

Lámpások

Feladatnév	Lanterns/Lámpások
Input file	standard input
Output file	standard output
Időkorlát	3 seconds
Memóriakorlát	1024 megabytes

János gazda elvitte a tehéncsordáját túrázni az Alpokban. Kis idő múltán besötétedett és ez véget vetett a túrának. Viszont néhány tehén a hegyvonulatok között ragadt, és János gazdának kell kimentenie őket.

A teheneket rejtő hegyvonulat n ponttal ábrázolható, amelyek egy függőleges, két dimenziós síkban helyezkednek el. Ezeket a pontokat a továbbiakban csúcsoknak nevezzük. A csúcsok 1-től n -ig vannak sorszámozva, sorrendben balról jobbra. Az i . csúcsnak a koordinátái (i, h_i) . A h_i érték az i . csúcs **magasságát** jelöli. Garantált, hogy a h_1, h_2, \dots, h_n értékek az $1 \dots n$ számok egy permutációját alkotják. (Ez azt jelenti, hogy minden $j = 1, \dots, n$ esetén pontosan egy olyan $i \in \{1, \dots, n\}$ van, amelyre $h_i = j$.)

Minden i -re ($1 \leq i < n$) az i . és $i + 1$. csúcs egy egyenes szakasszal van összekötve.

Mivel éjszaka van, János gazda csak úgy mozoghat a hegyvonulaton, ha van nála legalább egy működő lámpás. Szerencsére van k darab megvásárolható lámpás a csúcsokon. A j . lámpás ($1 \leq j \leq k$) a p_j csúcson vásárolható meg c_j frankért.

Sajnos a j . lámpás csak akkor működik, ha János gazda pillanatnyi magassága adott $[a_j, b_j]$ intervallumban van. Más szóval, a j . lámpás nem működik, ha János gazda jelenlegi magassága szigorúan kisebb mint a_j vagy szigorúan nagyobb, mint b_j . Tudjuk, hogy a lámpások nem mennek tönkre, ha elhagyják az intervallumukat. Például, ha János magassága túllépi b_j -t, akkor a j . lámpás nem működik, de amint ismét b_j magasságba ereszkedik, a lámpás megint működni fog.

Amennyiben János gazda jelenleg a p . csúcson van, az alábbi három dolgot teheti:

- Megvehet egyet a p . csúcson elérhető lámpások közül. Ha megvesz egy lámpást, azt utána örökké használhatja.
- Ha $p > 1$, átmehet a $p - 1$. csúcsra.
- Ha $p < n$, átmehet a $p + 1$. csúcsra.

János soha sem mozoghat működő lámpás nélkül. Csak akkor sétálhat át egy csúcsról egy szomszédos csúcsra, ha a séta minden pillanatában legalább egy meglévő lámpása működik. (Egy séta alatt nem csak egy lámpást használhat.)

Például tegyük fel, hogy János gazda egy 4 magasságú csúcson van, és át szeretne menni egy szomszédos 1 magasságú csúcsra. Ha van neki két lámpása, amelyek közül az egyik az $[1, 3]$, a másik a $[3, 4]$ magasságintervallumban működik, akkor át tud sétálni az egyik csúcsról a másikra.

Viszont ha Jánosnak olyan lámpásai vannak, amelyek az $[1, 1]$ és $[2, 5]$ intervallumokban működnek, akkor nem tud átsétálni a két csúcs között, hiszen például egyik lámpás sem működik 1.47 magasságban.

Több független kérdésre kell meghatároznod a választ.

Minden olyan j -re ($1 \leq j \leq k$), amelyre igaz, hogy $a_j \leq h_{p_j} \leq b_j$, tegyük fel, hogy János a p_j csúcson kezdi a bejárást, a j . lámpás megvételével. Ahhoz, hogy bejárja az egész hegyvonulatot, az n csúcs mindegyikét legalább egyszer meg kell látogatnia, a fenti háromféle lépés ismételt végrehajtásával. Minden ilyen j -re határozd meg azt a minimális teljes költséget frankban, amennyit ki kell fizetnie ahhoz, hogy bejárja a teljes hegyvonulatot. (Ez a költség tartalmazza a j . lámpás kezdeti megvásárlását.)

Input

A bemenet első sora két egész számot tartalmaz: n és k ($1 \leq n \leq 2000$, $1 \leq k \leq 2000$) - a hegycsúcsok számát és az elérhető lámpások számát, ilyen sorrendben.

A második sorban n darab, szóközzel elválasztott egész szám van: h_1, h_2, \dots, h_n ($1 \leq h_i \leq n$) - az egyes csúcsok magasságai. Biztosan tudjuk, hogy a h_i számok az $1 \dots n$ számok egy permutációja.

A j . sor a következő k sor közül négy, szóközzel elválasztott egész számot tartalmaz: p_j, c_j, a_j , és b_j ($1 \leq p_j \leq n$, $1 \leq c_j \leq 10^6$, $1 \leq a_j \leq b_j \leq n$) - a j . lámpás melyik csúcson vásárolható meg, a lámpás árát és a működési intervallumának határait, ebben a sorrendben.

Output

Minden j -re ($1 \leq j \leq k$):

- Ha h_{p_j} kívül esik az $[a_j, b_j]$ intervallumon, a kimenet legyen -1 .
- Különben, ha János gazda nem tudja bejárni a teljes hegyvonulatot a j . lámpással kezdve a vásárlást, a kimenet legyen -1 .
- Különben, a kimenet legyen az elköltött teljes költség frankban, amit János gazdának a teljes hegyvonulat bejárására kell költenie, a j . lámpással kezdve a vásárlást,

Pontozás

Öt tesztcsoport van:

Az 1. csoport értéke 9 pont: $n \leq 20$ és $k \leq 6$.

A 2. csoport értéke 12 pont: $n \leq 70$ és $k \leq 70$.

A 3. csoport értéke 23 pont: $n \leq 300$, $k \leq 300$ és $h_i = i$ minden $1 \leq i \leq n$.

A 4. csoport értéke 16 pont: $n \leq 300$, $k \leq 300$.

Az 5. csoport értéke 40 pont: nincs további megkötés.

Példa

standard input	standard output
7 8	7
4 2 3 1 5 6 7	-1
3 1 2 4	4
1 2 1 3	10
4 4 1 7	30
6 10 1 7	-1
6 20 6 6	-1
6 30 5 5	-1
7 40 1 6	
7 50 7 7	

Megjegyzés

Ha János gazda az 1. lámpást a 2. csúcson vásárolja meg, akkor a következő módon tudja bejárni a hegyvonulatot:

- balra indul az 1. csúcsra
- megveszi a 2. sorszámú lámpást
- jobbra mozog a 4. csúcsig
- megveszi a 3. számú lámpást
- jobbra megy a 7. csúcsig.

Ekkor János gazda minden csúcson járt már legalább egyszer és összesen $1 + 2 + 4 = 7$ frankot költött.

János gazda nem kezdheti a vásárlást a 2., 6., vagy a 7. lámpással, mivel azok nem működnek azokon a csúcson, amiken meg lehet őket vásárolni. Ezekre a lámpásokra a válasz -1 .

Ha János gazda a 3. vagy a 4. lámpással kezd, akkor minden csúcsot meg tud látogatni újabb lámpás vásárlása nélkül.

Ha az 5. lámpással kezdi a vásárlást, akkor a 3. lámpást még később meg kell vennie.

Ha a 8. lámpással kezd, akkor megakad a 7. csúcsnál. Még ha megveszi a 7. lámpást is, akkor sem tud a 7. csúcsról a 6. csúcsra jutni.