



Rádiové Veže

Na Jakarte je N rádiových veží. Veže sú umiestnené na priamke a očíslované 0 až $N - 1$ zľava doprava. Pre každé i , $0 \leq i \leq N - 1$, je výška veže i rovná $H[i]$ metrov. Výšky veží sú **navzájom rôzne**.

Pre nejakú kladnú rušivosť komunikácie δ platí, že každá dvojica veží i a j (kde $0 \leq i < j \leq N - 1$) sa navzájom počuje práve vtedy, keď existuje medziveža k , ktorá

- je napravo od veže i a naľavo od veže j , teda platí $i < k < j$, a zároveň
- výšky veží i a j sú obe nanajvyššie rovné $H[k] - \delta$ metrov.

Ujo Dengklek si chce prenajať niektoré rádiové veže a vytvoriť si rádiovú sieť.

Vašou úlohou je odpovedať na Q Dengklekových otázok ktoré sú nasledovného typu: pre dané parametre L , R a D ($0 \leq L \leq R \leq N - 1$ a $D > 0$), aký je najväčší počet veží ktoré si vie Dengklek prenajať, ak musia byť splnené nasledovné podmienky:

- Dengklek si smie prenajať len veže s číslami L až R (vrátane), a zároveň
- rušivosť komunikácie δ je rovná D , a zároveň
- každá dvojica veží ktoré si Dengklek prenajal, sa musí navzájom počuť

Špeciálne vás upozorňujeme, že každé dve prenajaté veže môžu na to, aby sa počuli, používať ľubovoľnú vyhovujúcu medzivežu. Použitá medziveža nemusí byť prenajatá.

Implementačné detaily

Naimplementujte nasledovné funkcie:

```
void init(int N, int[] H)
```

- N : počet rádiových veží.
- H : pole dĺžky N popisujúce výšky veží
- Táto funkcia bude zavolaná práve raz, pred prvým volaním funkcie `max_towers`.

```
int max_towers(int L, int R, int D)
```

- L , R : hranice úseku veží, ktoré si smie Dengklek prenajať
- D : hodnota rušivosti komunikácie, δ .

- Táto funkcia má vrátiť najväčší počet veží, ktoré si vie Denglek prenajať pre jeho rádiovú sieť, ak si môže prenajímať veže od L po R (vrátane) a rušivosť komunikácie δ je D .
- Táto funkcia bude zavolaná Q -krát.

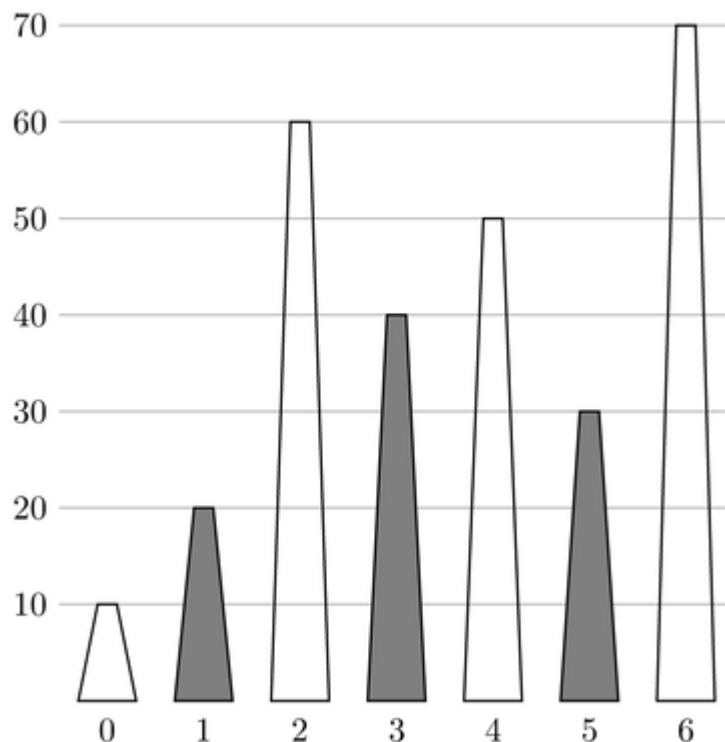
Príklad

Uvažujme nad nasledovnými volaniami:

```
init(7, [10, 20, 60, 40, 50, 30, 70])
```

```
max_towers(1, 5, 10)
```

Ujo Denglek si vie prenajať veže 1, 3 a 5. Tento príklad je ilustrovaný na nasledovnom obrázku, kde zafarbené lichobežníky sú prenajaté veže.



Veže 3 a 5 sa počujú vďaka veži 4, keďže $40 \leq 50 - 10$ a $30 \leq 50 - 10$. Veže 1 a 3 sa počujú vďaka veži 2. Veže 1 a 5 sa počujú vďaka veži 3. Neexistuje spôsob ako si prenajať viac ako 3 veže, volanie má teda vrátiť 3.

```
max_towers(2, 2, 100)
```

V úseku povolených veží je práve jedna veža, preto si Denglek vie prenajať iba 1 vežu. Volanie má teda vrátiť 1.

```
max_towers(0, 6, 17)
```

Dengklek si vie prenajať veže 1 a 3. Veže 1 a 3 sa počujú vďaka veži 2, keďže platí $20 \leq 60 - 17$ a $40 \leq 60 - 17$. Neexistuje spôsob ako si prenajať viac ako dve veže, volanie má teda vrátiť 2.

Obmedzenia

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq H[i] \leq 10^9$ (pre každé i pre ktoré platí $0 \leq i \leq N - 1$)
- $H[i] \neq H[j]$ (pre každé i a j pre ktoré platí $0 \leq i < j \leq N - 1$)
- $0 \leq L \leq R \leq N - 1$
- $1 \leq D \leq 10^9$

Podúlohy

1. (4 body) Existuje taká veža k ($0 \leq k \leq N - 1$), že
 - pre každé i pre ktoré platí $0 \leq i \leq k - 1$: $H[i] < H[i + 1]$, a zároveň
 - pre každé i pre ktoré platí $k \leq i \leq N - 2$: $H[i] > H[i + 1]$.
2. (11 bodov) $Q = 1$, $N \leq 2000$
3. (12 bodov) $Q = 1$
4. (14 bodov) $D = 1$
5. (17 bodov) $L = 0$, $R = N - 1$
6. (19 bodov) Hodnota D je rovnaká naprieč všetkými volaniami funkcie `max_towers`.
7. (23 points) Žiadne obmedzenia navyše.

Sample Grader

Ukázkový grader očakáva vstup v nasledovnom formáte:

- riadok 1: N Q
- riadok 2: $H[0]$ $H[1]$... $H[N - 1]$
- riadok $3 + j$ ($0 \leq j \leq Q - 1$): L R D pre otázku číslo j

Ukázkový grader vypíše návratové hodnoty tvojich funkcií nasledovne:

- riadok $1 + j$ ($0 \leq j \leq Q - 1$): návratová hodnota `max_towers` pre otázku číslo j