

MATEMATICKÁ OLYMPIÁDA NA STREDNÝCH ŠKOLÁCH

51. ROČNÍK, 2001/2002

Kategória P

Matematická olympiáda je súťaž pre študentov stredných škôl našej republiky. **Kategória P** je zameraná na programovanie a je určená študentom všetkých ročníkov.

Organizácia súťaže v kategórii P

Kategória P matematickej olympiády má tri postupové kolá — domáce, krajské a celoštátne.

V **I. kole** účastníci riešia štyri úlohy uvedené v tomto letáku. Riešenia odovzdajú svojmu učiteľovi informatiky do **9. novembra 2001**. Učitelia informatiky odošlú riešenia v tomto termíne zo školy na príslušnú adresu podľa krajov takto:

Košický, Prešovský:

RNDr. G. Andrejková, CSc., KMI PF UPJŠ, Jesenná 5, 041 54 Košice

Žilinský:

RNDr. P. Varša, KI FRI Žilinská univerzita, Moyzesova 20, 010 26 Žilina

Banskobystrický:

Mgr. L. Huraj, KI FPV UMB, Tajovského 40, 974 01 Banská Bystrica

Trnavský, Trenčiansky, Nitriansky:

Doc. Ing. V. Štoffová, CSc., KI FPV UKF, tr. A. Hlinku 1, 949 74 Nitra

Bratislavský:

RNDr. D. Pardubská, CSc., KVI FMFI UK, Mlynská Dolina, 842 48 Bratislava

Najúspešnejší riešitelia domáceho kola sú pozvaní do **II. kola** (krajského), kde riešia štyri teoretické úlohy. Do **III. kola** (celoštátneho) sú pozývaní najúspešnejší riešitelia všetkých krajských kôl, pričom riešia tri teoretické úlohy a dve praktické úlohy pri počítači. Z najlepších riešiteľov tohto kola SK MO vyberie družstvá pre Medzinárodnú informatickú olympiádu a Stredoeurópsku informatickú olympiádu.

Predbežné termíny 51. ročníka MO kat. P

I.	Domáce kolo	9. novembra 2001
II.	Krajské kolo	8. januára 2002
III.	Celoštátne kolo	10.-13. apríla 2002

Usporiadateľ súťaže

Matematickú olympiádu vyhlasuje *Ministerstvo školstva SR* v spolupráci s *Jednotou slovenských matematikov a fyzikov* a *Slovenskou komisiou Matematickej olympiády*. Súťaž organizuje *Slovenská komisia MO* a v jednotlivých krajoch ju riadia *krajské komisie MO* pri pobočkách JSMF. Na jednotlivých školách ju zaisťujú učitelia matematiky a informatiky. Celostátne kolo MO, tlač materiálov MO a ich distribúciu po organizačnej stránke zabezpečuje IUVENTA – zariadenie pre voľný čas detí, mládeže i dospelých MŠ SR v tesnej súčinnosti so slovenskou komisiou MO.

Formálna úprava riešenia

Riešenia súťažných úloh domáceho kola kategórie P pozostávajú z dvoch častí:

Popis riešenia. Riešenia musia obsahovať podrobný popis použitého algoritmu, zdôvodnenie jeho správnosti a diskusiu o efektivite zvoleného riešenia (t.j. posúdenie časových a pamäťových nárokov programu). Algoritmus by mal byť jasný už z popisu riešenia bez toho, aby bolo potrebné nahliadnuť do programu.

Program. V úlohách **P–I–2** až **P–I–4** je potrebné k riešeniu pripojiť odladený program napísaný v jazyku Pascal, C alebo C++. Program sa odovzdáva v písomnej forme (jeho výpis je teda súčasťou riešenia) i na diskete, aby bolo možné otestovať jeho funkčnosť.

Súbory na diskete pomenujte `p1x.pas`, `p1x.cpp`, kde x je číslo súťažnej úlohy. Disketu označte menom riešiteľa. Z jednej školy možno poslať všetky riešenia na jednej diskete. V tomto prípade pre každého riešiteľa vytvorte podadresár označený jeho priezviskom a disketu označte adresou školy.

V úlohe **P–I–1** je potrebné uviesť dostatočne podrobný zápis algoritmu (najlepšie zdrojový text najdôležitejších častí programu, zo zápisu možno vynechať jednoduché operácie ako sú vstupy, výstupy, implementácie jednoduchých matematických vzťahov a pod.). V úlohe **P–I–4b** nakreslite požadovanú komparátorovú sieť.

Písomnú časť riešenia vypracujte čitateľne na listy formátu A4. **Každú úlohu začnite na novom liste** a v záhlaví uveďte vaše meno, triedu, adresu školy a označenie príkladu podľa tohto letáku. Zadaní úloh nemusíte opisovať. Ak sa vám riešenie nezmestí na jeden list, uveďte na ďalších listoch vľavo hore svoje meno a označenie úlohy a očísľujte strany.

KATEGÓRIA P

P-I-1

Uvažujme štvorcovú maticu A rozmerov $n \times n$, ktorej prvky sú nuly a jednotky. Pod *riadkovou* resp. *stĺpcovou výmenou* máme na mysli výmenu ľubovoľných dvoch riadkov resp. ľubovoľných dvoch stĺpcov matice. V tejto úlohe sa zaujímame o to, či je možné postupnosťou výmen transformovať maticu A na maticu, ktorá má na hlavnej diagonále iba jednotky (hlavnú diagonálu tvoria prvky $A[1, 1], A[2, 2], \dots, A[n, n]$).

V mnohých aplikáciách (napr. riešenie sústav lineárnych rovníc) je výhodné danú maticu transformovať na ekvivalentnú maticu tak, aby hlavná diagonála obsahovala „veľké“ prvky. V tejto úlohe nuly reprezentujú „malé“ a jednotky „veľké“ prvky.

Súťažná úloha

Navrhňte čo najefektívnejší algoritmus, ktorý transformuje pomocou riadkových a stĺpcových výmen danú maticu tak, aby mala na diagonále samé jednotky, alebo zistí, že to nie je možné.

Príklad:

Matica A :

0 1 0 1
1 1 0 1
1 1 0 0
0 0 1 0

Výsledok:

1 1 0 0
1 1 0 1
0 0 1 0
0 1 0 1

Maticu možno transformovať. Stačí vymeniť tretí a štvrtý riadok a potom prvý a štvrtý stĺpec.

Príklad:

Matica A :

0 1 0
1 1 0
1 1 0

Výsledok:

Maticu nemožno transformovať.

P-I-2

Vo výskumnom ústave potrubí a rúr vyvinuli nový druh robota určeného na čistenie teplovodných potrubí. Robot sa sústavou rúr pohybuje podľa dopredu určeného postupu. Technológia čistenia je taká, že robot musí prejsť každou rúrou v sústave dvakrát (nezáleží ktorým smerom) – pri prvom prechode vykonáva chemické čistenie a pri druhom mechanické dočisťovanie. Robot môže vojsť do rúry z ľubovoľného konca, avšak ak raz vojde do rúry, musí ju celú prejsť.

Sústava rúr pozostáva z n uzlov očíslovaných od 1 po n , medzi ktorými vedie m teplovodných rúr očíslovaných od 1 po m . Každá rúra vedie medzi dvojicou uzlov a nemá žiadne odbočky alebo vetvenia. Medzi každou dvojicou uzlov vedie najviac jedna rúra. Môžete predpokladať, že sústava rúr je súvislá, t.j. robot sa môže dostať na ľubovoľné miesto sústavy. Robot začína v uzle číslo 1 a po skončení čistenia sa musí opäť do tohto uzla vrátiť.

Súťažná úloha

Napište program, ktorý pomôže plánovať trasu čistiaceho robota. Váš program načíta popis sústavy potrubí a zistí či existuje trasa, ktorá začína aj končí v uzle 1 a prechádza každou rúrou práve dvakrát. Ak takáto trasa existuje, váš program ju vypíše.

Formát vstupu. Prvý riadok vstupného súboru `rury.in` obsahuje dve kladné celé čísla n a m ($n \leq 100$), oddelené medzerou. Každý z nasledujúcich m riadkov popisuje jednu rúru. Obsahuje dve čísla oddelené medzerou – koncové uzly rúry.

Formát výstupu. Výstupný súbor `rury.out` pozostáva z jediného riadku, obsahujúceho $2m + 1$ čísel uzlov oddelených medzerami v poradí, v akom ich má robot navštíviť na navrhovanej trase. Prvé a posledné číslo na riadku musí byť 1. V prípade, že neexistuje trasa ktorá prejde každú rúru práve dvakrát, výstupný súbor bude obsahovať jediný riadok so slovom NIE. Ak existuje vhodných trás viac, vypíšete ľubovoľnú (jednu) z nich.

Príklad:

Súbor `rury.in`

```
5 6
1 3
1 4
1 5
2 4
3 5
4 5
```

Súbor `rury.out`

```
1 3 5 4 1 5 4 2 4 1 5 3 1
```

P-I-3

V rovine je daných $2n$ bodov, z toho n bielych a n čiernych. *Spravodlivá priamka* je taká priamka, ktorá:

- prechádza bodom $[0, 0]$,
- neprechádza žiadnym čiernym ani bielym bodom,
- rozdeľuje rovinu na dve polroviny, pričom v jednej z týchto polrovín je rovnako bielych bodov, ako v druhej čiernych, a naopak.

Súťažná úloha

Napište program, ktorý zo súboru `priamka.in` načíta súradnice bielych a čiernych bodov a do súboru `priamka.out` vypíše jednu spravodlivú priamku.

Môžete predpokladať, že žiadne tri zadané body neležia na jednej priamke, a že bod $[0, 0]$ neleží na žiadnej priamke určenej dvoma zadanými bodmi. Všetky zadané body majú celočíselné súradnice.

Formát vstupu. Súbor `priamka.in` na prvom riadku obsahuje číslo n . Každý z nasledujúcich $2n$ riadkov obsahuje súradnice jedného bodu oddelené medzerou. Prvých n bodov je bielych, druhých n bodov je čiernych.

Formát výstupu. Súbor `priamka.out` na prvom riadku obsahuje ľubovoľný bod rôzny od $[0, 0]$, ktorým hľadaná spravodlivá priamka prechádza. Súradnice oddelíte medzerou. Ak spravodlivá priamka pre danú množinu bodov neexistuje, vypíšete do súboru slovo NIE.

Príklad:

Súbor `priamka.in`

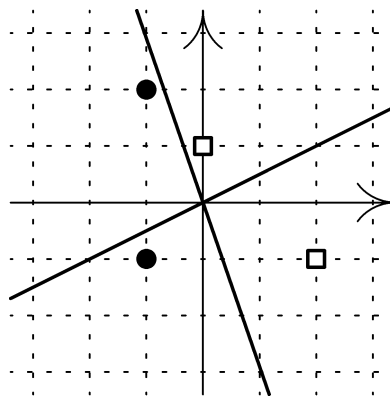
```
2
0 1
2 -1
-1 -1
-1 2
```

Súbor `priamka.out`

```
2 1
```

Iné správne riešenie

```
-1.0 2.9
```



Pomôcka: Nech $A_1 = [x_1, y_1]$, $A_2 = [x_2, y_2]$ a $A_3 = [x_3, y_3]$ sú body v rovine. Ak je hodnota $(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)$ kladná, bod A_3 leží naľavo od polpriamky A_1A_2 . Ak je táto hodnota záporná, tak leží napravo a ak je táto hodnota nulová, tak bod A_3 leží na priamke A_1A_2 .

P-I-4

Komparátorové siete

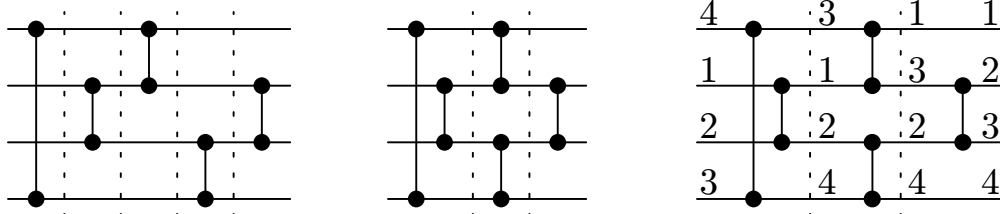
Komparátorové siete sa využívajú pri návrhu paralelných algoritmov. Tiež je jednoduché ich realizovať pomocou elektronických obvodov. *Komparátor* je jednoduché zariadenie, ktoré dostane na vstupe dve čísla, porovná ich a na vrchnom zo svojich výstupov vráti menšie z týchto dvoch čísel a na spodnom to väčšie. Z komparátorov možno zostaviť zložitejšie obvody, ktoré budeme volať komparátorové siete.

Komparátorová sieť pozostáva z n vodorovne usporiadaných *vodičov*, ktoré sú na niektorých miestach poprepájané komparátormi. Komparátory sú usporiadané do *vrstiev*, ktoré zodpovedajú krokom výpočtu. Na začiatku výpočtu (v kroku 0) sieť dostane na vstup n čísel. Po skončení kroku $k - 1$ sa výstupy z kroku $k - 1$ prenású vodičmi na komparátory vo vrstve k . Komparátor vo vrstve k spája vždy dva vodiče (nemusia byť susedné). Ak je na spodnom z nich menšia hodnota ako na vrchnom, vymení tieto hodnoty, v opačnom prípade ich nechá tak. V jednej vrstve môže byť aj viacero komparátorov (výpočet sa odohrá naraz, paralelne), ale v jednej vrstve môže jeden vodič vstupovať najviac do jedného komparátora. Po skončení celého výpočtu sú na výstupoch siete tie isté čísla ako na vstupe, iba v zmenenom poradí.

Graficky sa vodiče zobrazujú ako vodorovné čiary, komparátory ako zvislé spojnice svojich vstupných vodičov. Komparátory v jednej vrstve sa kreslia zvislo nad seba, prípadne do niekoľkých susediacich stĺpcov. Jednotlivé vrstvy oddelíme čiarkovanou čiarou. Výpočet prebieha zľava doprava.

Pri návrhu sietí sa pokúšame zostrojiť ich tak, aby čas výpočtu bol čo najmenší, t.j. aby sieť mala čo najmenej vrstiev. Druhým dôležitým kritériom je počet komparátorov (od tohoto počtu môže závisieť napríklad cena výroby siete).

Príklad.

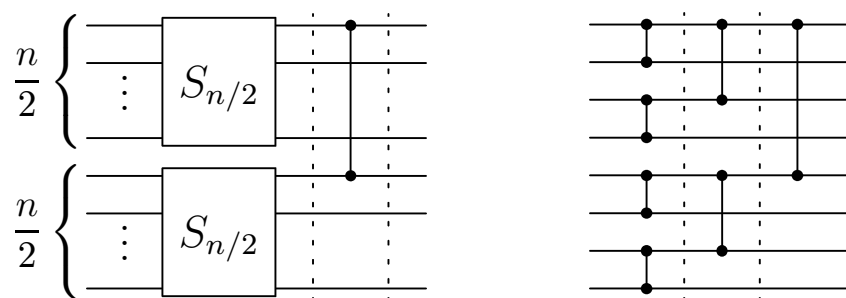


Uvažujme najľavejšiu sieť na obrázku. Táto sieť dostane štyri vstupy a vráti ich utriedené od najmenšieho po najväčší. Po prvých dvoch krokoch výpočtu bude najmenší vstup buď na prvom alebo druhom vodiči a najväčší vstup na treťom alebo štvrtom. Ďalšie dva kroky umiestnia najmenší a najväčší prvok na ich miesto a v poslednom kroku dotriedime

druhý a tretí vodič. Všimnite si, že prvý a druhý komparátor, podobne ako tretí a štvrtý, je možno zlúčiť do jednej vrstvy bez toho, aby sa zmenil výsledok výpočtu. Výsledná rýchlejšia sieť je na prostrednom obrázku. Pravý obrázok ukazuje priebeh výpočtu pre vstup 4, 1, 2, 3.

Príklad. Zostrojte sieť, ktorá na vstupe dostane n čísel a na výstupe umiestni najmenšie z týchto čísel na vodič 1 (na poradí ostatných čísel nám nezáleží). Môžete predpokladať, že n je mocnina dvoch.

Riešenie. Sieť zostrojíme rekurzívne. Označme S_n sieť, ktorá úlohu rieši pre n vstupov. Ak $n = 1$, S_n nebude obsahovať žiaden komparátor, lebo máme iba jeden vstup. Predpokladajme teda, že $n > 1$. Vstupy rozdelíme na dve polovice (hornú a dolnú). Na každú polovicu aplikujeme sieť $S_{n/2}$. Tieto dve siete polovičnej veľkosti môžu pracovať paralelne. Po skončení ich výpočtu máme na vodiči 1 najmenší z hornej polovice vstupov a na vodiči $n/2 + 1$ máme najmenší z dolnej polovice. Teraz stačí pridať jeden komparátor medzi vodičmi 1 a $n/2 + 1$ a dostaneme celkové minimum na prvom vodiči. Sieť S_n teda pozostáva z dvoch sietí $S_{n/2}$ a z jedného komparátora. Konštrukcia siete S_n je zobrazená na nasledujúcom obrázku vľavo, vpravo je príklad výslednej siete pre $n = 8$.



Všimnime si, že hĺbka rekurzie je $\log_2 n$, keďže veľkosť vstupu sa v každom rekurzívnom kroku zníži na polovicu. Každá úroveň rekurzie pridá do výslednej siete jednu vrstvu, preto sieť S_n má $\log_2 n$ vrstiev. Počet použitých komparátorov je v poslednej vrstve 1, v každej ďalšej vrstve odzadu sa vždy zdvojnásobí. Nech počet vstupov je $n = 2^k$. Potom počet použitých komparátorov je $1 + 2 + 4 + \dots + 2^{k-1} = 2^k - 1 = n - 1$ (súčet geometrickej postupnosti). Dostali sme teda sieť, ktorá má $O(\log n)$ vrstiev a používa $O(n)$ komparátorov.

Súťažné úlohy

- Napište program, ktorý bude simulovať komparátorové siete. Program vo vstupnom súbore dostane komparátorovú sieť a hodnoty vstupov. Navrhnite a popíšte vlastný vstupný formát pre zápis komparátorových sietí do súboru. Váš program by mal vypočítať výsle-

dok výpočtu komparátorovej siete a tiež by mal umožňovať grafy alebo semigrafy zobrazovať priebeh výpočtu siete. Vo vašom riešení uveďte aj príklady vstupu a výstupu z vášho programu.

- b) Na vstupe je $n - 1$ čísel utriedených od najmenšieho po najväčšie (prvých $n - 1$ vstupov). Posledný vstup obsahuje ľubovoľné číslo. Navrhnite komparátorovú sieť, ktorá zatriedi posledné číslo do tejto postupnosti (tzn. na výstupe bude všetkých n čísel v utriedenom poradí). Môžete predpokladať, že n je mocnina dvoch. Snažte sa, aby vaša sieť bola čo najrýchlejšia.

SLOVENSKÁ KOMISIA MATEMATICKEJ OLYMPIÁDY

51. ROČNÍK MATEMATICKEJ OLYMPIÁDY

Leták kategórie P

Vydala IUVENTA

pre vnútornú potrebu Ministerstva školstva SR

Autori príkladov B. Brejová, M. Forišek, M. Pál a T. Vinař

Programom L^AT_EX sadzbu pripravili T. Vinař a B. Brejová

© Slovenská komisia Matematickej olympiády, 2001