

Cserebere

Adott egy N elemű, egész számokból álló $V = [V_1, V_2, \dots, V_N]$ sorozat. A sorozaton az alábbi átalakítást hajtjuk végre:

- Egy lépésben végigmegyünk a sorozat elemein balról jobbra, és ha az éppen vizsgált elem páratlan, a következő elem (tehát az aktuális elem jobb szomszédja) viszont páros, akkor megcseréljük a két elemet.
- Ezt a lépést addig ismételjük, amíg a sorozatban legalább egy cserét végrehajtottunk.

Például az $[1, 4, 3, 2]$ sorozat esetén egyetlen lépés során a következő történik:

- Az első elem 1 (páratlan), a következő elem 4 (páros), így ezeket megcseréljük.
A sorozat: $[4, 1, 3, 2]$.
- A második elem 1 (páratlan), a következő elem 3 (páratlan), így nem cserélünk.
A sorozat: $[4, 1, 3, 2]$.
- A harmadik elem 3 (páratlan), a negyedik elem 2 (páros), így megcseréljük őket.
A sorozat: $[4, 1, 2, 3]$.

Írj programot, ami megadja a sorozatot az átalakítás végén.

Bemenet

A standard bemenet első sorában a sorozat elemeinek N száma található.

A második sor N nemnegatív egész számot tartalmaz, a sorozat V_i elemeit.

Kimenet

A standard kimenetre egyetlen sort kell írni, ami N egész számot tartalmaz: a sorozatot az átalakítás befejezése után.

Példa

Bemenet	Kimenet
4	4 2 1 3
1 4 3 2	

Az első lépést követően kapott sorozat a korábban leírtak alapján: $[4, 1, 2, 3]$. A második lépés után a $[4, 2, 1, 3]$ sorozat adódik. Mivel történt csere, így újabb lépés következik. A harmadik lépés után a $[4, 2, 1, 3]$ sorozatot kapjuk, tehát a sorozat nem változott, így véget ér az átalakítás.

Bemenet	Kimenet
6	2 4 6 3 5 1
3 2 5 4 1 6	

Korlátok

$$2 \leq N \leq 200\,000$$

$$0 \leq V_i \leq 10^9 \text{ minden } i = 1, \dots, N\text{-re}$$

Időlimit: 1.0 mp.

Memórialimit: 256 MB

Pontozás

A megoldásokat sok különböző tesztesetre lefuttatjuk. A tesztesetek részfeladatokba vannak csoportosítva. Egy-egy részfeladatot akkor tekintünk megoldottnak, ha volt legalább egy olyan beadásod, amely az adott részfeladat minden tesztesetére helyes megoldást adott. A feladat összpontszámát a megoldott részfeladatokra kapott pontszámok összege adja.

Részfeladat	Korlátok	Pontszám
0	a minta	0
1	$N = 2$	5
2	$N \leq 10$	10
3	$N \leq 1\,000$	15
4	nincsenek további megkötések	20

Három szám

Adott egy N elemű, pozitív egész számokból álló $A = [A_1, A_2, \dots, A_N]$ sorozat. Szeretnénk kiválasztani három elemet a sorozatból az eredeti sorrendjükben úgy, hogy nagyság szerint növekedjenek és az összegük a lehető legmagasabb legyen. Formálisan, olyan $1 \leq p < q < r \leq N$ indexeket keresünk, amikre $A_p < A_q < A_r$ teljesül, és az $A_p + A_q + A_r$ összeg maximális.

Írj programot, amely megkeres három megfelelő sorozatelemet.

Bemenet

A standard bemenet első sorában a számsorozat elemeinek N száma van. A második sor a számsorozat A_i elemeit tartalmazza.

Kimenet

A standard kimenet első sorába az elérhető legnagyobb M összeget kell írni.

A második sor három olyan $p < q < r$ indexet tartalmazzon, amikre $A_p < A_q < A_r$ és $M = A_p + A_q + A_r$ teljesül. Több helyes megoldás esetén bármelyik megadható.

Ha nem létezik a feltételeknek megfelelő három index, akkor egyetlen sort kell kiírni egyetlen 0 értékkel.

Példa

Bemenet	Kimenet
7	17
4 3 6 3 7 3 7	1 3 5

Korlátok

$$5 \leq N \leq 50\,000$$

$$1 \leq A_i \leq 1\,000\,000 \text{ minden } i = 1 \dots N\text{-re}$$

Időlimit: 0.5 mp.

Memórialimit: 256 MB

Pontozás

A megoldásodat sok különböző tesztesetre lefuttatjuk. A tesztesetek részfeladatokba vannak csoportosítva. Egy-egy részfeladatot akkor tekintünk megoldottnak, ha volt legalább egy olyan beadásod, amely az adott részfeladat minden tesztesetére helyes megoldást adott. A feladat összpontszámát a megoldott részfeladatokra kapott pontszámok összege adja.

Részfeladat	Korlátok	Pontszám
0	a minta	0
1	$N \leq 500$	10
2	$N \leq 5000$	14
3	nincsenek további megkötések	26

Minden részfeladatban **részpontokat** lehet szerezni: a pontszám 50%-a jár, ha az összes tesztesetben az első sorba kiírt szám (a három szám összege) helyes.

Kör

Egy kezdetben N elemű síkbeli ponthalmazhoz egyesével hozzáadunk Q további pontot. Minden beszúrás után jelölje r a legkisebb körnek a sugarát, amely az összes eddigi pontot tartalmazza.

Írj programot, ami kiszámít egy olyan R pozitív **egész** értéket minden beszúrás után, amire teljesül az $r \leq R \leq 2 \cdot r$ egyenlőtlenség.

Bemenet

A standard bemenet első sorában a kezdeti pontok N , és a hozzáadott pontok Q száma található.

A következő N sorban a kezdeti pontok (x_i, y_i) koordinátái találhatók.

A következő Q sorban az egyes hozzáadott pontok (x_i, y_i) koordinátái találhatók.

Kimenet

A standard kimenetre Q sort kell írni, minden beszúrt ponthoz egyet: egy olyan R egész értéket, melyre $r \leq R \leq 2 \cdot r$ teljesül.

Több lehetséges helyes megoldás esetén bármelyik megadható.

Példa

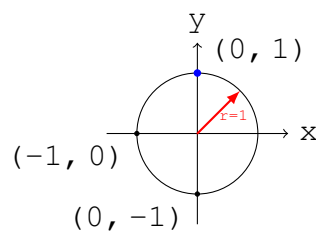
Bemenet

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

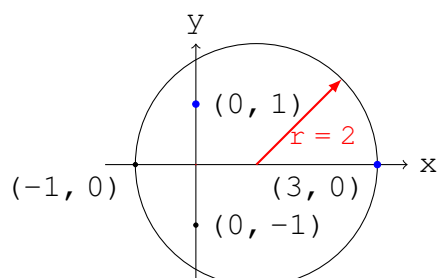
Kimenet

```
1
4
4
```

Az első pont beszúrása után a legkisebb kör, ami tartalmazza az összes eddigi pontot a $(0, 0)$ középpontú, $r = 1$ sugarú kör. Így $R = 1$ egy lehetséges helyes válasz.



A második pont beszúrása után a legkisebb kör középpontja $(1, 0)$ és sugara $r = 2$, így például $R = 4$ megfelelő.



A harmadik pont beszúrásával a ponthalmaz nem változik, így $R = 4$ továbbra is megfelelő.

Korlátok

$$1 \leq N \leq 100\,000$$

$$1 \leq Q \leq 100\,000$$

$$-10^8 \leq x_i, y_i \leq 10^8 \text{ minden } i = 1 \dots N + Q\text{-ra}$$

Időlimit: 0.5 mp.

Memórialimit: 256 MB

Pontozás

A megoldásokat sok különböző tesztesetre lefuttatjuk. A tesztesetek részfeladatokba vannak csoportosítva. Egy-egy részfeladatot akkor tekintünk megoldottnak, ha volt legalább egy olyan beadásod, amely az adott részfeladat minden tesztesetére helyes megoldást adott. A feladat összpontszámát a megoldott részfeladatokra kapott pontszámok összege adja.

Részfeladat	Korlátok	Pontszám
0	a minta	0
1	$N \leq 100, Q = 1$	15
2	$Q = 1$	15
3	nincsenek további megkötések	20

Versenypálya

Az esedékes országos terepkerékpár versenyt egy különleges pályán tartják. A pálya N elágazási pontot tartalmaz és bizonyos elágazási pontpárok között (közvetlen) egyirányú pályaszakaszok vannak. Minden versenyző az S startpontból indul és a C célpontba kell érkeznie, de az útvonalát szabadon választhatja meg. A pályára teljesül, hogy bármely elágazási pontot elhagyva nem létezik olyan útvonal, amin keresztül oda vissza lehetne térni.

Írj programot, amely meghatározza azokat a pályaszakaszokat, amelyeken biztosan áthaladnak a versenyzők, függetlenül attól, hogy milyen útvonalat választanak.

Bemenet

A standard bemenet első sorában négy egész szám van: az elágazási pontok N száma, a pályaszakaszok M száma, az S startpont és a C célpont azonosítója. Az elágazási pontokat az $1, \dots, N$ számokkal azonosítjuk.

A következő M sor mindegyike egy-egy pályaszakaszt megadó U_i, V_i számpárt tartalmaz: az U_i elágazási pontból a V_i elágazási pontba közvetlen pályaszakasz vezet.

Kimenet

A standard kimenet első sorába azon pályaszakaszok K számát kell írni, amelyeken minden versenyző áthalad, bármely útvonalat is választ.

A következő K sor mindegyike egy-egy ilyen pályaszakaszt tartalmazzon a pályaszakasz kezdő- és végpontjának U és V azonosítójával. A pályaszakaszok tetszőleges sorrendben kiírathatók.

Ha nincs ilyen pályaszakasz, akkor az első és egyetlen sorba a 0 számot kell írni.

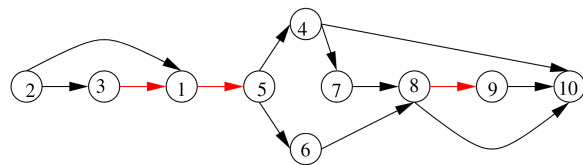
Példa

Bemenet

```
10 13 3 9
2 3
3 1
1 5
5 4
5 6
4 7
6 8
7 8
8 9
8 10
4 10
2 1
9 10
```

Kimenet

```
3
3 1
1 5
8 9
```



Korlátok

$$5 \leq N \leq 100\,000$$

$$N - 1 \leq M \leq 500\,000$$

$$1 \leq U_i \neq V_i \leq N \text{ minden } i = 1 \dots N\text{-re}$$

Bármely két elágazási pontot legfeljebb egy pályaszakasz köt össze.

Időlimit: 0.7 mp.

Memórialimit: 256 MB

Pontozás

A megoldásodat sok különböző tesztesetre lefuttatjuk. A tesztesetek részfeladatokba vannak csoportosítva. Egy-egy részfeladatot akkor tekintünk megoldottnak, ha volt legalább egy olyan beadásod, amely az adott részfeladat minden tesztesetére helyes megoldást adott. A feladat összpontszámát a megoldott részfeladatokra kapott pontszámok összege adja.

Részfeladat	Korlátok	Pontszám
0	a minta	0
1	$N \leq 20, M \leq 100$	5
2	$N \leq 110, M \leq 1000$	5
3	$N \leq 1500, M \leq 10\,000$	5
4	mindenhova van út S-ből és mindenholnan van út C-be	15
5	nincsenek további megkötések	20

Országbérlet

Benedek szeret tömegközlekedéssel utazni. Magyarországon kétféle bérlet kapható: országbérlet és vármegyebérlet. A vármegyebérlet B forintba, az országbérlet C forintba kerül. Mindkettő a megvásárlás napjától kezdve D egymást követő napon érvényes. A vármegyebérlettel csak egy vármegyén belül lehet utazni, az országbérlet azonban vármegyén belül és kívül egyaránt érvényes.

Benedek szeretné megtervezni a bérletvásárlásait a következő N napra az alábbi feltételek mellett. A napokat 1-től N -ig sorszámozzuk.

- Benedeknek minden nap rendelkeznie kell érvényes bérlettel (mindegy, hogy milyen típusúval).
- Benedek az év nagy részében nem hagyja el a vármegyét: az N nap során pontosan K napon fog máshova is utazni és ezeken a napokon érvényes országbérlettel kell rendelkeznie. Ezeknek a napoknak a sorszámai: A_1, A_2, \dots, A_K .

Benedek akármelyik napokon vásárolhat bérletet, és lehet több érvényes bérlete is egyszerre. Kezdetben nincsen semmilyen bérlete. Írj programot, ami megadja, hogy legkevesebb mennyibe fog kerülni Benedeknek a bérletvásárlás a következő N napra.

Bemenet

A standard bemenet első sorában a napok N száma, Benedek utazásainak a K száma, a vármegyebérlet B ára, az országbérlet C ára és a bérletek érvényességének D hossza található.

A második sor K számot tartalmaz: a vármegyén kívüli utazások napjainak A_i sorszámainak.

Kimenet

A standard kimenetre egy sort kell írni egyetlen számmal, a bérletek beszerzésének legkisebb lehetséges költségével.

Példa

Bemenet	Kimenet
10 3 5 6 4	17
1 3 7	

Az első példában egy lehetséges megoldás, ha az alábbi bérleteket veszi meg Benedek:

- Egy országbérletet az 1. napon. Ez az 1, 2, 3 és 4 sorszámú napokon érvényes.
- Egy vármegyebérletet az 5. napon. Ez az 5, 6, 7 és 8 sorszámú napokon érvényes.
- Egy országbérletet a 7. napon. Ez a 7, 8, 9 és 10 sorszámú napokon érvényes.

Ennek a költsége $6 + 5 + 6 = 17$. Belátható, hogy olcsóbban nem oldható meg a bérletvásárlás.

Bemenet	Kimenet
21 4 439 716 9	1871
1 7 16 20	

Korlátok

$$1 \leq N \leq 10^9$$

$$0 \leq K \leq 10\,000$$

$$1 \leq B \leq C \leq 10^9$$

$$1 \leq D \leq 500$$

$$1 \leq A_i \leq N \text{ minden } i = 0 \dots K - 1\text{-re}$$

$$A_i < A_j \text{ bármely } i < j\text{-re}$$

Időlimit: 1.0 mp.

Memórialimit: 256 MB

Pontozás

A megoldásodat sok különböző tesztesetre lefuttatjuk. A tesztesetek részfeladatokba vannak csoportosítva. Egy-egy részfeladatot akkor tekintünk megoldottnak, ha volt legalább egy olyan beadásod, amely az adott részfeladat minden tesztesetére helyes megoldást adott. A feladat összpontszámát a megoldott részfeladatokra kapott pontszámok összege adja.

Részfeladat	Korlátok	Pontszám
0	a minta	0
1	$D = 1$, tehát a bérletek 1 napig érvényesek	9
2	$N \leq 10\,000$	17
3	nincsenek további megkötések	24

Szegfűhalom

Bittonézia fővárosának legelőkelőbb negyede a Szegfűhalom. Itt minden telek rendkívül értékes, de még ezek között is előfordulnak különbségek.

A Szegfűhalom telkeit egy $N \times M$ -es négyzetrácsként modellezzük, ahol minden telek egy 1×1 -es négyzetnek felel meg. Minden telek értéke az $1, 2, \dots, N \cdot M$ számok egyikével jellemezhető és minden telek értéke egyedi (azaz semelyik két telek értéke sem azonos). Magukat az értékeket nem ismerjük, de azt minden egyes telekre tudjuk, hogy a vele élszomszédos telkek között hány olyan van, aminek alacsonyabb az értéke.

Gáborék telke az első sor első mezőjén található. Ki szeretné számolni, hogy mekkora az ő telkük értéke. Írj programot, ami megadja Gáborék telkének a lehető legkisebb és legnagyobb értékét.

Bemenet

A standard bemenet első sorában a négyzetrács sorainak N és az oszlopainak M száma található.

A következő N sor mindegyike pontosan M egész számot tartalmaz. Az i -edik sor j -edik eleme $A_{i,j}$, ami megadja, hogy hány olyan telek van az i -edik sor j -edik telkének szomszédságában, aminek alacsonyabb az értéke.

Kimenet

A standard kimenetre két sort kell írni egy-egy egész értékkel. Az első sorba Gáborék telkének a lehető legalacsonyabb értéke kerüljön, a második sorba pedig a lehető legmagasabb.

Példa

Bemenet	Kimenet
2 3	3
1 2 2	4
1 0 1	

Az alábbi ábrán két példa látható a telkek értékeire. A bal oldalin Gáborék telke a lehető legalacsonyabb, a jobb oldalin pedig a lehető legmagasabb értékű.

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

Bemenet	Kimenet
1 4	1
0 1 1 1	1

Korlátok

$$1 \leq N, M \leq 500$$

$$0 \leq A_{i,j} \leq 4 \text{ minden } i = 1 \dots N\text{-re és } j = 1 \dots M\text{-re}$$

Legalább egyféleképpen meg lehet választani a telkek értékeit úgy, hogy az $A_{i,j}$ értékekkel konzisztensek legyenek.

Időlimit: 2.0 mp.

Memórialimit: 256 MB

Pontozás

A megoldásokat sok különböző tesztesetre lefuttatjuk. A tesztesetek részfeladatokba vannak csoportosítva. Egy-egy részfeladatot akkor tekintünk megoldottnak, ha volt legalább egy olyan beadásod, amely az adott részfeladat minden tesztesetére helyes megoldást adott. A feladat összpontszámát a megoldott részfeladatokra kapott pontszámok összege adja.

Részfeladat	Korlátok	Pontszám
0	a minta	0
1	$N = 1$	8
2	$N = M \leq 3$	10
3	$N = M \leq 4$	10
4	nincsenek további megkötések	22

Minden részfeladatban a megszerezhető pontok 50 %-a jár akkor, ha az összes tesztesetben legalább az egyik kiírt sor helyes értéket tartalmaz, de legalább egy tesztben nem mindkét érték helyes. Ügyelj arra, hogy a programod minden esetben két sort írjon ki, soronként egy egész számmal, különben előfordulhat, hogy az értékelő formai hiba miatt nem adja meg a részpontokat.