MATEMATICKÁ OLYMPIÁDA NA STREDNÝCH ŠKOLÁCH

51. ročník, školský rok 2001/2002

Zadania úloh 3. kola kategórie P

1. súťažný deň

P-III-1

Daná je matica A s m riadkami a n stĺpcami. Každý prvok a_{ij} je buď kladné celé číslo, alebo tzv. *žolík*, ktorý budeme značiť znakom $*$. Prvky v každom stĺpci je možné ľubovoľne vymieňať, avšak nie je možné vymieňať prvky z rôznych stĺpcov. Otázka znie, či je možné prvky v matici preusporiadať tak, aby každý riadok matice obsahoval neklesajúcu postupnosť, t.j. pre každý riadok i platilo $a_{i1} \leq a_{i2} \leq \ldots \leq a_{in}$. Každý žolík je pre účely porovnania možné zameniť ľubovoľným celým číslom. Napríklad, riadok $(1, *, 4, *)$ je neklesajúci, pretože prvý žolík $(*)$ je možné zameniť číslom (povedzme) 3 a druhý žolík $(*)$ číslom (povedzme) 77 tak, že výsledný riadok $(1, 3, 4, 77)$ je neklesajúci. Riadok $(5, *, *, 4)$ nie je neklesajúci, pretože neexistujú čísla x, y ktorými by bolo možné zameniť žolíky tak, aby $5 \leq x \leq y \leq 4$.

Súťažná úloha

Navrhnite program, ktorý pre danú maticu A rozhodne, či je možné spermutovať prvky v každom stĺpci tak, aby každý riadok matice tvoril neklesajúcu postupnosť. Ak to možné je, váš program má jednu takúto maticu vypísať.

Príklad:

Vstup:

$m = 4$, $n = 4$

```
 1  10   8  12
 *   9  10   *
 8   *  11  10
12   *   *  10
```

Výstup:

Áno. Jedno možné preusporiadanie prvkov:

```
 1   *   8  10
12   *   *   *
 *  10  11  12
 8   9  10  10
```

Vstup:
$m = 3$, $n = 4$
```
*   *   *   *
*   *   *   *
4   3   2   1
```

Výstup:
Nie.

P-III-2

V istom lyžiarskom stredisku sa pravidelne konajú preteky. Na svahu je natrvalo vyznačených N orientačných bodov. Organizátori pretekov sa vždy snažia zvoliť trochu inú trasu, ale nesmú meniť pozície orientačných bodov. Navyše, *vyhovujúca trasa* musí spĺňať nasledujúce podmienky:

- začína v najvyššie položenom orientačnom bode a končí v najnižšie položenom orientačnom bode,
- môže prechádzať cez niekoľko ďalších orientačných bodov,
- medzi každými dvoma orientačnými bodmi vedie trasa po priamke,
- každý ďalší orientačný bod je položený nižšie ako predchádzajúci,
- v žiadnom orientačnom bode nemení trasa smer o viac ako 45°.

Úlohou je zistiť, koľko vyhovujúcich trás sa dá v lyžiarskom stredisku zostaviť.

Súťažná úloha

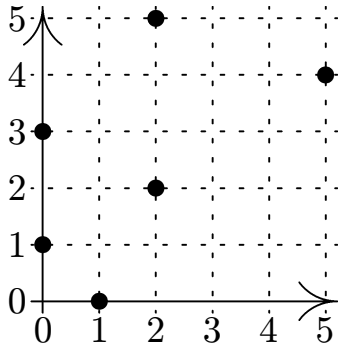
Na vstupe je počet orientačných bodov N a čísla (x_1, y_1), (x_2, y_2), …, (x_N, y_N) určujúce polohu jednotlivých orientačných bodov na svahu. Svah je jednoducho obdĺžnik, zvažujúci sa zhora nadol. Číslo x_i určuje vzdialenosť i-teho orientačného bodu od ľavého okraja svahu a y_i je vzdialenosť tohto bodu od dolného okraja svahu. Čím väčšia je y-ová súradnica bodu, tým má bod väčšiu nadmorskú výšku.

Napíšte program, ktorý nájde počet vyhovujúcich trás na lyžiarskom svahu. Môžete predpokladať, že žiadne dva orientačné body nemajú rovnakú y-ovú súradnicu a že žiadne tri orientačné body neležia na jednej priamke.

Pomôcka: Môžete predpokladať, že máte k dispozícii funkciu `uhol`(x, y), ktorá vráti uhol medzi bodom (x, y) a x-ovou osou. Presnejšie povedané, je to uhol, o ktorý treba otočiť v protismere hodinových ručičiek polpriamku začínajú v bode $(0, 0)$ a prechádzajúcu cez bod $(1, 0)$, aby prechádzala cez (x, y). Výsledné číslo je uhol v stupňoch z intervalu $\langle 0, 360)$.

Príklad:

Vstup:
$N = 6$,
body:
$(2, 5), (2, 2), (5, 4), (0, 3), (0, 1), (1, 0)$



Výstup:
Existujú 3 vyhovujúce trasy.

Poznámka:
Vyhovujúce trasy sú:
$(2, 5), (0, 3), (0, 1), (1, 0)$
$(2, 5), (2, 2), (1, 0)$
$(2, 5), (1, 0)$

P-III-3

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.

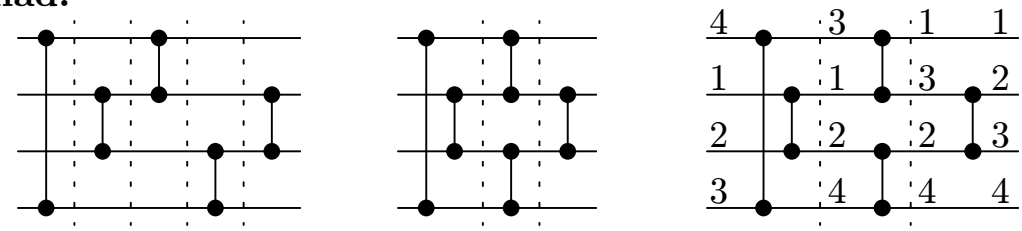
Komparátorová sieť pozostáva z n vodorovne usporiadaných *vodičov*, ktoré sú na niektorých miestach poprepájané komparátormi. Komparátory sú usporiadané do *vrstiev*, ktoré zodpovedajú krokom výpočtu. Na začiatku výpočtu (v kroku 0) sieť dostane na vstup n čísel. Po skončení kroku $k - 1$ sa výstupy z kroku $k - 1$ prenesú 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.

Gra cky 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).

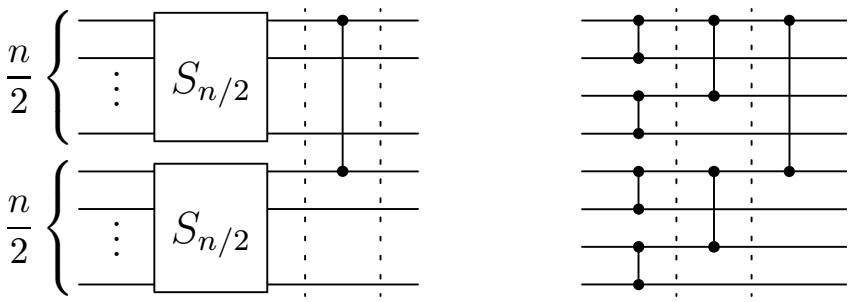
Príklad.



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 $1 + 2 + 4 + \ldots + 2^{k-1} = 2^k - 1 = n - 1$ (súčet geometrickej postupnosti). Dostali sme teda sieť, ktorá má $O(\log n)$ vrstiev a používa $O(n)$ komparátorov.

Súťažná úloha

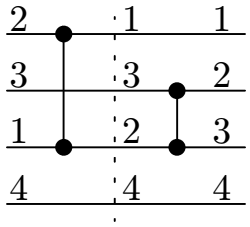
Daná je jedna konkrétna permutácia čísel $1, 2, \ldots, n$, t.j. postupnosť čísel $a_1, a_2, \ldots, a_n$, v ktorej sa každé z čísel $1, 2, \ldots, n$ nachádza práve raz. Dopredu vieme, že naša komparátorová sieť bude mať n vodičov a na vstupe bude i-ty vodič siete obsahovať číslo a_i. Treba navrhnúť komparátorovú sieť, ktorá tento konkrétny vstup utriedi.

Keďže výsledná sieť sa môže líšiť pre rôzne permutácie, vašou úlohou je napísať algoritmus, ktorý na vstupe dostane číslo n a permutáciu $a_1, a_2, \ldots, a_n$ a na výstupe vypíše výslednú sieť, ktorá túto permutáciu utriedi. Výslednú sieť vypíšte ako postupnosť komparátorov v jednotlivých vrstvách zľava doprava, komparátor vypíšte ako dvojicu vodičov, ktoré spája.

Váš algoritmus má pracovať v čase, ktorý je polynomiálny v závislosti od n. Snažte sa, aby sieť, ktorú algoritmus vypíše, mala málo vrstiev (a pokiaľ možno aj malý počet komparátorov). Existuje efektívny algoritmus, ktorý pre každú permutáciu vypíše sieť s $O(n)$ komparátormi a $O(\log n)$ vrstvami.

Príklad

Vstup:
$n = 4$, $a_1 = 2, a_2 = 3, a_3 = 1, a_4 = 4$.
Výstup:
Vstup je možné utriediť napríklad sieťou:
Prvá vrstva: komparátor $(1, 3)$
Druhá vrstva: komparátor $(2, 3)$



SLOVENSKÁ KOMISIA MATEMATICKEJ OLYMPIÁDY

51. ROČNÍK MATEMATICKEJ OLYMPIÁDY

Zadania 3. kola kategórie P

1. súťažný deň

Vydala IUVENTA
pre vnútornú potrebu Ministerstva školstva SR
Autori príkladov B. Brejová, M. Forišek, M. Pál a T. Vinař
Sadzba programom LaTeX
Zodpovedný redaktor M. Forišek

MATEMATICKÁ OLYMPIÁDA NA STREDNÝCH ŠKOLÁCH

51. ročník, školský rok 2001/2002

Zadania úloh 3. kola kategórie P

2. súťažný deň

P-III-4

Program: `rury.pas/rury.cpp`
Vstup: zo štand. vstupu
Výstup: na štand. výstup

Vo výskumnom ústave potrubí a rúr majú nový problém. Dostali zákazku od Vodární a kanalizácii mesta Blatislavy, ktoré potrebujú prečistiť celú Blatislavskú kanalizáciu od nánosov blata. Technológia čistenia rúr od blata je jednoduchá – robot musí prejsť každou rúrou práve raz (keby šiel už vyčistenou rúrou, hrozí, že ju silné čistiace prostriedky poškodia). Robot môže vojsť do rúry z ľubovoľného konca, avšak ak raz vojde do rúry, musí ju celú prejsť.

Pracovníci výskumného ústavu si rýchlo uvedomili, že za takýchto podmienok sa môže stať, že nech by pustili čistiaceho robota po akejkoľvek dráhe, nepodarí sa mu vyčistiť celú kanalizáciu. Preto sa rozhodli, že do kanalizácie pošlú naraz viacero robotov a naprogramujú ich tak, aby roboti spolu vyčistili celú kanalizáciu.

Čistiaci robot je však zatiaľ nepredstaviteľne drahý, preto ich chcú pracovníci ústavu poslať do kanalizácie čo najmenej. Ale koľko ich treba a ako ich majú poslať? To už je úloha pre vás.

Kanalizácia pozostáva z n uzlov očíslovaných od 1 po n, medzi ktorými vedie m rúr. Každá rúra vedie medzi dvojicou uzlov a nemá žiadne odbočky alebo vetvenia. Medzi každou dvojicou uzlov vedie najviac jedna rúra.

Súťažná úloha

Napíšte program, ktorý pomôže plánovať trasy čistiacich robotov. Váš program načíta popis kanalizácie, zistí, koľko najmenej robotov stačí na vyčistenie kanalizácie a navrhne ich trasy tak, aby dokopy roboti prešli každou rúrou práve raz.

Formát vstupu. Prvý riadok vstupu obsahuje dve kladné celé čísla n a m ($1 \leq n \leq 200$), oddelené medzerou. Každý z nasledujúcich m riadkov popisuje jednu rúru. Obsahuje dve čísla oddelené medzerou – koncové uzly rúry.

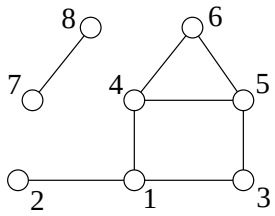
Formát výstupu. Na prvom riadku výstupu je jediné celé číslo R – najmenší počet robotov, ktorí spolu dokážu vyčistiť celú kanalizáciu. Nasleduje R riadkov, i-ty z nich popisuje trasu i-teho spomedzi robotov.

Nech i-ty robot začína trasu v uzle a_1, odtiaľ prejde rúrou do uzlu a_2, odtiaľ do a_3, atď. a skončí v uzle a_k. Potom i-ty z týchto riadkov by mal obsahovať čísla a_1, a_2, …, a_k v tomto poradí, oddelené od seba medzerami.

Pokiaľ je možných trás robotov viac, stačí nájsť a vypísať jedno ľubovoľné riešenie.

Príklad:

Vstup	**Výstup**
8 8 1 2 1 3 1 4 3 5 4 5 4 6 5 6 7 8	3 2 1 3 5 4 1 4 6 5 8 7

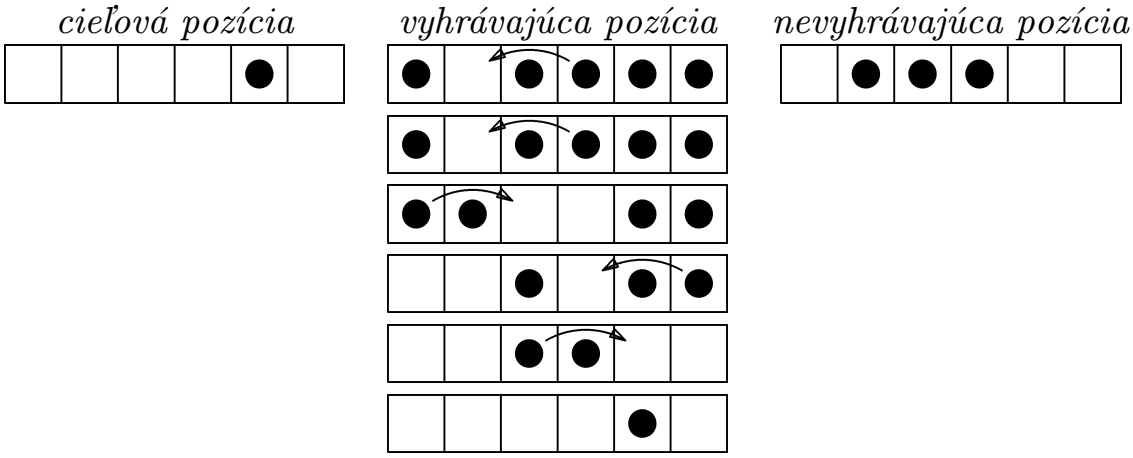


P-III-5

Program: `kamene.pas/kamene.cpp`
Vstup: zo štand. vstupu
Výstup: na štand. výstup

Uvažujme nasledujúcu hru pre jedného hráča. Hrací plán pozostáva z K políčok v jednom riadku. Na tomto pláne sa nachádza niekoľko hracích kameňov (na každom políčku najviac jeden kameň). Zadaná je *cieľová pozícia*, t.j. určité rozostavenie kameňov, ktoré je potrebné dosiahnuť. V jednom ťahu môžeme zobrať kameň na políčku p a preskočiť ním susedný kameň. Presnejšie, ak je na políčku napravo (resp. naľavo) od p kameň a nasledujúce políčko napravo (resp. naľavo) je voľné, môžeme preskočiť kameňom z políčka p na toto voľné políčko. Kameň, ktorý takto preskočíme, je odobraný z hry. Cieľom hry je postupným skákaním dosiahnuť cieľovú pozíciu. Pozíciu, z ktorej existuje postupnosť ťahov, ktorými sa dostaneme do cieľovej pozície, nazývame *vyhrávajúca pozícia*.

Napríklad, nech pozícia na nasledujúcom obrázku vľavo je cieľová. V strednej časti obrázku je vyhrávajúca pozícia pre túto cieľovú pozíciu aj s príslušnou postupnosťou ťahov. Pozícia v pravej časti obrázku nie je vyhrávajúca, lebo v prvom ťahu zjavne musíme skákať stredným kameňom, potom nám ostanú dva kamene, medzi ktorými je dve políčka medzera.



Súťažná úloha

Napíšte program, ktorý načíta celé čísla K a N a cieľovú pozíciu a vypíše počet (rôznych) vyhrávajúcich pozícií pozostávajúcich z najviac N kameňov.

Formát vstupu. Vstup obsahuje na prvom riadku celé čísla K a N oddelené medzerou. Na druhom riadku je zadaná cieľová pozícia ako postupnosť K núl a jednotiek oddelených medzerami, kde jednotka znamená

pozíciu, na ktorej je kameň a nula znamená pozíciu, na ktorej nie je kameň. Môžete predpokladať, že $1 \leq N \leq K \leq 100$.

Formát výstupu. Výstup obsahuje jediný riadok a na ňom jedno celé číslo určujúce počet vyhrávajúcich pozícií pozostávajúcich z najviac N kameňov. Môžete predpokladať, že výsledné číslo nepresiahne 10 000.

Príklad:

Vstup	**Výstup**
`6 3` `0 0 1 1 0 0`	`3`

Vstup	**Výstup**
`8 5` `1 0 0 0 0 0 0 1`	10

SLOVENSKÁ KOMISIA MATEMATICKEJ OLYMPIÁDY

51. ROČNÍK MATEMATICKEJ OLYMPIÁDY

Zadania 3. kola kategórie P

2. súťažný deň

Vydala IUVENTA
pre vnútornú potrebu Ministerstva školstva SR
Autori príkladov B. Brejová, M. Forišek, M. Pál a T. Vinař
Sadzba programom LaTeX
Zodpovedný redaktor M. Forišek