

## Úlohy 2. kola letnej časti

Termín odoslania riešení tejto série je pondelok **21. mája 2012.**

### 1. Zmätený bocian

kat. Z; 7 b za popis, 3 b za program

Bocianí bratia sa vybrali do mimoriadne pompézneho lunaparku. Je postavený na rozľahlej lúke a každá atrakcia stojí osobitne od ostatných. Medzi jednotlivými atrakciami premávajú vláčiky po všakovakých klukatých trasách pomedzi fontány, bistrá a kletky s papagájmi. Vláčiky premávajú len jednosmerne: vždy od jednej atrakcie ku druhej. Z každej atrakcie vychádza práve jeden vláčik (k nejakej inej atrakcii).

Bociany sú veľmi rozmarné, a tak sa často stáva, že ich nejaká atrakcia omrzí – vtedy zatúžia po zmene. Preto by im prišlo vhod, kebyže sa z každého miesta lunaparku vedia dostať do ktoréhokoľvek iného. Takáto cesta môže pokojne viesť cez kopu ďalších atrakcií – aspoň sa cestou ešte viac povožia.

#### Úloha

Zistite, či sa dá z každej atrakcie lunaparku vláčikmi dostať do všetkých ostatných atrakcií.

#### Formát vstupu

V prvom riadku vstupu je zadaný počet atrakcií  $n$  ( $2 \leq n \leq 10^6$ ). Jednotlivé atrakcie sú označené číslami  $1, 2, \dots, n$ .

V druhom riadku je  $n$  čísiel –  $i$ -te z nich udáva, kam odchádza vláčik z atrakcie číslo  $i$ .

#### Formát výstupu

Ak sa dá dostať z každej atrakcie ku každej inej, vypíšte „Ano“, ináč vypíšte „Nie“.

#### Priklady

vstup

```
3
2 3 2
```

výstup

Nie

Od atrakcií 2 a 3 sa nedá dostať k atrakcii 1.

vstup

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

výstup

Ano

### 2. Zúfalstvo z postupností

kat. Z; 5 b za popis, 5 b za program

Marek Ospalý, Lin Jahodová, Miško Zelený a jeho princeznička slečna L. sú na dovolenke na horách.

Každý večer si rozložia mapu a rozhodujú sa, na ktorý kopec sa vydajú nasledujúci deň. Lenže také vyberanie túry nie je jednoduchá záležitosť, teda aspoň nie pre tamtú štvoricu. Fascinujú ich totiž číselné postupnosti a podľa nich si aj vyberajú túru. Našťastie majú so sebou instantného Janíčka v orákulu a on im vie povedať postupnosť nadmorských výšok po každom metri trasy.

Miško je viac informatikom ako matematikom, preto sa mu páčia aj aritmetické, aj geometrické postupnosti. Slečna L. je viac matematickou ako informatičkou (zatiaľ), preto sa jej páčia geometrické postupnosti. Lin je opica<sup>1</sup> a má veľmi úzky okruh obľúbených postupností. Aby sa jej postupnosť páčila, musí byť aritmetická a zároveň aj geometrická. Marek o sebe tvrdí, že si názor na postupnosti ešte nevytvoril, preto mu nie je dobrá žiadna.<sup>2</sup>

## Úloha

Rozpoznávať postupnosti je vcelku nudná činnosť. Našej štvorici výletníkov by sa preto zišiel program, ktorý pre Janíčkovu postupnosť nadmorských výšok vypíše zoznam účastníkov takejto túry.

Postupnosť je aritmetická, ak sa dá zapísať v tvare  $a, a + d, a + 2d, \dots, a + (n - 1)d$  pre nejaké  $a, d \in \mathbb{R}$ .

Postupnosť je geometrická, ak sa dá zapísať v tvare  $a, aq, aq^2, aq^3, \dots, aq^{n-1}$  pre nejaké  $a, q \in \mathbb{R}$ .

## Formát vstupu

Na prvom riadku je číslo  $1 < n < 10^6$ , určujúce počet prvkov postupnosti. Na druhom riadku nasleduje  $n$  čísel oddelených medzerami. Prvky sú celé čísla, ktoré sa zmestia do 32-bitovej premennej so znamienkom.

## Formát výstupu

Na prvý riadok vypíšte slovo „ano“/„nie“ podľa toho, či je postupnosť aritmetická. Na druhý riadok vypíšte slovo „ano“/„nie“ podľa toho, či je postupnosť geometrická.

## Príklad

vstup	výstup
6 -8 1 1000 32 238 -47	nie nie

*Postupnosť nie je aritmetická ani geometrická. Taký výlet by sa nikomu nepáčil.*

vstup	výstup
7 5 8 11 14 17 20 23	ano nie

*Aritmetická postupnosť, za sebou idúce členy sa od seba líšia o 3. Páči sa iba Miškovi.*

vstup	výstup
3 8 24 72	nie ano

*Geometrická postupnosť, za sebou idúce členy sa líšia trojnásobne. Lin Jahodovej sa nepáči, takže Miško so slečnou L. budú mať romantický výlet :-)*

<sup>1</sup><https://plus.google.com/116535840236499706820/posts>

<sup>2</sup>To je len zámienka. V skutočnosti je lenivý a nikam sa mu nechce chodiť.

vstup

výstup

```
4
99 0 0 0
```

```
nie
ano
```

*Táto postupnosť je geometrická, pretože jej členy môžeme zapísať ako  $99, 99 \cdot 0, 99 \cdot 0^2, 99 \cdot 0^3$ . Páči sa Miškovi a slečne L.*

### 3. Zábavná hra

kat. Z; 7 b za popis, 3 b za program

Maru potrebovala súrne vymyslieť hru na sústredenie KSP. Vtom jej do oka padla krabica plná klbiek vlny rôznych farieb. A vtedy dostala nápad. Vlna sa predsa dá hádzať a rozmotávať. Určite bude účastníkov baviť urobiť čo najväčšiu farebnú motanicu. Ale aby to nebola len taká obyčajná ohadzovačka, mala by mať aj nejaké pravidlá.

Maru vymyslela takéto: Hráči si sadnú do kruhu. Zoberie sa klbko farebnej vlny, ktoré si hráči hádžu. Vždy, keď hráč dostane klbko, obmotá si kúsok vlny okolo ruky a snaží sa ostatných rozosmiať. Klbko potom hodí tomu, kto sa vydrží čo najdlhšie nesmiať. Ak sa klbko minie, zoberie sa nové klbko inej farby a môže sa pokračovať v hre. Keď sa minú všetky klbká, hra sa končí. Vyhrá ten hráč, ktorý má najfarebnejšiu ruku, čiže chytil najviac rôznych klbiek.

Pomôžte Maru a napíšte program, ktorý zistí, kto je víťazom.

#### Úloha

Je daný počet hráčov a počet klbiek. Žiadne dve klbká nemajú rovnakú farbu. O každom klbku vlny vieme, v akom poradí si ho hráči hádzali. Zistite, ktorý hráč má najfarebnejšiu ruku.

#### Formát vstupu

V prvom riadku sú dve celé čísla  $n$  a  $m$  – počet hráčov a počet klbiek vlny. Nasleduje  $m$  riadkov. V  $i$ -tom riadku je postupnosť čísel hráčov (hráči sú očíslovaní od 1 po  $n$ ), ktorí mali v rukách  $i$ -te klbko. Postupnosť je ukončená nulou.

Môžete predpokladať, že  $1 \leq n \leq 1\,000\,000$  a že počas celej hry boli klbká dokopy chytené najviac  $1\,000\,000$ -krát.

#### Formát výstupu

Vypíšte číslo hráča, ktorý má najfarebnejšiu ruku. Ak je takých hráčov viac, vypíšte toho s najmenším číslom spomedzi nich.

#### Príklady

vstup

```
5 4
1 4 3 3 5 4 1 3 4 0
1 2 5 2 3 1 4 2 5 3 4 0
1 5 2 2 0
5 3 4 3 0
```

výstup

```
5
```

*Piaty hráč mal ako jediný v ruke klbká všetkých štyroch farieb.*

## 4. Zamotané mosty

kat. Z a O; 10 b za popis, 5 b za program

Jazerbajdžan je netypická krajina. Bežná krajina sa skladá prevažne zo súše a sem-tam nejakej vodnej plochy. V Jazerbajdžane je to presne naopak: Celá krajina je jedno veľké jazero s kopou ostrovov.

Vofakedy dávno napadlo jednému nemenovanému architektovi, že ostrovy pospája mostami. Nanešťastie sa predtým, ako začal navrhovať, najedol hubovej praženice.<sup>3</sup> No a pod jej vplyvom si vymyslel dve zásady:

- každým mostom sa bude dať prejsť len v jednom smere,
- z každého ostrova povedie smerom von práve jeden most.

(Iste ste si všimli, že sa náš architekt inšpiroval vláčikmi z lunaparku, ktorý zamlada navštevoval.)

A tak sa stalo, že ak sa niekto vyberie z ostrova preč, nemusí sa vedieť naň vrátiť späť.

### Úloha

Monika sa rozhodla, že sa na niektorý z ostrovčekov presťahuje, pretože sa v Jazerbajdžane neplatia dane.<sup>4</sup> Chce si však vybrať taký ostrov, na ktorý by sa po odchode z neho vrátil vedela. Koľko možných ostrovov má na výber?

### Formát vstupu

Prvý riadok obsahuje číslo  $n$  ( $2 \leq n \leq 1\,000\,000$ ) – počet ostrovov.

Druhý riadok obsahuje  $n$  čísel  $a_1, a_2, \dots, a_n$  oddelených medzerami ( $1 \leq a_i \leq n$ ,  $a_i \neq i$ ). Z ostrova  $i$  vedie most na ostrov  $a_i$ .

### Formát výstupu

Vypíšte jedno číslo – počet ostrovov vhodných pre Moniku.

### Príklad

vstup

```
4
2 3 2 3
```

výstup

```
2
```

Monike vyhovujú iba ostrovy 2 a 3.

## 5. Odpornosti

kat. Z a O; 4+6+7 b za popis, 8 b za program

Mal som napísanú peknú rozprávku, ale zožral mi ju Luxusko.

V tejto úlohe sa budeme zaoberať problémom pridelovania práce. Máme  $n$  úloh (očíslovaných od 1 po  $n$ ), pričom vykonanie úlohy  $i$  trvá presne  $a_i$  sekúnd. Tiež máme  $k$  úradníkov (očíslovaných od 1 po  $k$ ), ktorým tieto úlohy môžeme prideliť. Naším cieľom je tieto úlohy rozdeliť medzi úradníkov tak, aby čo najskôr boli všetky hotové. Inými slovami, chceme, aby úradník, ktorý bude mať najviac práce, pracoval čo najkratšie. Tento čas označíme  $t$ .

Tento problém je sám o sebe vcelku ťažký a nepoznáme rýchly algoritmus, ktorý ho vie optimálne riešiť. Napriek tomu sa dajú vymyslieť algoritmy, ktoré sú rýchle a dajú takmer optimálne riešenie. Jeden z nich sa dá popísať nasledovne:

*Usporiadaj úlohy od najdlhšej po najkratšiu. Teraz v tomto poradí postupne každej úlohe priradíme úradníka, a to nasledovne: Úlohu vždy priradíme tomu úradníkovi, ktorý má zatiaľ najmenej práce. Ak je takých úradníkov viac, dostane úlohu ten z nich s najmenším číslom.*

<sup>3</sup>To samotné by nebolo až tak nešťastné, keby tie huby neboli značne halucinogénne.

<sup>4</sup>Vybieračov daní znechutil fakt, že sa možno nevedia dostať na každý z ostrovov, tak to radšej vzdali.

### Podúloha a) – 8 bodov

Čo najefektívnejšie implementujte práve popísaný algoritmus. Odovzdajte len program, popis netreba.

#### Formát vstupu

V prvom riadku vstupu sú dve čísla  $n$  a  $k$  ( $1 \leq n, k \leq 200\,000$ ). V druhom riadku je  $n$  čísel  $a_1, \dots, a_n$ : dĺžky trvania jednotlivých úloh. Tieto dĺžky sú kladné celé čísla menšie ako 1 000 000.

#### Formát výstupu

Vypíšte jedno číslo: dĺžku trvania práce najdlhšie pracujúceho úradníka. Dajte si pozor na to, že toto číslo sa nemusí zmestiť do 32-bitovej premennej.

#### Príklad

vstup	výstup
7 3 1 4 10 7 14 2 2	14

Po usporiadaní máme dĺžky trvania úloh 14, 10, 7, 4, 2, 2, 1. Úlohu s časom 14 dáme prvému úradníkovi, úlohu s časom 10 druhému, úloha s časom 7 pôjde tretiemu. Úloha s časom 4 pôjde tretiemu, lebo ten má toho zatiaľ najmenej. Prvá úloha s časom 2 pôjde druhému a ďalšia zase tretiemu. No a posledná úloha pôjde opäť druhému úradníkovi.

Prvý úradník skončí po 14 sekundách, druhý po  $10+2+1=13$  sekundách, tretí po  $7+4+2=13$  sekundách. Celkovo teda skončíme po 14 sekundách. Lepšie to zjavne nejde, preto je odpoveďou číslo 14.

### Podúloha b) – 4 body

Spomínaný algoritmus nemusí vždy dávať optimálne riešenie. Nájdite ľubovoľný jeden vstup, na ktorom ho nedáva. (Uveďte aj to, aký je pre váš vstup optimálny výstup a aký výstup dá náš algoritmus.)

### Podúloha c) – 6 bodov

Existujú však situácie, kedy vieme dokázať, že spomínaný algoritmus vždy dá optimálne riešenie. Dokážte: Nech  $t^*$  je výsledný čas pri optimálnom riešení. Ak je čas spracovania každej úlohy ostro väčší ako  $t^*/3$ , tak náš algoritmus určite nájde optimálne riešenie.

Pri dokazovaní sa snažte urobiť čo najpochvejši dôkaz. Mávanie rukami nedostane plný počet bodov. Dobrý dôkaz má napríklad nasledovnú kostru: Nech  $R_o$  je nejaké (ľubovoľné) optimálne riešenie a  $R$  je riešenie, ktoré dá náš algoritmus. Potom viem bez zmeny ceny prerobiť  $R_o$  na  $R$  (alebo naopak).

### Podúloha d) – 7 bodov

A tiež sa dá dokázať, že nech už vstup vyzerá akokoľvek, náš algoritmus nikdy nie je príliš zlý. Dokážte: Nech  $t^*$  je výsledný čas pri optimálnom riešení. Potom náš algoritmus určite nájde riešenie, ktorého výsledný čas je maximálne  $4t^*/3$ .

## 6. Obdarený nespratník

kat. O; 12 b za popis, 8 b za program

Keď mal Georg štyri roky, dostal na Vianoce hračkársku sadu murárskeho náčinia. Jeho rodičia zrejme dúfali, že ho týmto navedú na cestu k perspektívnej kariére architekta, no Georg mal už vtedy vo svojej budúcnosti jasno. Stane sa z neho demolátor výškových budov.

Kelnu okamžite zahodil za skriňu, špachtľu si ani nevšimol a olovnicou priviazal svojho kocúra Mura k nohe stoličky, aby mu nezavadzal. Potom schytil kladivko, jediný užitočný kus celej sady, a začal rozvíjať svoje deštruktívne vloh y a rozbíjať mamine hrnce. Keď sa minuli, presunul svoju pozornosť na vázy, dedovo koleno a babkine okuliare, prasiatko s úsporami, nábytok, a bol by možno porozbíjal aj okná, keby ho nezaujali zložené čísla.

### Úloha

Keď kladivkom udrieme po zloženom čísle  $n$ , rozbije sa na dve prirodzené čísla  $x$  a  $y$ , pričom obe budú väčšie ako 1 a bude platiť  $x \cdot y = n$ . Napríklad z čísla 12 vznikne buď dvojica 2 a 6, alebo 3 a 4. To, ktorú z možných dvojíc však naozaj získame, nevieme ovplyvniť.

Samozrejme, prvočísla sú nerozbitné.

Georg má číslo  $n$  a rád by z neho získal číslo  $k$ . Môže si byť istý, že sa mu to podarí?

### Formát vstupu

V prvom riadku vstupu sú zadané dve čísla  $n$  a  $k$  oddelené medzerou ( $2 \leq n, k \leq 10^{13}$ ).

### Formát výstupu

Ak je vždy možné pomocou kladivka vyrobiť z  $n$  číslo  $k$ , vypíšte **ano**, inak vypíšte **nie**.

### Príklady

vstup

5 20

výstup

nie

*Keďže 5 je prvočíslo, kladivko nám vôbec nepomôže. Navyše je celkom márne pokúšať sa rozbíjaním číslo zväčšiť.*

vstup

20 5

výstup

ano

*Číslo 20 sa môže rozbiť na 4 a 5, alebo na 2 a 10. V druhom prípade ešte rozbijeme 10 a získame 5.*

vstup

12 4

výstup

nie

*Mohlo by sa stať, že sa číslo 12 rozbije na 2 a 6. Zo 6 sa dá potom vyrobiť už len 2 a 3, čo nám nepomôže.*

vstup

12 12

výstup

ano

## 7. Opička, banány a ADHD

kat. O; 13 b za popis, 7 b za program

Nedávno postavili v džungli Továreň na spracovanie banánov a v rámci znižovania nezamestnanosti najali všetky zvieratká z okolia, vrátane zelenej opičky Lin.

Navedeli však, že trpí poruchou pozornosti, známou ako ADHD.<sup>5</sup> Preto sa jej podarilo hneď prvý deň v práci rozbiť veľký odšťavovač ovocia. Aby sa zamedzilo ďalším škodám, Lin okamžite preradili do Triediarne, kde musí robiť plno byrokratickej práce.

## Úloha

V Triediarni sa nachádza dlhý pás, na ktorom sú zaradom poukladané banány. Každý banán má priradenú nejakú kladnú hodnotu, predstavujúcu stupeň nahnitia banánu. Každý postupnosti banánov sa dá priradiť nezáporná hodnota, ktorú zistíme ako rozdiel maximálnej a minimálnej nahnitosti banánov v danej postupnosti. Túto hodnotu voláme *rozptyl nahnitosti*.

Teda napr. postupnosť banánov s nahnitosťou 4, 2, 1, 4 a 7 má rozptyl nahnitosti rovný  $7 - 1 = 6$ , lebo maximálna nahnitosť je 7 a minimálna 1.

Lin dostala nasledovnú úlohu: Na páse je postupnosť banánov. Lin má pre každý súvislý podúsek tejto postupnosti zistiť a zapísať jeho rozptyl nahnitosti. Na záver musí všetky zapísané hodnoty sčítať. Keďže je však Lin strašne lenivá, poprosila vás, či by ste jej nepomohli.

## Formát vstupu

V prvom riadku vstupu je číslo  $n$  ( $1 \leq n \leq 1\,000\,000$ ), ktoré označuje počet prvkov postupnosti. V druhom riadku je daných  $n$  čísel: nahnitosť jednotlivých banánov v poradí, v ktorom ležia na páse. Môžete predpokladať, že všetky čísla v postupnosti sú medzi 0 a 1 000 000 vrátane.

## Formát výstupu

Na výstup vypíšete jeden riadok a v ňom súčet rozptylov nahnitosti všetkých súvislých podpostupností danej postupnosti.

## Príklad

vstup	výstup
4 4 1 7 4	30

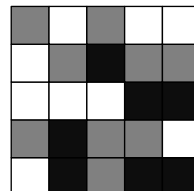
Postupnosť obsahuje nasledovné podúseky: (4), (4,1), (4,1,7), (4,1,7,4), (1), (1,7), (1,7,4), (7), (7,4) a (4). Ich rozptyly nahnitosti sú postupne 0, 3, 6, 6, 0, 6, 6, 0, 3 a 0.

## 8. Oblej to!

kat. O; 2 b za popis, 20 b za program

Ak ste boli v minulej sérii smutní, že sa hra *Flood-it!* (*sic*)<sup>6</sup> dostala len do kategórie Z, v tejto chvíli váš žiaľ končí a máte ju tu pre zmenu vy, riešitelia kategórie O. A bude to pestré!

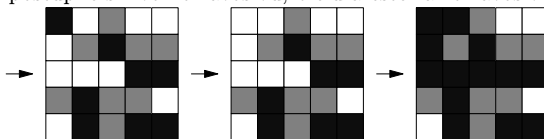
Hrací plán v hre *Flood-it!* má tvar veľkého štvorca rozdeleného na  $n \times n$  malých štvorcov. Každý z týchto štvorcov na začiatku dostane jednu náhodnú farbu spomedzi  $f$  možných. Samozrejme, z času na čas sa takto stane, že dva susedné štvorce dostanú tú istú farbu. A občas takto vzniknú aj tri či štyri štvorce pokope. Tým na hracom pláne vznikajú veľké jednofarebné oblasti. Na obrázku vpravo je príklad pre  $n = 5$  a  $f = 3$ . Vidíme 15 jednofarebných oblastí, najväčšiu z nich tvoria 4 štvorce.



<sup>5</sup> Attention Deficit Hyperactivity Disorder

<sup>6</sup>To „sic“ je skrátená podoba latinského „sic erat scriptum“, teda „tak bolo napísané“. V typografii sa to používa v situáciách, kedy citujete niečo, čo obsahuje chybu. Pridaním „sic“ teda hovoríte: „Viem, že je tam chyba, spravili ju oni, nie ja!“. A zrazu vás nik nemôže mať za blbca :) V tomto prípade je chybou samozrejme ten spojovník medzi slovami, ktorý tam nemá čo robiť. P.S.: Use „sic“ like a boss: vysádzajte ho kurzívou!

Samotná hra prebieha v ťahoch. Ťah spočíva v tom, že si hráč zvolí jednu farbu. Na tú farbu sa následne prefarbí jednofarebná oblasť obsahujúca ľavý horný roh (ako keby sme v grafickom editore použili nástroj „vedierko“). Tým sa samozrejme môže stať, že sa táto jednofarebná oblasť rozšíri. Na nasledujúcom obrázku vidíme, čo sa stane, ak hráč začne s hracím plánom z prvého obrázka a postupne si zvolí tmavosivú, bielu a ešte raz tmavosivú farbu.



Cieľom hry je pomocou dostatočne malého počtu ťahov dosiahnuť, aby bola úplne celá plocha tej istej farby (je jedno, ktorej). Napríklad v situácii z posledného obrázka vieme hru dohrať pomocou troch ďalších ťahov, v poradí svetlosivá, biela, tmavosivá.

## Úloha

V ťažkej obtiažnosti hry *Flood-it!* je  $n = 22$  a  $f = 6$ , cieľom je vyhrať hru pomocou nanajvýš 36 ťahov. Napíšte program, ktorý bude hrať a čo najčastejšie vyhrávať túto verziu hry. Farby políček v každej hre budú naozaj volené dostatočne náhodne – teda nečakajú vás žiadne zákerné ručne robené levely.

## Formát vstupu

Vstupný súbor začína číslom  $h$ , udávajúcim počet hier. Následne je tam pre každú hru popis začiatočného hracieho plánu: 22 riadkov a v každom z nich reťazec 22 cifier z rozsahu 1 až 6.

## Formát výstupu

Pre každú hru vypíšte jeden riadok a v ňom *presne* 36 cifier z rozsahu 1 až 6: čísla farieb, v poradí, v akom si ich chcete voľiť. (Ak nájdete riešenie na menej ako 36 ťahov, doplňte mu na koniec ľubovoľné povolené cifry, nič tým nepokazíte.) Nevypisujte žiadne medzery.

## Hodnotenie

Popis píše veľmi stručný: len pár vetami popíšte hlavnú myšlienku toho, ako vaše riešenie funguje. Každý rozumný popis dostane tie dva body, bez ohľadu na kvalitu algoritmu.

Počas odovzdávania budú vaše riešenia provízorne testované na 20 vstupoch a časový limit bude 20 sekúnd. Po skončení série vaše *posledné* odovzdané riešenie zoberieme a pretestujeme. Pri tomto testovaní použijeme 1000 vstupov a časový limit bude 1000 sekúnd. Odporúčame vám aj doma robiť podobné testy počas riešenia úlohy. Ak váš program prekročí pri tomto záverečnom testovaní časový limit, vyhodnotíme tie hry, ktoré stihol odhrať a vypísať na výstup.

Body za program budú vypočítané nasledovne: ak z  $h$  hier vyhrá  $v$ , dostane  $\lfloor 20.07 \cdot v/h \rfloor$  bodov. Inými slovami,  $20.07 \cdot v/h$  bodov, zaokrúhlené *nadol* na celé číslo. (Plný počet bodov teda dostane program, ktorý pri záverečnom testovaní z  $h = 1000$  hier vyhrá aspoň 997.)

## Príklad

Ukážkový vstup je celkom veľký a zaberal by tu veľa miesta. Preto si ho môžete stiahnuť na adrese: <http://www.ksp.sk/wiki/uploads/Zadania/29-4-8.in>. Ukážkový výstup nájdete na adrese <http://www.ksp.sk/wiki/uploads/Zadania/29-4-8.out>.



## 9. Thajsko a chobotník

kat. T; 15 b za popis, 10 b za program

Malý chobotník<sup>7</sup> Zemčo sa práve vracia domov do svojho útesu neďaleko thajského mesta Pattaya. Vracia sa zo stretnutia chobotníkov a chobotníc, ktoré sa každoročne koná niekde strážne ďaleko, na druhom konci Thajského zálivu. Na stretnutí poriadne vyhladel, preto by sa chcel cestou domov čo najlepšie najesť.

### Úloha

Keď si Thajský záliv predstavíme ako štvorčekovú sieť, Zemčov domov je v bode  $[0, 0]$  (ľavý dolný roh) a strážne ďaleké miesto, odkiaľ Zemčo cestuje, je v bode  $[a, b]$  (pravý horný roh).

Aby sa chobotník cestou domov nestratil, rozhodol sa, že sa bude pohybovať len rovnobežne so súradnicovými osami a pôjde jednou z najkratších ciest – takže sa bude pohybovať iba dole alebo doľava. Na niektorých miestach sa nachádzajú húfy rýb a Zemčo môže v každom húfe, cez ktorý prejde, uloviť nejaké ryby. Samozrejme, chce ich dokopy uloviť čo najviac.

### Formát vstupu

Na vstupe dostanete tri kladné celé čísla  $a$ ,  $b$  a  $n$ , ktoré určuje štartovnú pozíciu Zemča a počet húfov rýb v Thajskom zálive ( $1 \leq a, b \leq 10^9$ ,  $1 \leq n \leq 200\,000$ ). Ďalej nasleduje  $n$  riadkov, z ktorých  $i$ -ty obsahuje tri celé čísla  $x_i$ ,  $y_i$  a  $f_i$ : pozíciu  $i$ -teho húfu a počet rýb, ktoré môže Zemčo v  $i$ -tom húfe uloviť. Platí  $0 \leq x_i \leq a$ ,  $0 \leq y_i \leq b$  a  $0 \leq f_i \leq 1\,000$ . Žiadne dva húfy nie sú na rovnakej pozícii.

### Formát výstupu

Vášou úlohou je vypísať jedno číslo: koľko najviac rýb dokáže Zemčo uloviť, pokiaľ pôjde jednou z najkratších ciest (v súradnicovej sústave sa bude pohybovať iba dole alebo doľava).

### Hodnotenie

Za úplne správne riešenie je samozrejme plný počet bodov. Aspoň 7+4 bodov však môžete dostať za správne riešenie, ktoré predpokladá, že  $a, b \leq 1\,000$ . Inak predpokladajte, že tieto čísla môžu byť naozaj obrovské.

### Príklad

vstup

```
1 2 3
0 0 1
0 2 2
1 0 3
```

výstup

```
4
```

```
2#
```

```
~~
```

```
13
```

Zemčo cestuje z # do húfu 1 cez huf 3.

---

<sup>7</sup> samec chobotnice, nie zvierá s chobotom – slon

vstup

6	6	9
0	3	2
1	2	6
2	1	2
2	0	5
2	5	9
4	2	5
3	3	3
2	4	4
5	4	8

výstup

20
~~~~~#
~9~~~~
~4~~8~
2~~3~~~
~6~~5~~
~~2~~~~
@5~~~~

Zemčo cestuje z # do @ a najviac sa mu opláti  
vzbierať húfy s počtami 8, 5, 2, 5 alebo 9, 4,  
2, 5

## 10. Tunely v hyperpriestore

kat. T; 10 b za popis, 15 b za program

Konfederácia Spriatelnených Planét, najväčšia aliancia v blízkom vesmíre, má problém. Na vzdialenej planéte Aspir Adora práve začali vyrábať najlacnejšie vysávače v celej galaxii a všetci ľudia, mimozemšťania aj energetické bytosti chcú jeden mať. Ale transportný systém, ktorý sa používal doteraz, nevláda taký veľký dopyt.

Transportný systém Konfederácie sa skladá z množstva hyperpriestorových tunelov, cez ktoré sa dá „skákať“ z planéty na planétu. Je to omnoho rýchlejšie, ako letieť normálnou podsvetelnou rýchlosťou, ale tunelmi sa nemusí dať dostať všade – sú jednosmerné a z každej planéty vedie iba jeden.<sup>8</sup>

Obyvatelov galaxie už nebaví čakať na svoje vysávače a začali hroziť vzburou, ak nebude každá objednávka vybavená dostatočne rýchlo. Existujúce tunely to nezvládajú, takže vašou úlohou je postaviť niekoľko nových tunelov tak, aby sa z Aspir Adory dalo všade dostať na najviac  $k$  skokov. (Nové tunely budú tiež jednosmerné, ale môžete ich postaviť medzi ľubovoľnými planétami.) Postaviť nový tunel je dosť drahé, takže ich spravte najmenej, ako sa len dá.

### Formát vstupu

Na prvom riadku je počet planét  $n$  a maximum skokov  $k$  ( $2 \leq n \leq 500\,000$ ,  $1 \leq k \leq 20\,000$ ).

Druhý riadok obsahuje  $n$  čísel  $a_1, a_2, \dots, a_n$  oddelených medzerami ( $1 \leq a_i \leq n$ ,  $a_i \neq i$ ). Z planéty  $i$  vedie tunel na planétu  $a_i$ . Aspir Adora má číslo 1.

### Formát výstupu

Vypíšte jeden riadok s jediným číslom – minimálnym počtom tunelov, čo treba postaviť, aby sa z Aspir Adory dalo na každú planétu dostať na  $k$  alebo menej skokov.

### Príklad

vstup

8	3						
2	3	5	5	6	7	8	5

výstup

2
---

<sup>8</sup>Povráva sa, že inžinier, čo tunely navrhoval, sa inšpiroval svojou domovinou, Jazerbajdžanom.

vstup

14	4
2	3
4	5
6	3
5	10
8	9
14	11
12	13

výstup

3
---

*Například môžeme postaviť tunely  $2 \rightarrow 7$ ,  
 $1 \rightarrow 14$  a  $14 \rightarrow 10$ .*