

## Legmagasabb (Highest)

Egy alternatív univerzumban Vlad a Poenari erőd egy futurisztikus változatában rekedt, amely most  $n$  emeletet foglal magába, 0-tól  $n - 1$ -ig. Minden egyes  $i$  emeletről ( $0 \leq i \leq n - 1$ ) csak úgy tud felfelé menni, hogy vagy a lépcsőn megy, és fizet 1 vércseppet (ezzel a pénznemmel fizetnek a vámpírok Romániában), vagy denevérré változik, és átmegy a szellőzőnyílásokon, amiért 2 vércseppet kell fizetnie. A lépcsőn  $v[i]$  emelettel, míg a szellőzőnyílásokon  $w[i]$  emelettel juthat feljebb, ahol  $v$  és  $w$  két adott tömb:  $v = v[0], v[1], \dots, v[n - 1]$  és  $w = w[0], w[1], \dots, w[n - 1]$ .

Formálisan, az  $i$  emeletről ( $0 \leq i \leq n - 1$ ) Vlad mehet:

- az  $i + 1$  emelettől az  $i + v[i]$  emeletig bárhová, anélkül, hogy túlmenne az  $n - 1$  emeleten, 1 költséggel.
- az  $i + 1$  emelettől az  $i + w[i]$  emeletig bárhová anélkül, hogy túlmenne az  $n - 1$  emeleten, 2 költséggel.

Továbbá, testvérei Radu és Mircea  $m$  forgatókönyvet javasoltak Vlad számára, amelyek mindegyike két,  $A$  és  $B$  emeletből áll ( $A \leq B$ ). Vladnak a következő  $m$  kérdésre kell válaszolnia: mennyi a legkevesebb vér, amit fel kell áldoznia ahhoz, hogy az  $A$  emeletről az  $B$  emeletre jusson?

## Implementációs részletek

A `solve` függvényt kell implementálnotok:

```
std::vector<int> solve(std::vector<int> &v, std::vector<int> &w,  
    std::vector<std::pair<int,int>> &queries);
```

- Megkapja  $v$  vektort, ami a lépcsőházak magasságait és a  $w$  vektort, ami a szellőzőrendszerek magasságait tárolja, mindkettő  $n$  méretű.
- Megkapja a lekérdezéseket is, egy  $m$  méretű párokból álló vektort. Minden pár tartalmazza az utasításban leírtak szerint az  $A$  és  $B$  értékeket.
- Visszaad egy  $m$  méretű vektort, amely az  $m$  lekérdezésre adott válaszokból áll.

## Korlátok

- $1 \leq n, m \leq 500\,000$ .
- $1 \leq v[i], w[i] \leq n$  minden  $0 \leq i \leq n - 1$  esetén.
- $0 \leq A \leq B \leq n - 1$  minden lekérdezésre.

## Részfeladatok

1. (5 pont)  $1 \leq n \leq 300$ ,  $1 \leq m \leq 500\,000$
2. (7 pont)  $1 \leq n \leq 3\,000$ ,  $1 \leq m \leq 3\,000$
3. (11 pont)  $1 \leq n \leq 20\,000$ ,  $1 \leq m \leq 20\,000$
4. (44 pont)  $1 \leq n \leq 200\,000$ ,  $1 \leq m \leq 200\,000$
5. (8 pont)  $1 \leq n \leq 500\,000$ ,  $1 \leq m \leq 500\,000$ ,  $v[i] \leq v[j]$  és  $w[i] \leq w[j]$  minden  $0 \leq i < j \leq n-1$  esetén
6. (25 pont) Nincsenek egyéb megkötések.

## Példák

### 1. példa

Legyen a következő hívás:

```
solve({2, 3, 1, 1, 1, 1, 2}, {3, 4, 1, 2, 1, 2, 2},  
      {{0, 4}, {0, 5}, {0, 6}})
```

Itt  $n = 7$  és 3 lekérdezésünk van,  $v = [2, 3, 1, 1, 1, 1, 2]$  és  $w = [3, 4, 1, 2, 1, 2, 2]$ .

Az első,  $(0, 4)$  lekérdezéshez Vladnak két 1 költségű ugrást kell végrehajtania: 0-ról 1-re (bár ugorhat 2-re is, de az 1 emeletről tovább fog jutni), majd 1-ről 4-re. Teljes költség:  $1 + 1 = 2$ .

A  $(0, 5)$  második lekérdezéshez 2 optimális útvonal létezik: 0-ról 1-re (1 költség), 1-ről 4-re (1 költség), 4-ről 5-re (1 költség); a második útvonal: 0-ról 1-re (1 költség), 1-ről 5-re (2 költség). Teljes költség:  $1 + 1 + 1 = 1 + 2 = 3$ .

A harmadik,  $(0, 6)$  lekérdezésnél a 4 költségű útvonal például a következő: 0-ról 1-re (1 költség), 1-ről 5-re (2 költség), 5-ről 6-ra (1 költség). Teljes költség:  $1 + 2 + 1 = 4$ .

Tehát a függvény által visszaadott vektornak a következőnek kell lennie:

```
[2, 3, 4]
```

### 2. példa

Legyen a következő hívás:

```
solve({1, 1, 1, 2, 3, 2, 1, 1, 2, 3}, {2, 4, 1, 4, 1, 4, 1, 3, 2, 3},  
      {{3, 9}, {0, 9}, {0, 7}, {0, 4}, {3, 5}})
```

Ezek az egyes lekérdezésekhez tartozó optimális utak:

(3,9): 3-ról 5-re (1 költség), 5-ről 9-re (2 költség)  $\implies$  összesen: 3

(0,9): 0-ról 1-re (1 költség), 1-ről 5-re (2 költség), 5-ről 9-re (2 költség)  $\implies$  összesen: 5

(0,7): 0-ról 1-re (1 költség), 1-ről 5-re (2 költség), 5-ről 7-re (1 költség)  $\implies$  összesen: 4

(0,4): 0-ról 1-re (1 költség), 1-ről 4-re (2 költség)  $\implies$  összesen: 3

(3,5): 3-ról 5-re (1 költség)  $\implies$  összesen: 1

Tehát a függvény által visszaadott vektornak a következőnek kell lennie:

```
[3, 5, 4, 3, 1]
```

## Minta értékelő

A minta értékelő a bemenetet a következő formátumban olvassa:

- 1. sor:  $n$
- 2. sor:  $v[0] \ v[1] \dots v[n-1]$
- 3. sor:  $w[0] \ w[1] \dots w[n-1]$
- 4. sor:  $m$
- $5 + i$ . sor ( $0 \leq i < m$ ):  $A \ B$

és  $m$  sorba írja ki a `solve` függvény hívásából kapott eredményt.