

MATEMATICKÁ OLYMPIÁDA NA STREDNÝCH ŠKOLÁCH

50. ROČNÍK, školský rok 2000/2001

Zadania úloh 2. kola kategórie P

Druhé kolo 50. ročníka MO kategórie P sa koná v utorok 9. 1. 2001 v dopoludňajších hodinách. Na riešenie úloh máte 4 hodiny čistého času. Riešenie každého príkladu musí obsahovať (pokiaľ nie je v zadaní uvedené ináč):

- **Popis riešenia**, to znamená slovný popis použitého algoritmu, argumenty zdôvodňujúce jeho správnosť (prípadne dôkaz správnosti algoritmu), diskusiu o efektivite vášho riešenia (časová a pamäťová zložitost'). Slovný popis riešenia musí byť jasný a zrozumiteľný i bez nahliadnutia do samotného zápisu algoritmu (do programu).
- **Program**. V úlohách **P-II-1**, **P-II-2** a **P-II-3** treba uviesť dostatočne podrobný zápis algoritmu, najlepšie v tvare zdrojového textu najdôležitejších častí programu v jazyku Pascal alebo C. Zo zápisu môžete vynechať jednoduché operácie ako vstupy, výstupy, implementáciu jednoduchých matematických vzťahov a pod. V úlohe **P-II-4** uveďte namiesto zdrojového textu zápis dlaždicového programu.

Hodnotí sa nielen správnosť programu, ale tiež kvalita popisu riešenia a efektivita zvoleného algoritmu.

KATEGÓRIA P

P-II-1

(Trojuholníky)

Jožko našiel na povale u babičky krabicu s drevenými paličkami. Začal sa s nimi hrať a zostavovať z nich trojuholníky rôznych tvarov. Uvidel ho jeho otec a začalo ho zaujímať, koľko rôznych trojuholníkov sa dá z týchto paličiek zostaviť. Vašou úlohou je pomôcť mu nájsť odpoveď na túto otázku.

Súťažná úloha:

Váš program na vstupe dostane celé číslo N (počet paličiek) a ďalej N navzájom rôznych kladných čísel d_1 až d_N (dĺžky paličiek). Úlohou vášho programu je určiť počet trojíc celých čísel i, j, k takých, že $1 \leq i < j < k \leq N$ a čísla d_i , d_j a d_k spĺňajú trojuholníkovú nerovnosť. (Teda $d_i < d_j + d_k$, $d_j < d_i + d_k$ a $d_k < d_i + d_j$.)

Príklad:

Pre $N = 5$ a $d_1 = 5.5$, $d_2 = 1.5$, $d_3 = 2.0$, $d_4 = 2.5$, $d_5 = 7.5$ váš program vráti číslo 2, lebo trojuholník sa dá zostaviť len z trojice paličiek s dĺžkami d_1 , d_4 a d_5 a z trojice s dĺžkami d_2 , d_3 a d_4 . Všimnite si, že ani trojica paličiek s dĺžkami d_1 , d_3 a d_5 netvorí trojuholník.

Spoločnosť pre rovnoprávnosť robotov a ľudí sa snaží vyrobiť robota, ktorý by sa mohol sám pohybovať v miestnosti s prekážkami. Bohužiaľ tejto spoločnosti chýba softvérový expert, a preto sa rozhodla, že vás požiada o pomoc.

Robot sa má pohybovať v obdĺžnikovej miestnosti, ktorá má na podlahe nakreslenú štvorčekovú sieť. Na niektorých políčkach v miestnosti sú postavené prekážky – na tieto políčka nesmie robot počas svojho pohybu vstúpiť. Robot sa môže po miestnosti pohybovať iba rovnobežne s niektorou zo stien.

Spoločnosť však trpí aj nedostatkom schopných technikov, a tak sú možnosti pohybu robota po miestnosti značne limitované. Robot rozpoznáva len tri príkazy: **Krok**, **Doľava** a **Doprava**. Po prijatí príkazu **Krok** sa robot presunie na susedné políčko v smere, do ktorého je práve natočený. Po prijatí príkazu **Doľava** sa otočí o 90 stupňov doľava a po prijatí príkazu **Doprava** sa otočí o 90 stupňov doprava.

Súťažná úloha:

Vašou úlohou je napísať program, ktorý na vstupe dostane rozmery štvorcovej siete na podlahe miestnosti (M a N), súradnice políčka, na ktorom sa robot práve nachádza a súradnice políčka, na ktoré sa robot má presunúť. Súradnice políčok číslujeme od 1, prvá súradnica je riadok, druhá stĺpec. Ľavý horný roh miestnosti má súradnice $[1, 1]$ (viď príklad nižšie). Každý z nasledujúcich M riadkov obsahuje N čísel 0 alebo 1. Ak je j -te číslo na i -tom radku 1, potom je na políčku so súradnicami $[i, j]$ prekážka, ak je toto číslo 0, tak tam prekážka nie je. Úlohou je nájsť a vypísať postupnosť príkazov, podľa ktorých robot dôjde zo začiatočného políčka na cieľové. Úloha však má ešte jeden háčik: Spracovanie príkazov **Doľava** a **Doprava** je časovo veľmi náročné. Preto by ich vami vytvorená postupnosť inštrukcií pre robota mala obsahovať čo najmenej. Počet príkazov **Krok** môže byť ľubovoľný. Začiatočné natočenie robota si môžete zvoliť. V prípade, že robot nemôže prejsť zo začiatočného políčka na cieľové, vypíšete o tom vhodnú správu.

Príklad:

Predstavme si miestnosť so štvorcovou sieťou 4×8 z nasledujúceho obrázku:

	○	○	○	○	○		
	○				○	○	
S			○				C

Úlohou je presunúť robota z políčka označeného S (so súradnicami $[4, 1]$) na políčko označené C (so súradnicami $[4, 8]$). Vstup vášho programu by teda vyzeral nasledovne:

```
4 8
4 1
4 8
0 0 0 0 0 0 0 0
0 1 1 1 1 1 0 0
0 1 0 0 0 1 1 0
0 0 0 1 0 0 0 0
```

Optimálny program na presun robota je nasledovný (začiatočné natočenie robota je dohora):

Krok Krok Krok Doprava Krok Krok Krok Krok

Krok Krok Krok Doprava Krok Krok Krok

Pri tejto ceste robot urobí 13 krokov a dvakrát sa otočí. Všimnite si, že existuje kratšia cesta s 9 krokmi a 4 otočeniami – táto cesta je síce kratšia, ale podľa zadania nie je optimálna, lebo sa počas nej robot viackrát otočí.

P-II-3

(Jožko)

Malému Jožkovi (ktorého poznáte už zo zadania 1. úlohy) sa jedného dňa dostali do rúk nožnice. A keďže Jožko zdedil talent po ockovi, ktorý je moderným maliarom, rozhodol sa vylepšiť jeden jeho obraz v tvare konvexného n -uholníka. Vybral si dva jeho vrcholy a obraz prestrihol po ich spojnici. Potom si na jednej zo vzniknutých častí opäť vybral dva vrcholy a časť opäť prestrihol. Keď sa takto Jožko chvíľu hral, prichytil ho ocko, nožnice mu nekompromisne zobral a začal zachraňovať, čo sa dalo. Po chvíli zistil, že obraz už do pôvodného stavu nezloží. Rozhodol sa teda, že aspoň nájde časť, ktorá má najviac vrcholov a tú vystaví na svojej najbližšej výstave ako miniatúru. A práve s hľadaním tejto časti mu máte pomôcť.

Súťažná úloha:

Navrhните čo najefektívnejší algoritmus, ktorý dostane na vstupe počet vrcholov pôvodného obrazu n , počet Jožkových prestrihnutí k a popis jednotlivých prestrihnutí a na základe týchto údajov určí počet vrcholov tej výslednej časti, ktorá ich má najviac. Každé prestrihnutie je popísané dvojicou čísel (a_i, b_i) , čo sú čísla vrcholov v pôvodnom n -uholníku, medzi ktorými Jožko strihal. Vrcholy n -uholníka sú číslované po obvode číslami 1 až n . Snažte sa, aby časová ani pamäťová zložitosť vášho programu nezávisela na počte vrcholov obrazu.

Príklad:

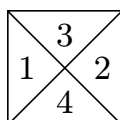
Pre $n = 10$, $k = 3$ a prestrihnutia $(1, 8)$, $(7, 5)$ a $(4, 2)$ má najväčšia časť 6 vrcholov.

P-II-4

(Dlaždice)

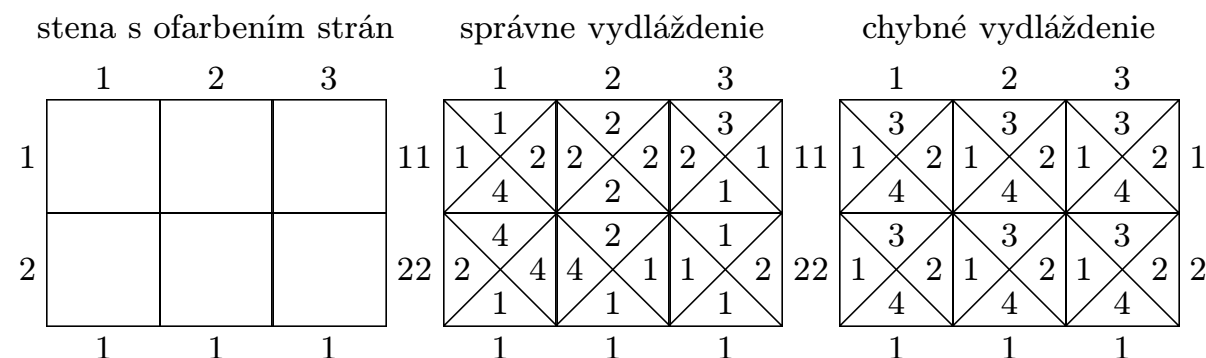
(De nícia dlaždicových programov je rovnaká ako v domácom kole, len príklad ich použitia je zložitejší a ukazuje, ako možno využívať viac riadkov dlaždíc.)

Najprv niekoľko de nícií: *Dlaždice* sú rovnako veľké štvorce s ofarbenými hranami. Konkrétne priradeniu farieb hranám dlaždice budeme hovoriť *typ* dlaždice a budeme ho zapisovať ako usporiadanú štvoricu (l, p, h, d) udávajúcu farby v poradí ľavej, pravej, hornej a dolnej hrany. Aby sme si uľahčili prácu, budeme farby označovať rôznymi symbolmi — písmenami, číslami a podobne. Dlaždica typu $(1, 2, 3, 4)$ bude teda vyzeráť nasledovne:



Priestor, ktorý budeme dláždiť (budeme mu hovoriť *stena*), má tvar obdĺžnika o veľkosti $m \times n$ (m aj n sú prirodzené čísla; jednotkou dĺžky nech je dĺžka hrany dlaždice). Strany obdĺžnika sú rozdelené na úseky jednotkovej dĺžky a každému úseku je opäť priradená farba. Naším cieľom je *vydláždiť* stenu dlaždicami tak, aby v každom z $m \cdot n$ jednotkových štvorcov steny bola umiestená práve jedna dlaždica, susedné dlaždice se dotýkali vždy hranami tej istej farby a rovnako krajné dlaždice priliehali k okraju steny vždy hranou takej farby, akú má aj príslušný úsek okraja steny. Dlaždice nie je povolené otáčať.

Príklad:



Pomocou dláždenia môžeme ľahko riešiť úlohy, ktorých výsledkom je odpoveď ÁNO alebo NIE: zostavíme vhodnú množinu typov dlaždíc (tá je pre daný problém pevná – nezávisí na vstupe), vezmeme vhodne veľkú stenu, jej horný okraj ofarbíme podľa vstupu nášho problému, ostatné okraje ponecháme jednofarebné a budeme sa pýtať, či je túto stenu možné vydláždiť alebo nie. Pritom chceme, aby tento výsledok bol zhodný s riešením našej úlohy.

Aby sme sa nemuseli zaoberať tým, ako presne veľkú stenu máme zvoliť pre ten či onen vstup úlohy, budeme šírku steny voliť vždy rovnakú, ako je dĺžka vstupu (horný okraj teda bude celý zaplnený vstupom), zatiaľ čo výšku steny použijeme najmenšiu, pre ktorú existuje vydláždenie s použitím našej sady dlaždíc.

Keď tento spôsob počítania porovnáme s klasickým programovaním, zistíme, že zvolená sada dlaždíc tvorí v našom modeli niečo podobné programu a potrebná výška steny vzdialene odpovedá dobe behu výpočtu – budeme sa preto snažiť, aby v našich riešeniach bola čo najmenšia.

Formálne povedané, *dlaždicový program* je usporiadaná štvorica

$$D = (T, l_0, p_0, d_0),$$

kde T je konečná množina typov dlaždíc $\{(l_1, p_1, h_1, d_1), \dots, (l_k, p_k, h_k, d_k)\}$ a l_0 , p_0 a d_0 sú okrajové farby. *Rozhodovacou úlohou* $P(x)$ rozumieme úlohu zistiť, či vstup x (konečná postupnosť symbolov, resp. farieb z vopred určenej konečnej množiny) má požadovanú vlastnosť P . Hovoríme, že dlaždicový program *rieši rozhodovaciu úlohu* $P(x)$, ak platí, že $P(x) = \text{ÁNO}$ práve vtedy, keď existuje $v > 0$ také, že je možné vydláždiť dlaždicami typov obsiahnutých v množine T stenu veľkosti $|x| \times v$ s hornou hranou ofarbenou vstupom x a ľavou, pravou a dolnou hranou ofarbenou postupne farbami l_0 , p_0 a d_0 . Z každého typu je možné použiť ľubovoľne mnoho dlaždíc. *Zložitostou* programu D pre daný vstup x nazveme najmenšie v , pre aké to je možné; pokiaľ také v neexistuje, a teda $P(x) = \text{NIE}$, definujeme zložitost ako nulovú. Zložitost programu je funkcia dĺžky vstupu n , ktorej hodnota udáva maximum zo zložitostí programov pre jednotlivé vstupy tejto dĺžky.

Príklad:

Skonstruujme dlaždicový program, ktorý bude overovať, či je daná postupnosť tvorená prirodzenými číslami x_1, \dots, x_n ($0 \leq x_i \leq 9$) *vyvážená*, tzn. či obsahuje rovnaký počet párných a nepárných čísel.

Myšlienka nášho riešenia je veľmi jednoduchá: zostrojíme sadu dlaždíc, ktorá bude umožňovať práve také vydláždenia, v ktorých v každom riadku prepíšeme jedno párne a jedno nepárne číslo na \bullet . Dolný okraj steny ofarbíme tiež farbou \bullet . Keďže ofarbenie spodného okraja je vyvážené a vydláždenie každého riadku vyváženost zachováva, akákoľvek vstupná postupnosť, pre ktorú existuje vydláždenie, je nutne vyvážená. A naopak, ak máme vyváženú postupnosť, potom ľahko overíme, že vydláždenie existuje: kým ešte máme nejaké čísla, vyberieme si ľubovoľné párne a ľubovoľné nepárne číslo (z vyváženosti vieme, že také čísla v postupnosti určite sú), tie jedným riadkom prepíšeme na \bullet a toto opakujeme tak dlho ($n/2$ -krát), kým neprepíšeme všetky čísla. Ak sa nám teda podarí takýto dlaždicový program zostrojiť, bude zadanú úlohu riešiť so zložitostou $O(n)$.

Hľadaný program môže vyzeráť napríklad nasledovne:

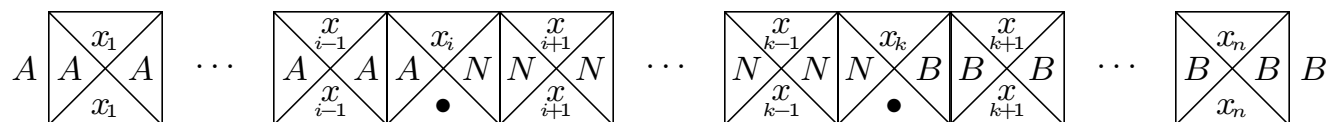
$$T = \left\{ \begin{array}{c} \begin{array}{|c|c|} \hline x & \\ \hline A & A \\ \hline & x \\ \hline \end{array}, \begin{array}{|c|c|} \hline x & \\ \hline P & P \\ \hline & x \\ \hline \end{array}, \begin{array}{|c|c|} \hline x & \\ \hline N & N \\ \hline & x \\ \hline \end{array}, \begin{array}{|c|c|} \hline x & \\ \hline B & B \\ \hline & x \\ \hline \end{array}, \begin{array}{|c|c|} \hline p & \\ \hline A & P \\ \hline & \bullet \\ \hline \end{array}, \begin{array}{|c|c|} \hline n & \\ \hline A & N \\ \hline & \bullet \\ \hline \end{array}, \begin{array}{|c|c|} \hline p & \\ \hline N & B \\ \hline & \bullet \\ \hline \end{array}, \begin{array}{|c|c|} \hline n & \\ \hline P & B \\ \hline & \bullet \\ \hline \end{array} \right\};$$

$$x \in \{0, \dots, 9, \bullet\}, p \in \{0, 2, 4, 6, 8\}, n \in \{1, 3, 5, 7, 9\}$$

Ľavý okraj ofarbíme farbou A , pravý farbou B a dolný farbou \bullet . Z týchto dlaždíc je možné konštruovať jedine riadky typu

$$A \begin{array}{|c|c|} \hline x_1 & \\ \hline A & A \\ \hline & x_1 \\ \hline \end{array} \cdots \begin{array}{|c|c|c|} \hline x_{i-1} & x_i & x_{i+1} \\ \hline A & P & P \\ \hline x_{i-1} & \bullet & x_{i+1} \\ \hline \end{array} \cdots \begin{array}{|c|c|c|} \hline x_{k-1} & x_k & x_{k+1} \\ \hline P & B & B \\ \hline x_{k-1} & \bullet & x_{k+1} \\ \hline \end{array} \cdots \begin{array}{|c|c|} \hline x_n & \\ \hline B & B \\ \hline & x_n \\ \hline \end{array} B$$

kde x_i je párne a x_k nepárne, prípadne



pre x_i nepárne a x_k párne. To sú presne riadky, ktoré sme potrebovali.

Súťažná úloha:

Zostrojte dlaždicový program, ktorý o danej postupnosti prirodzených čísel x_1, x_2, \dots, x_n ($0 \leq x_i \leq 9$) rozhodne, či je symetrická, t.j. či $x_1 = x_n, x_2 = x_{n-1}, \dots, x_i = x_{n-i+1}, \dots, x_n = x_1$.

SLOVENSKÁ KOMISIA MATEMATICKEJ OLYMPIÁDY

50. ROČNÍK MATEMATICKEJ OLYMPIÁDY

Zadania 2. kola kategórie P

Vydala IUVENTA – zariadenie pre voľný čas detí, mládeže i dospelých MŠ SR
pre vnútornú potrebu Ministerstva školstva SR
Programom \TeX sadzbu pripravil Michal Forišek

Autori príkladov: Jan Kára
Daniel Kráľ
Martin Mareš

© Slovenská komisia matematickej olympiády, 2000