešení mohou mít ve výstupním souboru na druhém řádku následující identifikátory druhů kostek:

2 7 10 11 12
2 7 11 11 12
4 4 7 10 11
4 4 9 10 11

5. Poštovní úřady

V jednom nejmenovaném státě se nachází jedna dlouhá rovná dálnice s několika vesnicemi podél ní. Do každé vesnice vede jeden výjezd z dálnice; všechny tyto výjezdy jsou očíslovány celými kladnými čísly. Číslo výjezdu je rovno jeho vzdálenosti od začátku dálnice. U žádného z výjezdů neleží dvě různé vesnice. Vzdálenost mezi dvěma vesnicemi je rovna absolutní hodnotě rozdílu čísel výjezdů, které vedou do těchto vesnic (zanedbáváme tedy vzdálenost vesnic od dálnice).

V některých z těchto vesnic (ne nutně ve všech) mají být zřízeny poštovní úřady. Ředitelství pošt chce zřídit poštovní úřady tak, aby součet dojezdových vzdáleností všech vesnic byl co nejmenší. Dojezdovou vzdáleností vesnice rozumíme její vzdálenost od poštovního úřadu, který je k ní nejbližší.

Vytvořte program, který obdrží na vstupu zadána čísla jednotlivých výjezdů, u kterých leží vesnice, a počet poštovních úřadů, které plánuje ředitelství pošt zřídit. Program poté určí minimální možný součet dojezdových vzdáleností všech vesnic a nalezne rozmístění poštovních úřadů, které dosahuje tohoto součtu.

Vstup: Vstupní soubor se jmenuje `POST.IN`. Jeho první řádek obsahuje dvě celá kladná čísla: první z nich, V , představuje počet vesnic podél dálnice ($1 \leq V \leq 300$) a druhé z nich, P , představuje počet poštovních úřadů, které se plánují vybudovat ($1 \leq P \leq 30$, $P \leq V$). Druhý řádek obsahuje rostoucí posloupnost V celých kladných čísel. Těchto V čísel představuje čísla výjezdů z dálnice u jednotlivých vesnic. Pro každé z těchto čísel, označme ho X , platí následující omezení: $1 \leq X \leq 10\,000$.

Výstup: Výstupní soubor se jmenuje `POST.OUT`. První řádek obsahuje jediné celé číslo, které představuje součet dojezdových vzdáleností všech vesnic pro vámi nalezené řešení. Druhý řádek obsahuje rostoucí posloupnost P kladných celých čísel. Tato čísla jsou čísla výjezdů, u kterých navrhujete zřídit poštovní úřad. Pokud existuje více optimálních řešení, váš program vypíše právě jedno z nich.

Příklad vstupu:

`POST.IN`

`10 5`

`1 2 3 6 7 9 11 22 44 50`

`POST.OUT`

`9`

`2 7 22 44 50`

Hodnocení. Pokud výstupní soubor nemá formát odpovídající zadání nebo pokud součet uvedený na prvním řádku neodpovídá řešení uvede-

ném na druhém řádku, váš program bude v příslušném testu hodnocen 0 body. V opačném případě se počet vámi získaných bodů spočte podle níže uvedené tabulky následujícím postupem: Označme S součet dojezdových vzdáleností všech vesnic pro vámi nalezené řešení a $S_{\min}$ tento součet pro optimální řešení; váš program bude hodnocen tolika body, kolik je uvedeno pro odpovídajíc poměr $q = S/S_{\min}$ v druhém řádku tabulky.

$q = S/S_{\min}$	Počet bodů
$q = 1,0$	10
$1,0 < q \leq 1,1$	5
$1,1 < q \leq 1,15$	4
$1,15 < q \leq 1,2$	3
$1,2 < q \leq 1,25$	2
$1,25 < q \leq 1,3$	1
$1,3 < q$	0

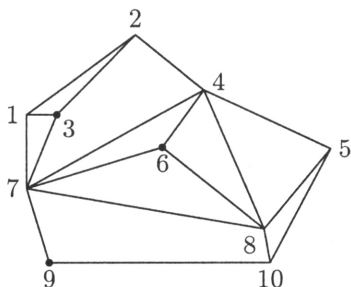
6. Velké zdi

V jednom nejmenovaném státě bylo zbudováno několik „Velkých zdí“. Každá z těchto zdí spojuje právě dvě města, žádné dvě zdi se nekříží a mezi každými dvěma městy vede nejvýše jedna zeď. Stát je takto rozdělen na několik oblastí takových, že k přesunu z jedné oblasti do druhé je třeba projít některým z měst nebo přelézt zeď.

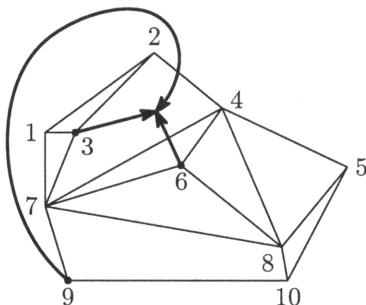
Demonstranti proti ekonomické globalizaci žijí v různých městech, v každém městě však žije nejvýše jeden z demonstrantů. Občas chtějí demonstranti uspořádat party nebo demonstraci. Za tímto účelem se potřebují všichni sejít v některé z oblastí ohraničených zdmi; z pochopitelných důvodů se však nemohou sejít v žádném z měst. Při své cestě na kolech na místo srazu nechtějí projet žádným městem, aby je nezadržela policie, a navíc chtějí přelézat co nejmenší počet zdí (s kolem to není zrovna nejsnazší).

Během své cesty na místo srazu musí každý z demonstrantů přelézt několik zdí (možná i žádnou). Demonstranti proto chtějí zvolit takovou oblast pro setkání, aby součet počtů přelezení zdí uskutečněných všemi demonstranty byl co nejmenší.

Města jsou označena celými čísly od 1 do N , kde N je počet všech měst. Očíslované uzly na obr. 74 představují města a úsečky spojující uzly odpovídají zdem. Předpokládejme, že ve státě žijí tři demonstranti a ti bydlí ve městech 3, 6 a 9. Nejvýhodnější oblast pro jejich setkání a příslušné trasy pro jednotlivé demonstranty jsou vyznačeny na obr. 75.



Obr. 74



Obr. 75

Demonstranti budou muset přelézt dvě zdi: demonstrant z města 9 musí přelézt zeď vedoucí mezi městy 2 a 4, demonstrant z města 6 musí přelézt zeď vedoucí mezi městy 4 a 7.

Vytvořte program, který obdrží na vstupu počet měst, popis oblastí ohraničených zdmi a seznam měst, kde bydlí demonstranti. Program poté určí optimální oblast pro setkání demonstrantů a spočítá minimální počet přelezení zdí uskutečněných dohromady všemi demonstranty při jejich cestě na místo srazu.

Vstup: Vstupní soubor se jmenuje `WALLS.IN`. Jeho první řádek obsahuje jedno celé kladné číslo M — počet oblastí ($2 \leq M \leq 200$). Druhý řádek obsahuje jedno celé kladné číslo N — počet měst ($3 \leq N \leq 250$). Třetí řádek obsahuje jedno celé kladné číslo L — počet demonstrantů ($1 \leq L \leq 30$, $L \leq N$). Čtvrtý řádek vstupního souboru obsahuje rostoucí posloupnost L různých celých kladných čísel — seznam čísel měst, v nichž žijí demonstranti.

Poté ve vstupním souboru následuje $2M$ řádků. Každá dvojice po sobě následujících řádků popisuje jednu z oblastí, tzn. první dva z těchto řádků popisují první oblast, následující dva řádky popisují druhou oblast, atd. První řádek v každé dvojici udává počet měst, I , která leží na hranici příslušné oblasti. Druhý řádek dvojice pak obsahuje posloupnost I celých kladných čísel, která jsou čísla měst ležících na hranici příslušné oblasti a která jsou zde uvedena seřazená po směru hodinových ručiček (začátek této posloupnosti může být zvolen libovolně). Poslední oblast uvedená v souboru je vnější oblast; v popisu této oblasti jsou města na její hranici uspořádána proti směru hodinových ručiček (to je jediná výjimka proti pravidlu o pořadí výpisu měst na hranici oblasti). Pořadí, v jakém jsou jednotlivé oblasti uvedeny ve vstupním souboru, určuje jejich očíslování,

tzn. oblast uvedená jako první má číslo 1, oblast uvedená jako druhá má číslo 2 atd. Povšimněte si, že ve vstupním souboru je uveden popis všech oblastí včetně „vnější“ oblasti.

Výstup: Výstupní soubor se jmenuje WALLS.OUT. První řádek obsahuje jediné celé číslo, které představuje minimální celkový počet přelezení zdí na cestě demonstrantů na místo srazu. Druhý řádek obsahuje jediné celé číslo, kterým je pořadové číslo oblasti vybrané k uskutečnění srazu demonstrantů. Pokud existuje více optimálních řešení, váš program vypíše právě jedno z nich.

Příklad vstupu: Následující vstupní a výstupní soubor odpovídají příkladu uvedenému v zadání.

WALLS.IN		WALLS.OUT
10	3	2
10	4 8 6	3
3	3	
3 6 9	6 8 7	
3	3	
1 2 3	4 5 8	
3	4	
1 3 7	7 8 10 9	
4	3	
2 4 7 3	5 10 8	
3	7	
4 6 7	7 9 10 5 4 2 1	

doc. RNDr. Leo Boček, CSc.,
RNDr. Karel Horák, CSc.,
doc. RNDr. Jaromír Šimša, CSc.,
RNDr. Jaroslav Švrček, CSc.,
doc. RNDr. Pavel Töpfer, CSc.

**ČTYŘICÁTÝ DEVÁTÝ ROČNÍK
MATEMATICKÉ OLYMPIÁDY
NA STŘEDNÍCH ŠKOLÁCH**

Vydala Jednota českých matematiků a fyziků,
Žitná 25, 117 10 Praha 1
jako svou publikaci č. 57-551-05
Sazbu programem $\text{\TeX}$ připravil Karel Horák
Vytisklo Polygrafické středisko
Univerzity Palackého v Olomouci
1. vydání
Praha 2005

ISBN 80-7015-984-7

**matematická
olympiáda**

MO
M

ISBN 80-7015-984-7

ISBN 80-7015-984-7



9 788070 159842