

41. ročník matematické olympiády na středních školách

4. Mezinárodní olympiáda v informatice

In: Andrej Blaho (editor); Leo Boček (editor); Karel Horák (editor); Jozef Moravčík (editor); Václav Sedláček (editor); Jaromír Šimša (editor); Pavel Töpfer (editor): 41. ročník matematické olympiády na středních školách. Zpráva o řešení úloh ze soutěže konané ve školním roce 1991/1992. 33. mezinárodní matematická olympiáda. 4. mezinárodní olympiáda v informatice. (Czech). Praha: Jednota českých matematiků a fyziků, 1997. pp. 134–141.

Persistent URL: <http://dml.cz/dmlcz/404964>

Terms of use:

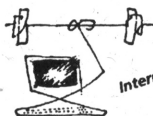
Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



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>

4. mezinárodní olympiáda v informatice

Čtvrtá mezinárodní olympiáda v informatice (IOI'92) pro středoškoly se konala ve dnech 12.–21. července 1992 v Bonnu pod záštitou UNESCO s přispěním několika desítek sponzorů, z nichž k nejvýznamnějším patřili IBM, APPLE, Springer Verlag, SIEMENS, Borland atd. Olympiády se zúčastnilo 166 soutěžících (z toho 5 dívek) ze 45 zemí ze všech kontinentů a navíc 5 dalších zemí vyslalo své pozorovatele.



4.
Internationale Olympiade
Bonn, Deutschland
in Informatik
IOI'92

Československý tým byl složen na základě výsledků studentů v celostátním kole matematické olympiády v kategorii P a podle výsledků testů, konaných v průběhu týdenního soustředění 11 vítězů v této kategorii, které se konalo na KU v Bratislavě. Složení našeho družstva bylo následující: *Matej Ondrušek*, student 3. ročníku gymnázia J. Hronca v Bratislavě, *Jan Kotas*, absolvent gymnázia na Mikulášském náměstí v Plzni, *Jan Kybic*, absolvent gymnázia v Korunní ulici v Praze a *Tomáš Vinař*, student 2. ročníku gymnázia na Šrobárově ulici v Košicích. Vedoucím delegace byl doc. *Václav Sedláček* z Masarykovy university v Brně a jeho zástupcem Mgr. *Richard Nemec* z Univerzity Komenského v Bratislavě.

V rámci vlastní soutěže byly řešeny 2 soutěžní úlohy, na které měli soutěžící vyhrazeno vždy 5 hodin čistého času. Úlohy byly vybrány v den jejich řešení mezinárodní porotou, složenou z vedoucích delegací všech zúčastněných států. Pro výběr byly vědeckou komisí soutěže předloženy vždy 3 úlohy. Každý student pak řešil úlohu na přiděleném osobním počítači. Výsledné produkty byly za přítomnosti studenta a vedoucího týmu testovány koordinátory, kteří je podrobili předepsaným testům a na jejich základě bodově ohodnotili tak, že maximální možný bodový zisk byl 100 bodů. Případné námítky ze strany delegací řešilo shromáždění koordinátorů, které rovněž stanovilo definitivní bodové hodnocení soutěžících. Ocenění zlatými, stříbrnými a bronzovými medailemi pak bylo dáno bodovým ziskem za obě úlohy.

Celkem bylo uděleno 13 zlatých (pro soutěžící se ziskem 198–200 bodů), 31 stříbrných (175–195 bodů) a 41 bronzových medailí (123–172 bodů). Českoslovenští soutěžící navázali na vynikající výsledky z předchozí olympiády a ve dvojnásobné konkurenci ve srovnání s předchozí olympiádou získali M. Ondrušek zlatou medaili (200 bodů), Jan Kotas (185 bodů) a Jan Kybic (190 bodů) medaile stříbrné. V neoficiální soutěži družstev, do níž byly sečteny body všech členů družstva, byl československý tým uveden mezi 12 nejúspěšnějšími družstvy se ziskem 645 bodů.

Mezinárodní olympiáda v programování se svým charakterem liší od naší kategorie P matematické olympiády především v tom, že účastníci řeší zadané úlohy na osobním počítači, který má každý soutěžící k dispozici. Zadání úlohy je mu k dispozici v angličtině a v rodném jazyku. Případné dotazy je možno klást pouze písemně přes mezinárodní jury, která dotaz předá vedoucímu delegace. Možné odpovědi na dotazy jsou jenom „ano“, „ne“, „bez komentáře“, a to pouze prvních 30 minut soutěže. Po skončení soutěže jsou úlohy soutěžících podrobeny testům. Na základě výsledků testů jsou úlohy obodovány. V žádném případě se neprovádí analýza zdrojového textu s ohledem na složitost navrhovaných algoritmů. Ostatně jak uvidíte z textu zadání úloh, byly v letošním roce vybrány mezinárodní jury obě úlohy, které bylo myslitelné řešit technikou backtracking. Trend v návrzích úloh je v současné době takový, že se uplatní úlohy typu spíše vytváření prostředků (tools) pro sérii úloh. Přestože naši soutěžící byli v rámci matematické olympiády orientováni poněkud jiným směrem, ukazuje se na základě jejich výsledků, že jejich průprava v psaní efektivních programů z hlediska paměťové či časové složitosti jim umožňuje úspěšné výsledky i v mezinárodních soutěžích s přímým použitím počítačů. Snad jenom větší pečlivost při návrhu testů omezujících podmínek pro vstupní data byla příčinou oněch nepatrných bodových ztrát, které nás dělily od dalších zlatých medailí. Přitom efektivita resp. osekávání možností pro backtracking nebyly vůbec zohledňovány ve výsledném bodovém ohodnocení, takže úloha trvající na počítači 1 sekundu byla ohodnocena stejně jako úloha trvající 8 hodin.

Mimo vlastní soutěž připravili organizátoři z Institutu pro matematiku a zpracování dat a z Institutu Gustava Stresemanna pod vedením dr. Petra Heyderhoffa bohatý kulturní, společenský a doprovodný program. Všichni účastníci soutěže byli přijati před zahájením soutěže starostou Bonnu, v průběhu zájezdu do Düsseldorfu byli přijati zemským ministrem školství, navštívili Kolín nad Rýnem, Heidelberg, poznali Bonn, jeho okolí i část Porýní. Sponzoři soutěže věnovali mimo jiné dr-

žitelům zlatých medailí hodnotné ceny a všem účastníkům věcné dárky. Při slavnostním vyhlášení výsledků byly tyto ceny předány řadou oficiálních osobností v čele s federálním ministrem školství SRN. Účastníkům olympiády poslal pozdravný telegram prof. Konrad Zuse.

Závěrem uvádíme doslovné znění úloh, které byly řešeny účastníky olympiády, abyste si na nich mohli prověřit svoje programátorské zkušenosti. Nezapomeňte dodržet všechna omezení a pokyny, jejichž nedodržení v soutěži by znamenalo bodovou ztrátu.

Texty soutěžních úloh

7. Ostrovy v moři

MORE je reprezentováno sítí velikosti N krát N . Každý OSTROV je „*“ v této síti. Vaší úlohou je zrekonstruovat MAPU ostrovů pouze z KÓDOVANÉ INFORMACE o horizontálním a vertikálním rozložení těchto ostrovů. Pro ilustraci tohoto kódování sledujte následující mapu:

*		*	*			1	2
		*	*	*		3	1
*		*		*		1	1 1
		*	*	*	*	5	
*	*		*		*	2	1 1
			*			1	
1	1	4	2	2	1		
1	2		3		2		
1							

Čísla napravo v každém řádku reprezentují pořadí a velikost každé skupiny ostrovů v tomto řádku. Například „1 2“ v prvním řádku znamená, že jeden ostrov je následován skupinou dvou ostrovů, které jsou obklopeny mořem nějaké délky nalevo i napravo od každé skupiny ostrovů. Podobně posloupnost „1 1 1“ pod prvním sloupcem znamená, že tento sloupec obsahuje tři skupiny, každá po jednom ostrovu, atd.

Zadání úlohy: Implementujte program, který opakuje následující kroky, dokud není úplně přečten vstupní soubor, který obsahuje několik takových bloků informací:

1. Přečti následující blok informací ze vstupního ASCII souboru (pro strukturu dat v tomto souboru viz příklad dále) a zobraz jej na obrazovce.

Každý blok informací je složen z velikosti čtvercové sítě, následované informacemi o řádcích a sloupcích. Popis každého řádku i sloupce se objevuje v jednom řádku souboru jako posloupnost čísel oddělených mezerami a ukončená 0.

2. Zrekonstruuje mapu (nebo všechny možné mapy, pokud je možné více než jedno řešení, viz Příklad-4 dále) a zobrazte je na obrazovce.

3. Zapište mapu (mapy) na konec výstupního ASCII souboru. Každé prázdné místo musí být reprezentováno dvojicí mezer. Každý ostrov nechť je reprezentován „*“ následovanou jednou mezerou. Různé mapy vyhovující stejným omezením nechť jsou odděleny prázdným řádkem. Pokud neexistuje řešení vyhovující popisům, indikujte to řádkem s textem „no map“. Řešení odpovídající různým blokům informací musí být oddělena řádkem s textem „next problem“.

Technická omezení:

Omezení-1: N nesmí být menší než 1 a větší než 8

Omezení-2: Uložte váš výsledný program do textového ASCII souboru C:\IOI\DAY-1\413-PROG.xxx.

Přípona .xxx je:

.BAS pro programy v BASICu, .C pro programy v jazyku C, .LCN pro LOGO programy, .PAS pro programy v PASCALu.

Omezení-3: Jméno vstupního ASCII souboru pro čtení kódované informace musí být C:\IOI\DAY-1\413-SEAS.IN.

Omezení-4: Jméno výstupního souboru pro zápis mapy (map) musí být C:\IOI\DAY-1\413-SEAS.OU.

Příklady:

6 Příklad-1. (viz problém v zadání): 6 je velikost
1 2 0 <-- začátek popisu řádků síť
3 1 0
1 1 1 0
5 0
2 1 1 0
1 0
1 1 1 0 <-- začátek popisu sloupců
1 2 0
4 0
2 3 0
2 0
1 2 0

4 Příklad-2. Řešení:sloupce: 1 2 3 4
0 řádek 1:
1 0 řádek 2: *
2 0 řádek 3: * *
0 řádek 4:
0
1 0
2 0
0

2 Příklad-3. Všimněte si, že neexistuje řešení
0 vyhovující těmto popisům
0
2 0
2 0

2 Příklad-4. Všimněte si, že existují dvě různé
1 0 mapy, vyhovující těmto popisům
1 0
1 0
1 0

Vzorové soubory:

Tyto správné příklady souborů jsou vám k dispozici v souborech:
C:\IOI\DAY-1\413-SEAS.IN a C:\IOI\DAY-1\413-SEAS.OU.

Upozornění: Úspěšný běh vašeho programu s těmito příklady nezaručuje nutně, že váš program je správný!!!

Bodové hodnocení:

Přečti blok informací ze vstupního souboru a zobraz jej ...	5 bodů
Zpracovávej všechny bloky informací postupně po jednom, dokud není úplně přečten celý vstupní soubor	10 bodů
Zrekonstruu jednu mapu pro každý blok informací (pokud existuje řešení) a zobraz ji.....	35 bodů
Zapiš výslednou mapu do výstupního souboru.....	5 bodů
Zrekonstruu všechny možné mapy (pokud existuje více ře- šení) a zobraz je	20 bodů
Zapiš všechny výsledné mapy do výstupního souboru správ- ně oddělené.....	10 bodů
Identifikuj bloky informací, které nemají řešení.....	5 bodů
Dodržení technických omezení	10 bodů

maximálně 100 bodů

8. Slézání hory

Klub horolezců má P členů očíslovaných 1 až P . Každý člen leze stejnou rychlostí a není rozdílu mezi rychlostí lezení nahoru a dolů. Horolezec číslo i spotřebuje denně $C(i)$ jednotek ZÁSOb a je schopen nést nejvýše $S(i)$ takových jednotek. Všechna $C(i)$ a $S(i)$ jsou celá čísla.

Předpokládejte, že horolezec s dostatečným množstvím zásob by potřeboval N dnů k dosažení vrcholu. Hora ale může být příliš vysoká, takže samotný horolezec není schopen nést všechny potřebné zásoby. Tudíž jistá SKUPINA horolezců začne na stejném místě a ve stejném čase. Horolezec, který se začne předčasně vracet ještě před dosažením vrcholu, odevzdá své nepotřebné zásoby ostatním horolezcům. Horolezci během expedice neodpočívají.

Vaší ÚLOHOU je najít rozvrh pro klub horolezců. Aspoň jeden horolezec musí dosáhnout vrcholu hory a všichni horolezci z vybrané skupiny se musí vrátit do startovního místa.

Zadání úlohy: Implementujte program, který dělá následující:

1. Přečte z klávesnice celé číslo N dnů potřebných k dosažení vrcholu, počet P horolezců v klubu a (pro všechna $1 \leq i \leq P$) hodnoty $S(i)$ a $C(i)$. Můžete předpokládat, že vstupní hodnoty jsou celá čísla. Zamítněte vstupy, které jsou nesmyslné.

2. Pokuste se najít plán pro slezání hory. Určete možnou skupinu $a(1), \dots, a(k)$ horolezců, kteří by měli být vybráni do skupiny a (pro všechna $1 \leq j \leq k$) množství zásob $M(j)$, které horolezec $a(j)$ ponese na startu.

Poznámka: Nemusí existovat plán pro všechny možné kombinace N , $S(i)$ a $C(i)$.

3. Vypište následující informace na obrazovce:

- Počet horolezců, kteří se skutečně zúčastní výpravy,
- celkové množství potřebných zásob,
- čísla horolezců $a(1), \dots, a(k)$,
- pro všechna $a(j)$ ($1 \leq j \leq k$) počáteční množství zásob $M(j)$, které ponese horolezec $a(j)$,
- den $D(j)$, kdy horolezec $a(j)$ začne sestupovat.

4. Rozvrh je OPTIMÁLNÍ, jestliže:

- počet vybraných horolezců je minimální a
- mezi všemi skupinami splňujícími podmínku a) jsou celkové spotřebované zásoby minimální.

Pokuste se najít řešení pokud možno nejbližší k optimálnímu.

Technická omezení:

Omezení-1: Uložte váš výsledný program do textového ASCII souboru C:\IOI\DAY-2\422-PROG.xxx. Přípona .xxx je: .BAS pro programy v BASICu, .C pro programy v jazyku C, .LCN pro LOGO programy, .PAS pro programy v PASCALu.

Omezení-2: Program musí zamítnout vstupy, kdy N je menší než 1 nebo větší než 100. P nesmí být menší než 1 ani větší než 20.

Vzorové soubory: Pro vaše potřeby je připraveno několik souborů obsahujících testovací data a vzory správných výstupů; podívejte se do adresáře C:\IOI\DAY-2.

Upozornění: Úspěšný běh vašeho programu s těmito příklady nezaručuje nutně, že váš program je správný!!!

Příklady: Následující text by mohl být dialogem s vaším programem, který může být i v rodném jazyce:

Dní na dosažení vrcholu:	4
Počet členů klubu:	5
Maximální zásoby horolezce	1:7
Denní spotřeba horolezce	1:1
Maximální zásoby horolezce	2:8
Denní spotřeba horolezce	2:2
Maximální zásoby horolezce	3:12
Denní spotřeba horolezce	3:2
Maximální zásoby horolezce	4:15
Denní spotřeba horolezce	4:3
Maximální zásoby horolezce	5:7
Denní spotřeba horolezce	5:1

2 potřební horolezci, potřebné množství zásob je 10.

Horolezci 1, 5 půjdou.

Horolezec 1 nese 7 a sestupuje po 4 dnu.

Horolezec 5 nese 3 a sestupuje po 1 dnu.

Naplánovat další výpravu? (Y/N) Y

Dní na dosažení vrcholu	:2
Počet členů klubu	:1
Maximální zásoby horolezce	1:3
Denní spotřeba horolezce	1:1

Uskutečnění výpravy není možné.

Naplánovat další výpravu? (Y/N) N

Bodové hodnocení:

Uživatelský dialog podle ilustračního příkladu	10 bodů
Nalezení řešení pro speciální případ, kdy všechna $C(i) = 1$ a všechna $S(i)$ jsou stejná	20 bodů
Nalezení řešení pro obecný případ	20 bodů
Nalezení řešení co nejblíže optimálnímu pro obecný případ	30 bodů
Detekce neřešitelných situací	10 bodů
Dodržení technických omezení	10 bodů

maximálně 100 bodů

