

## Bányász RPG

Gyuri kedvenc számítógépes játéka egy bányász szimulátor. Ebben a játékban különböző alapanyagokat lehet bányászni, és minél többet bányásztunk már korábban, annál tapasztaltabb lesz a karakterünk, így gyorsabbá válik az alapanyagok kitermelése (az elején a karakter teljesen tapasztalatlan, tehát 0 tapasztalati pontja van).

A játékban több különböző féle alapanyag található. Tetszőleges alapanyagból egyszerre egy egységnyi mennyiség termelhető ki, ez kezdetben 2 percbe telik, és az alapanyag mellett a karakterünk 1 tapasztalati ponttal is gazdagodik.

Minden alapanyagnak van egy tapasztalat-korlátja, amit ha elér a karakter (vagyis az eddig szerzett összes tapasztalati pontja legalább ennyi), akkor az adott típusú alapanyag kitermelése már csak 1 percbe telik.

Gyurinak ahhoz, hogy tovább tudjon lépni a játékban, ki kell termelnie minden alapanyagból egy-egy megadott mennyiséget.

Írj programot, ami meghatározza, hogy mennyi az a legkevesebb idő, ami alatt Gyuri teljesíteni tudja a feladatot!

### Bemenet

A standard bemenet első sorában az alapanyag fajták száma van ( $1 \leq N \leq 100\,000$ ). A második sorban az egyes alapanyagok gyorsabb kitermeléséhez szükséges  $l_i$  tapasztalat-korlátok találhatóak. A harmadik sorban az egyes alapanyagokból kitermelendő  $m_i$  mennyiségek találhatóak ( $1 \leq l_i, m_i \leq 100\,000$ ).

### Kimenet

A standard kimenetre a minimális időt kell írni (percben), ami alatt Gyuri teljesítheti a feladatot!

### Példa

Bemenet	Kimenet
3	10
4 3 5	
5 1 1	

### Korlátok

Időlimit: 0.3 mp.

Memórialimit: 32 MB

### Pontozás

A pontszám fele szerzhető olyan tesztekre, ahol  $N \leq 1\,000$ .

A pontszám 30%-a szerzhető olyan tesztekre, ahol az  $l_i$  értékek megegyeznek.

## Következő permutáció

A permutációk a mai napig intenzív kutatások tárgya. Az  $1, \dots, N$  természetes számok permutációi az olyan  $N$ -hosszú  $p_1, p_2, \dots, p_N$  sorozatok, melyek az  $1, \dots, N$  számok mindegyikét pontosan egyszer tartalmazzák.

Most olyan permutációkat vizsgálunk, amelyekben nem fordul elő bizonyos minta. Azt mondjuk, hogy a  $p_1, p_2, \dots, p_N$  permutáció 3-1-2 minta-mentes, ha nincs olyan három index  $1 \leq i < j < k \leq N$ , amelyekre egyszerre teljesülnek a  $p_i > p_j$ ,  $p_i > p_k$  és  $p_j < p_k$  egyenlőtlenségek.

Készíts programot, amely meghatározza egy 3-1-2 minta-mentes permutáció rákövetkezőjét a lexikografikus sorrendben! Egy  $a_1, a_2, \dots, a_N$  sorozat pontosan akkor előz meg egy  $b_1, b_2, \dots, b_N$  sorozatot a lexikografikus sorrendben, ha van olyan  $1 \leq i \leq N$  index, hogy a sorozatok első  $i-1$  darab (lehet 0 is) eleme azonos és  $a_i < b_i$ .

### Bemenet

A standard bemenet első sora az  $N$  számot ( $3 \leq N \leq 10\,000$ ) tartalmazza. A második sor az  $1, \dots, N$  számok egy 3-1-2 mentes permutációját tartalmazza. A bemenet nem az  $N, N-1, \dots, 2, 1$  csökkenő sorozat.

### Kimenet

A standard kimenet első sorába azt a 3-1-2 minta-mentes permutációt kell írni, amely a bemenet rákövetkezője a lexikografikus sorrendben!

### Példa

Bemenet  
5  
2 3 5 4 1

Kimenet  
2 4 3 1 5

### Korlátok

Időlimit: 0.25 mp.

Memórialimit: 32 MB

### Pontozás

A pontok 22%-a szerezhető olyan tesztekre, ahol  $N \leq 10$ .

## Legmesszebbi rossz sorrendű

Egy rendezési feladat arról szól, hogy az elemeket állítsuk növekvő sorrendbe. Rendezés előtt előfordulhat, hogy egyes elemek nagyobbak valamely náluk későbbi elemnél.

Írj programot, amely megadja egy sorozat két, egymástól legmesszebb levő elemét, amelyek sorrendje nem jó (azaz a korábbi nagyobb a későbbinél)!

### Bemenet

A standard bemenet első sorában az elemek száma van ( $2 \leq N \leq 100\,000$ ). A második sor az elemeket tartalmazza ( $-100\,000 \leq E_i \leq 100\,000$ ).

### Kimenet

A standard kimenet első sorába két, egymástól legmesszebb levő elem indexét kell írni, az első a korábbi, a második a későbbi legyen! Több megoldás esetén azt kell kiírni, amelyik első eleme a bemeneti sorozatban legkorábban volt. Ha nincs megoldás, akkor egyetlen  $-1$ -et kell kiírni!

### Példa

Bemenet	Kimenet
10	1 6
7 2 4 17 6 5 10 10 18 19	

### Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

## A lehető legkevesebb metróval utazás

A világ legnagyobb városában, Megalopoliszban sok metróvonal van. A metróvonalakat és a metróállomásokat sorszámokkal azonosítjuk. A metrószerelvények rendszeresen közlekednek, mindkét menetirányban.

Írj programot, amely megadja, hogy minimum hány metróval kell utazni, hogy eljuthassunk egy adott állomásról egy másikra!

### Bemenet

A standard bemenet első sorában a metróvonalak száma ( $1 \leq N \leq 200$ ), az állomások száma ( $1 \leq M \leq 10\,000$ ), az induló állomás sorszáma ( $1 \leq \text{Ind} \leq M$ ) és a célállomás sorszáma ( $1 \leq \text{Erk} \leq M$ ,  $\text{Erk} \neq \text{Ind}$ ) van. A következő  $N$  sorban egy-egy metróvonal leírása található. Minden sor első száma a vonal metrómegállóinak a száma ( $2 \leq A_i \leq 5000$ ), amelyet a megálló sorszámjai követnek ( $1 \leq \text{Meg}_{i,j} \leq M$ ).

### Kimenet

A standard kimenet első sorába azt a minimális  $K$  számot kell írni, ahány metróvonalat használva eljuthatunk az első állomásról az utolsó! A második sor  $K$  olyan metróvonal sorszámát tartalmazza, amelyekkel  $K-1$  átszállással el lehet jutni a célállomásra! A metróvonalakat az utazás sorrendjében kell kiírni!

Több megoldás esetén bármelyik kiírható. Ha nincs megoldás, akkor egyetlen sorba -1-et kell kiírni!

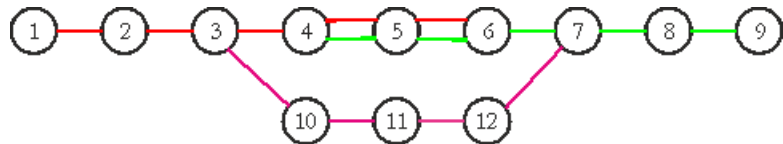
### Példa

Bemenet

```
3 12 1 8
6 1 2 3 4 5 6
6 4 5 6 7 8 9
5 3 10 11 12 7
```

Kimenet

```
2
1 2
```



### Korlátok

Időlimit: 0.5 mp.

Memórialimit: 32 MB

## MI bróker

Manapság egyre inkább jellemző, hogy a tőzsdei kereskedők a mesterséges intelligencia (MI) segítségét is igénybe veszik részvények megvételéhez és eladásához. Pali úgy döntött, hogy felveszi a kesztyűt és készít egy saját kereskedő MI-t.

Pali programja egy adott részvényből képes minden másodpercben vagy venni egy darabot, vagy eladni egy darabot. A program csak akkor adhat el részvényt, ha birtokol egyet, és egyszerre legfeljebb egy részvényt birtokolhat. A részvény értéke minden másodpercben változhat.

A kereskedéshez meg kell adni az MI-nek egy  $(V, E)$  párt, ahol  $V < E$  és  $V$  az a legnagyobb érték, amennyiért a program még hajlandó részvényt vásárolni,  $E$  pedig az a minimális érték, amennyiért a program részvényt ad el. Formálisan megfogalmazva, az MI minden másodpercben a következő akciót hajtja végre: ha ebben a másodpercben a részvény értéke  $a$ , akkor

1. ha nincs részvény a birtokában és  $a \leq V$  teljesül, akkor  $a$  áron venni fog egyet;
2. ha birtokol egy részvényt és  $a \geq E$ , akkor  $a$  áron eladja azt;
3. minden egyéb esetben nem csinál semmit.

Pali megszerezte a részvény értékeinek sorozatát az elmúlt  $N$  másodpercben. Ez másodpercenként egy  $\$1$  és  $\$500$  közötti egész  $a_i$  érték. Pali több lehetséges  $(V, E)$  pár mindegyikéhez megszeretné tudni, hogy annak választása esetén mekkora lett volna a haszna (az eladásokból szerzett összeg és a vételékért fizetett összeg különbsége) az  $N$  másodperc során, feltéve, hogy kezdetben nem birtokolt részvényt, viszont van elég pénze részvényt venni bármelyik másodpercben.

Írj programot, ami meghatározza a Pali által megadott összes párra a haszon értékét!

### Bemenet

A standard bemenet első sorában a másodpercek ( $2 \leq N \leq 10\,000$ ) és Pali kérdéseinek száma ( $1 \leq Q \leq 150\,000$ ) található.

A második sorban a részvény másodpercenkénti árfolyamai ( $1 \leq a_i \leq 500$ ) vannak.

A következő  $Q$  sor a vételi és eladási korlátokat ( $1 \leq V_j < E_j \leq 500$ ) tartalmazza az egyes esetekben.

### Kimenet

A standard kimenetre összesen  $Q$  sorba soronként egy egész számot kell írni, a haszon értékét az egyes esetekben!

### Példa

Bemenet	Kimenet
5 3	90
30 120 40 125 100	175
35 120	-5
40 120	
100 125	

Magyarázat: az első esetben 30-ért részvényt vesz a program az első másodpercben, melyet a másodikban 120-ért elad és többé nem vesz részvényt, így  $-30+120=90$  a haszon.

A harmadik esetben 30-ért részvényt vesz az első másodpercben, melyet a negyedik másodpercben 125-ért elad, majd 100-ért részvényt vesz az utolsó másodpercben, így  $-30+125-100=-5$  a haszon.

**Korlátok**

Időlimit: 1.0 mp.

Memórialimit: 64 MB

**Pontozás**

A pontszám 16%-a szerezhető olyan tesztekre, ahol  $Q \leq 50$ .

A pontszám további 24%-a szerezhető olyan tesztekre, ahol a bemenetben pontosan kétféle  $a_i$  érték szerepel.

## Szigetek

Szigetek vannak egy vonalban, amelyek kikötőiben kikötői díjat szednek. Minden szigeten egy kikötő van. Bendegúz szeretne a hajójával eljutni az első szigetről az utolsóra. Legfeljebb 2 sziget távolságra tud eljutni egy tankolással (azaz például az első szigetről közvetlenül vagy a másodikra, vagy a harmadikra mehet). Ahol kiköt, ott fizetnie kell (az első szigeten is, ahonnan indul a hajó).

Írj programot, amely megadja, hogy minimum mekkora összegbe kerül eljutni az első szigetről az utolsóra!

### Bemenet

A standard bemenet első sorában a szigetek száma van ( $2 \leq N \leq 100\,000$ ). A második sor az egyes szigetek kikötőiben fizetendő díjakat tartalmazza ( $1 \leq D_i \leq 1000$ ).

### Kimenet

A standard kimenet első sorába a minimális fizetendő összeget kell írni!

### Példa

Bemenet

6

3 5 9 7 1 6

Kimenet

19

### Korlátok

Időlimit: 0.2 mp.

Memórialimit: 32 MB

## Színes szobák

Van sorban  $N$  szoba, mindegyik  $K$  szín valamelyikével kifestve. Több szobának is lehet azonos a színe. Minden szobában van egy ember.

$M$  darab utasítást mondanak be hangosbemondón, minden utasítás egy  $(c, x)$  számpár, ahol  $c$  egy szín és  $x$  értéke  $+1$  vagy  $-1$ . Mindenkinek, aki  $c$  színű szobában van, át kell mennie a mostani  $+x$  sorszámú szobába (azaz az egyel kisebb vagy nagyobb sorszámúba). Akik kilépnek a szobákból ( $1$ -esből  $0$ -ra vagy  $N$ -ről  $N+1$ -re mennének), távoznak.

Írj programot, amely meghatározza, hogy az  $M$  utasítás végrehajtása során hány ember távozott!

### Bemenet

A standard bemenet első sora a szobák ( $1 \leq N \leq 100\,000$ ), a színek ( $1 \leq K \leq 100\,000$ ) és az utasítások ( $1 \leq M \leq 100\,000$ ) számát tartalmazza. A második sor  $N$  számot tartalmaz, az egyes szobák színeit ( $1 \leq c_i \leq K$ ). A következő  $M$  sor mindegyike egy utasítást tartalmaz  $c \ x$  számpár formájában, ahol  $c$  egy szín érték ( $1 \leq c \leq K$ ),  $x$  pedig a  $+1$  vagy  $-1$  szám.

### Kimenet

A standard kimenet első sorába azon emberek számát kell írni, ahányan távoztak az  $M$  utasítás végrehajtása során!

### Példa

Bemenet	Kimenet
8 3 4	4
1 1 2 2 2 1 2 1	
1 1	
2 -1	
1 -1	
1 -1	

### Korlátok

Időlimit: 0.3 mp.

Memórialimit: 32 MB

### Pontozás

A pontok 16%-a szerezhető olyan tesztekre, ahol minden utasításban a lépés értéke azonosan  $+1$  vagy  $-1$ .

A pontok 16%-a szerezhető olyan tesztekre, ahol minden  $s, z$  színre legfeljebb 10 olyan szoba van, amelynek a színe  $s, z$ .

A pontok 16%-a szerezhető olyan tesztekre, ahol  $N \leq 10\,000$  és  $M \leq 10\,000$ .