

45. ročník matematické olympiády na středních školách

8. mezinárodní olympiáda v informatice

In: Leo Boček (editor); Karel Horák (editor); Richard Kollár (editor); Václav Sedláček (editor); Jaromír Šimša (editor); Pavel Töpfer (editor): 45. ročník matematické olympiády na středních školách. Zpráva o řešení úloh ze soutěže konané ve školním roce 1995/1996. 37. mezinárodní matematická olympiáda. 8. mezinárodní olympiáda v informatice. (Czech). Praha: Jednota českých matematiků a fyziků, 1997. pp. 157–166.

Persistent URL: <http://dml.cz/dmlcz/405006>

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>

8. mezinárodní olympiáda v informatice

Na konci července se sjeli nejlepší mladí programátoři z celého světa do maďarského města Veszprém. Konal se tam již osmý ročník mezinárodní olympiády v informatice (IOI — International Olympiad in Informatics). Soutěže se zúčastnilo celkem 215 studentů středních škol z 56 zemí. Mezi nimi nechybělo ani české reprezentační družstvo. Naši zemi zastupovali nejúspěšnější řešitelé celostátního kola 45. ročníku matematické olympiády — kategorie P, a sice Daniel Král (gymnázium Zlín, Lesní čtvrť), Stanislav Mikeš (gymnázium České Budějovice, Jírovcova), Robert Špalek (gymnázium Brno, tř. kpt. Jaroše) a Tomáš Tichý (gymnázium Pardubice, Dašická ul.). Vedoucími české delegace byli doc. RNDr. *Václav Sedláček*, CSc., z Fakulty informatiky Masarykovy univerzity v Brně a RNDr. *Pavel Töpfer*, CSc., z Matematicko-fyzikální fakulty Univerzity Karlovy v Praze.



*International Olympiad
in Informatics*

National Committee,
John von Neumann Computer Society
H-1054 Budapest, Báthori u. 16.

Maďarští organizátoři zvolili pro uspořádání olympiády hezké a pohodlné město Veszprém ležící nedaleko Balatonu. Všichni účastníci byli ubytováni v internátech místních škol, vlastní soutěž probíhala v prostorách krásného moderního gymnázia. Stačilo jen rozmístit do tříd několik stovek nových osobních počítačů — některé přímo pro soutěž, jiné pro samostatnou práci soutěžících ve volném čase a další pro vedoucí všech delegací.

Tým odborných pracovníků z maďarských univerzit připravil velmi pěkné a zajímavé soutěžní úlohy. V každém ze dvou soutěžních dní řešili účastníci olympiády po třech příkladech. Na svou práci měli vždy pět hodin času. Požadovaným výsledkem každé úlohy byl odladěný program. Soutěžící nesměli používat žádné vlastní pomůcky ani písemné materiály, k dispozici měli pouze přidělené osobní počítače. Při své práci mohli používat programovací jazyky Pascal, C/C++ a Basic. Odevzdané programy byly plně automaticky testovány pomocí předem připravených tajných sad testovacích dat. Při testování se sledovala nejen správnost výsledků,

ale také doba výpočtu. Za správný a dostatečně rychlý výpočet programu s každými testovacími vstupními daty získal řešitel několikabodové ocenění. Každý den bylo možné dosáhnout maximálně sta bodů. To se však v prvním soutěžním dnu nepodařilo nikomu, zatímco ve druhém dnu soutěže bylo „stobodových řešitelů“ více.

Účastníci s nejvyššími bodovými zisky obdrželi na závěr soutěže medaile a hodnotné věcné ceny. Krásné medaile vyrobila zvláště pro tuto olympiádu světoznámá porcelánka Herend ležící nedaleko Veszprému. Celkem bylo uděleno 20 zlatých, 36 stříbrných a 52 bronzových medailí. Čeští reprezentanti zakončili své vystoupení na olympiádě tradičně výborně, všichni čtyři se umístili na medailových místech. Vynikajícím úspěchem je pak absolutní vítězství *Daniela Krála* v celé soutěži. Zlatou medaili a 1. místo získal za plných 196 bodů z 200 možných. Ostatní tři soutěžící z České republiky obdrželi bronzové medaile s následujícím umístěním: *Tomáš Tichý* 71.–72. místo (133 bodů), *Stanislav Mikeš* 84.–89. místo (116 bodů) a *Robert Špalek* 95.–98. místo (111 bodů).

1. Čína	736	11. Thajsko	558
2. Rusko	709	12. Česká republika	556
3. Slovensko	684	13. Bělorusko	535
4. Polsko	682	14. Chorvatsko	531
5. Rumunsko	652	15. Vietnam	515
6. Tchajwan	647	16. Bulharsko	507
7. Litva	611	17.–18. Estonsko	497
8. Írán	607	Maďarsko	497
9. Jugoslávie	575	19. Německo	469
10. Korea	565	20. Turecko	458

Texty soutěžních úloh

1. Hra (30 bodů)

Dva hráči hrají následující hru. Hrací plán je tvořen řadou kladných celých čísel. Hráči se pravidelně střídají na tahu. Hráč, který je na tahu, si vezme číslo buď z levého, nebo z pravého konce řady. Toto číslo je odstraněno z hracího plánu. Hra končí ve chvíli, když jsou odebrána z hracího plánu všechna čísla. První hráč vyhrává, pokud součet jím odebraných čísel je alespoň roven součtu čísel odebraných druhým hráčem. Druhý hráč hraje nejlepším možným způsobem. První hráč začíná hru. Pokud

hrací plán obsahuje na začátku hry sudý počet čísel, potom existuje vyhrávající strategie pro prvního hráče.

Napište program, který implementuje vyhrávající strategii prvního hráče. Odpovědi druhého hráče jsou poskytovány daným počítačovým programem. Oba hráči spolu komunikují pomocí tří procedur z modulu Play, který je vám k dispozici. Jsou to procedury `StartGame`, `MyMove` a `YourMove`. První hráč začíná hru voláním procedury bez parametrů `StartGame`. Pokud první hráč svým tahem odebírá číslo z levého konce dané řady čísel, provede proceduru `MyMove('L')`. Podobně provedení procedury `MyMove('R')` oznamuje druhému hráči, že si první hráč vybral číslo z pravého konce řady. Druhý hráč, tj. počítačový program, provádí svůj tah okamžitě. První hráč se dozví jeho tah zavoláním procedury `YourMove(C)`, kde C je znaková proměnná (v jazyce C/C++ volání procedury zapíšete ve tvaru `YourMove(&C)`). Hodnotou C je 'L' nebo 'R' podle toho, zda druhý hráč odebral číslo z levého nebo pravého konce řady.

Vstupní data. První řádek vstupního souboru `INPUT.TXT` obsahuje počáteční velikost N hracího plánu (počet čísel v řadě). Číslo N je sudé a $2 \leq N \leq 100$. Zbývajících N řádků vstupního souboru popisuje hrací plán zleva doprava. Každý z těchto N řádků obsahuje jedno číslo zadané řady. Každé číslo je rovno nejvýše 200.

Výstupní data. Po skončení hry zapíše váš program výsledek do výstupního souboru `OUTPUT.TXT`. Soubor obsahuje jediný řádek se dvěma celými čísly. Prvním číslem je součet čísel odebraných prvním hráčem, druhé číslo je součtem čísel odebraných druhým hráčem. Váš program musí hru skutečně hrát a výsledek musí odpovídat průběhu hry.

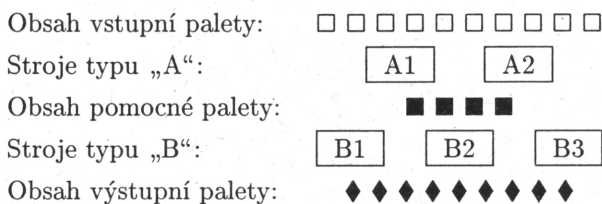
Příklad vstupu a výstupu. Následující obrázek ukazuje vstupní soubor obsahující počáteční hrací plán a možný výstupní soubor.

INPUT . TXT
6
4
7
2
9
5
2

OUTPUT . TXT
15 14

2. Opracovávání výrobků (30 bodů)

V továrně pracuje výrobní linka. S každým výrobkem je nutno provést dvě operace: nejprve operaci „A“, potom operaci „B“. Pro každou z těchto operací je k dispozici několik strojů. Obrázek 40 znázorňuje uspořádání výrobní linky, která pracuje následujícím způsobem. Stroj provádějící operaci „A“ vezme výrobek ze vstupní palety, provede s ním operaci „A“ a umístí výrobek do pomocné palety. Stroj typu „B“ bere výrobky z pomocné palety, provádí s nimi operaci „B“ a pak je ukládá do výstupní palety. Všechny stroje mohou pracovat zároveň nezávisle jeden na druhém, velikost palet není omezena. Stroje mají různé parametry, pro každý stroj známe jeho dobu opracovávání výrobku.



Obr. 40

Určete nejkratší čas, ve kterém mohou být dokončeny operace „A“ pro všech N výrobků, je-li opracovávání výrobků zahájeno v čase 0 (podúloha A). Dále vypočtete minimální čas, který je potřebný pro dokončení obou operací se všemi N výrobky (podúloha B).

Vstupní data. Soubor INPUT.TXT je tvořen pěti řádky s kladnými celými čísly. První řádek obsahuje číslo N , které představuje celkový počet výrobků ($1 \leq N \leq 1000$). Na druhém řádku je uveden počet M_1 strojů typu „A“ ($1 \leq M_1 \leq 30$). Na třetím řádku je M_1 kladných celých čísel, která udávají dobu opracovávání pro jednotlivé stroje typu „A“. Čtvrtý a pátý řádek obsahují podobně počet M_2 strojů typu „B“ ($1 \leq M_2 \leq 30$) a dobu opracovávání pro jednotlivé stroje typu „B“. Dobu opracovávání výrobku měříme v jednotkách času. Tato doba zahrnuje i čas nezbytný pro přenos výrobku z palety před opracováním a čas na uložení opracovaného výrobku na paletu. Každý z časových údajů je nejméně 1 a nejvýše 20.

Výstupní data. Výstupní soubor OUTPUT.TXT je tvořen dvěma řádky. První řádek obsahuje jedno kladné celé číslo představující řešení podúlohy A. Druhý řádek obsahuje jedno kladné celé číslo — řešení podúlohy B.

Příklad vstupu a výstupu. Následující obrázek ukazuje možný vstupní soubor a jemu odpovídající výstupní soubor.

INPUT . TXT
5
2
1 1
3
3 1 4

OUTPUT . TXT
3
5

3. Síť škol (40 bodů)

Určitý počet škol je zapojen do počítačové sítě. Školy mají mezi sebou uzavřeny dohody, podle kterých si zasílají software. Každá škola má seznam škol, kterým zasílá software („zásobované školy“). Pokud je škola B na seznamu škol zásobovaných ze školy A, pak škola A nemusí být nutně na seznamu škol zásobovaných školou B.

Napište program, který spočítá minimální počet škol, kterým musí být zaslána kopie nového softwaru, aby byl tento software postupným předáváním mezi školami na základě uzavřených dohod nakonec rozeslán na všechny školy v síti (podúloha A).

Dalším úkolem je navrhnout modifikaci dosavadních smluv mezi školami tak, aby stačilo zaslat nový software na jednu libovolnou školu a přitom bylo jisté, že se v síti rozšíří na všechny školy. Vypočtete minimální počet nových smluv, které je třeba uzavřít, aby po zaslání nového softwaru na jednu libovolnou školu obdržely tento software nakonec postupným šířením v síti všechny školy (podúloha B). Jedna nová smlouva znamená přidání jedné školy do seznamu zásobovaných škol jedné jiné školy.

Vstupní data. První řádek vstupního souboru INPUT.TXT obsahuje jedno celé číslo N , kterým je celkový počet škol v síti ($2 \leq N \leq 100$). Tyto školy jsou označeny celými kladnými čísly od 1 do N . Každý z následujících N řádků obsahuje seznam zásobovaných škol. Řádek číslo $i + 1$ obsahuje čísla škol zásobovaných školou číslo i . Každý seznam je ukončen nulou. Pokud škola nezasobuje žádné jiné školy (její seznam zásobovaných škol je prázdný), pak na příslušném řádku vstupního souboru je pouze nula.

Výstupní data. Výstupní soubor OUTPUT.TXT je tvořen dvěma řádky. První řádek obsahuje jedno celé kladné číslo představující řešení podúlohy A. Druhý řádek obsahuje řešení podúlohy B.

Příklad vstupu a výstupu. Obrázek ukazuje možný vstupní soubor a jemu odpovídající výstupní soubor.

INPUT . TXT
5
2 4 3 0
4 5 0
0
0
1 0

OUTPUT . TXT
1
2

4. Magické čtverce (40 bodů)

Poté, co pan Rubik vynalezl úspěšnou magickou kostku, navrhl také její rovinnou verzi a nazval ji magické čtverce. Je to obdélník složený z osmi čtverců stejné velikosti (obr. 41).

1	2	3	4
8	7	6	5

Obr. 41. — Počáteční konfigurace.

V této úloze budeme uvažovat verzi, v níž jsou čtverce obarveny různými barvami. Barvy jsou označeny celými čísly od 1 do 8 (obr. 41). Konfigurace obdélníku je popsána posloupností osmi čísel. Jsou to čísla barev v tom pořadí, jaké získáme při procházení jednotlivých čtverců ve směru pohybu hodinových ručiček počínaje v levém horním rohu obdélníku. Například konfigurace na obr. 41 je popsána posloupností čísel (1, 2, 3, 4, 5, 6, 7, 8). Tato konfigurace je zároveň počáteční konfigurací.

Mezi konfiguracemi lze přecházet pomocí tří základních operací označených písmeny 'A', 'B' a 'C':

'A': výměna horního a dolního řádku,

'B': jednoduchý cyklický posuv celého obdélníku směrem vpravo,

'C': jednoduchá rotace prostředních čtyř čtverců ve směru pohybu hodinových ručiček.

Všechny konfigurace jsou dosažitelné z počáteční konfigurace prováděním těchto tří základních operací. Význam základních operací ukazuje obr. 42. Čísla vně obdélníku označují pozice jednotlivých čtverců. Jestliže

čtverec na pozici p obsahuje číslo i , znamená to, že provedením příslušné operace se čtverec z pozice i přesunul na pozici p .

	1	2	3	4
A	8	7	6	5
	1	2	3	4
	8	7	6	5

	1	2	3	4
B	4	1	2	3
	5	8	7	6
	8	7	6	5

	1	2	3	4
C	1	7	2	4
	8	6	3	5
	8	7	6	5

Obr. 42. — Základní operace.

Napište program určující posloupnost základních operací, pomocí níž je počáteční konfigurace z obr. 41 převedena na určenou cílovou konfiguraci (podúloha A). Pokud délka nalezené posloupnosti nepřesáhne 300, získáte další dva body (podúloha B).

Vstupní data. Vstupní soubor INPUT.TXT obsahuje jediný řádek. Na něm je zapsáno 8 celých kladných čísel popisujících cílovou konfiguraci.

Výstupní data. Na prvním řádku výstupního souboru OUTPUT.TXT je zapsána délka L nalezené posloupnosti operací. Dalších L řádků popisuje nalezenou posloupnost operací. Každý z nich obsahuje na první pozici vždy jedno z písmen 'A', 'B', 'C' — označení příslušné základní operace.

Pomůcka. V pracovním adresáři úlohy máte k dispozici program MTOOL.EXE, který vám umožní hrát si s magickými čtverci. Zavoláním programu „mtool input.txt output.txt“ můžete experimentovat s cílovou konfigurací a s posloupností operací.

Příklad vstupu a výstupu. Následující obrázek ukazuje příklad vstupního a jemu odpovídajícího výstupního souboru.

INPUT.TXT
2 6 8 4 5 7 3 1

OUTPUT.TXT
7
B
C
A
B
C
C
B

5. Nejdelsí prefix (40 bodů)

Strukturu některých biologických objektů lze reprezentovat posloupností jejich základních prvků. Tyto prvky budeme označovat velkými písmeny. Biology zajímá možnost rozkládat dlouhé posloupnosti prvků na menší. Takové krátké úseky budeme nazývat komponenty. Řekneme, že posloupnost S může být vytvořena z dané množiny komponent P , jestliže v množině P existuje takových n komponent p_1, \dots, p_n , jejichž zřetězením získáme posloupnost S . Zřetěžením komponent p_1, \dots, p_n rozumíme jejich spojení v daném pořadí, a to bezprostředně za sebou bez mezer. Některé komponenty se v tomto zřetězení mohou vyskytnout vícekrát. Všechny komponenty obsažené v uvažované množině P nemusí být ve zřetězení použity. Například posloupnost $ABABACABAAB$ může být vytvořena z množiny komponent $\{A, AB, BA, CA, BBC\}$.

Prvních K znaků posloupnosti S tvoří prefix posloupnosti S délky K . Napište program, který přečte ze vstupu množinu komponent P a posloupnost prvků T . Program spočítá délku nejdelšího prefixu posloupnosti T , který může být vytvořen z komponent obsažených v P .

Vstupní data. Vstupní data jsou uložena ve dvou souborech. Soubor `INPUT.TXT` popisuje množinu komponent P , soubor `DATA.TXT` obsahuje zkoumanou posloupnost T . První řádek souboru `INPUT.TXT` obsahuje číslo N , které představuje počet komponent v množině P ($1 \leq N \leq 100$). Každá komponenta je pak popsána na dvou po sobě jdoucích řádcích. První z nich udává délku L komponenty ($1 \leq L \leq 20$), druhý obsahuje řetězec velkých písmen (z rozmezí od 'A' do 'Z') délky L . Všechny zadané komponenty jsou navzájem různé.

První pozice každého řádku v souboru `DATA.TXT` obsahuje jedno velké písmeno. Na posledním řádku tohoto souboru je na první pozici znak tečka ('.'). Délka posloupnosti T je alespoň 1 a nejvýše 500 000.

Výstupní data. Výstupní soubor `OUTPUT.TXT` obsahuje jediné číslo — délku nejdelšího prefixu posloupnosti T , kterou lze vytvořit z množiny komponent P .

Příklad vstupu a výstupu. Následující obrázek ukazuje dva vstupní soubory a jim odpovídající výstupní soubor.

6. Třídění posloupnosti tři hodnot (20 bodů)

Třídění je jedním z nejčastějších úkolů prováděných při výpočtech. Uvažujme zvláštní problém třídění, kdy klíčové položky ve tříděných záznamech mohou nabývat nejvýše tří různých hodnot. Tato situace nastane například při třídění držitelů medailí podle druhu získané medaile

INPUT.TXT
5
1
A
2
A B
3
B B C
2
C A
2
B A

DATA.TXT
A
B
A
B
A
C
A
B
A
A
B
C
B

OUTPUT.TXT
11

tak, aby držitelé zlatých medailí byli první, po nich následovali držitelé stříbrných medailí a nakonec držitelé bronzových medailí.

V této úloze budou přípustnými hodnotami klíčů celá čísla 1, 2 a 3. Požaduje se setřídít záznamy podle hodnot klíčů v neklesajícím pořadí. Třídění musí být provedeno posloupností operací výměn. Pro každou operaci výměny jsou dána dvě čísla p, q určující pozici ve tříděné posloupnosti. Operace výměny spočívá ve vzájemné záměně prvků uložených v posloupnosti na pozicích p, q .

Je dána posloupnost hodnot klíčů. Napište program, který určuje minimální počet operací výměny, pomocí nichž lze zadanou posloupnost setřídít (podúloha A). Dále program nalezne odpovídající posloupnost operací výměny pro toto třídění (podúloha B).

Vstupní data. První řádek vstupního souboru INPUT.TXT obsahuje počet záznamů N ($1 \leq N \leq 1000$). Každý z následujících N řádků obsahuje jednu hodnotu klíče.

Výstupní data. První řádek výstupního souboru OUTPUT.TXT obsahuje minimální počet operací výměny L , které zajistí setřídění zadané posloupnosti (výsledek podúlohy A). Následujících L řádků výstupního souboru popisuje posloupnost výměn v tom pořadí, v jakém budou prováděny. Na každém z těchto řádků jsou zapsána dvě čísla p, q udávající pozice vyměňovaných prvků (výsledek podúlohy B). Pozice v posloupnosti jsou očíslovány celými čísly od 1 do N .

Příklad vstupu a výstupu. Následující obrázek ukazuje příklad vstup-

ního a jemu odpovídajícího výstupního souboru.

INPUT.TXT
9
2
2
1
3
3
3
2
3
1

OUTPUT.TXT
4
1 3
4 7
9 2
5 9

