



## Torri radio

A Giacarta ci sono  $N$  torri radio disposte in linea retta, la  $i$ -esima di altezza  $H[i]$  (per  $0 \leq i < N$ ). Le altezze sono tutte **distinte**.

Dato un valore di potenza  $D$ , la coppia di torri  $i$  e  $j$  ( $0 \leq i < j \leq N - 1$ ) può comunicare a quella potenza solo se c'è una torre intermedia  $k$  ( $i < k < j$ ) tale che:

- le altezze delle torri  $i$  e  $j$  sono **minori o uguali** di  $H[k] - D$ .

Pak Dengklek vuole affittare alcune torri per la sua trasmissione. Per aiutarlo, devi rispondere a  $Q$  domande di questo tipo: dati  $L, R$  ( $0 \leq L \leq R < N$ ) e  $D > 0$ , qual è il massimo numero di torri che può affittare, assumendo che

- Pak può solo affittare torri con indici tra  $L$  ed  $R$  (inclusi),
- ogni coppia di torri affittate deve poter comunicare a potenza  $D$  (anche usando torri intermedie non affittate)?

## Dettagli di implementazione

Devi implementare la seguente funzione:

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

- $N$ : il numero di torri radio.
- $H$ : un array di lunghezza  $N$  contenente le altezze delle torri.
- Questa funzione è chiamata esattamente una volta, prima di ogni chiamata a `max_towers`.

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

- $L, R$ : i limiti dell'intervallo di torri disponibili.
- $D$ : il valore di potenza richiesto.
- La funzione deve restituire il massimo numero di torri radio nell'intervallo tra  $L$  ed  $R$  (inclusi) che Pak può affittare, facendo in modo che ogni coppia di torri affittate possa comunicare.
- Questa funzione è chiamata esattamente  $Q$  volte.

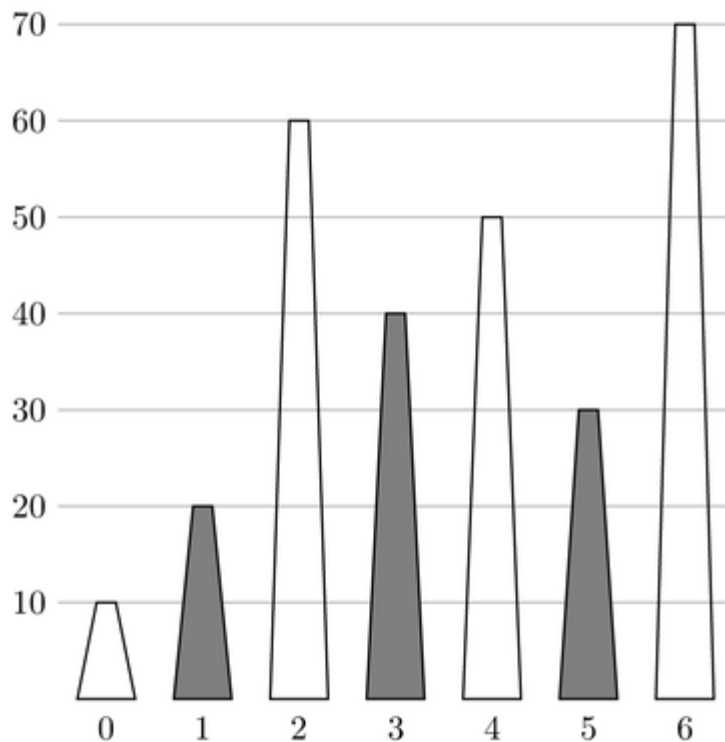
## Caso di esempio

Considera la seguente sequenza di chiamate:

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

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

Pak Dengklek può affittare le torri 1, 3 e 5. L'esempio è illustrato nella seguente immagine, dove le torri colorate sono quelle affittate.



Le torri 3 e 5 possono comunicare con 4 come intermediario, dato che  $40 \leq 50 - 10$  e  $30 \leq 50 - 10$ . Le torri 1 e 3 possono comunicare con 2 come intermediario. Le torri 1 e 5 possono comunicare con 3 come intermediario. Non si possono affittare più di 3 torri tutte comunicanti a potenza 10, quindi la funzione deve restituire 3.

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

In questo intervallo c'è una sola torre, e dato che Pak può affittarla la funzione deve restituire 1.

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

Pak può affittare le torri 1 e 3. Le torri 1 e 3 possono comunicare con 2 come intermediario, dato che  $20 \leq 60 - 17$  e  $40 \leq 60 - 17$ . Non si possono affittare più di 2 torri tutte comunicanti a potenza 17, quindi la funzione deve restituire 2.

## Assunzioni

- $1 \leq N \leq 100\,000$ .
- $1 \leq Q \leq 100\,000$ .
- $1 \leq H[i] \leq 10^9$  (per  $0 \leq i < N$ ).
- $H[i] \neq H[j]$  (per  $i \neq j$ ).
- $0 \leq L \leq R < N$ .
- $1 \leq D \leq 10^9$ .

## Subtask

1. (4 punti) Esiste un  $k$  ( $0 \leq k < N$ ) per cui
  - $H[i] < H[i + 1]$  per ogni  $i < k$ , e
  - $H[i] > H[i + 1]$  per ogni  $i \geq k$ .
2. (11 punti)  $Q = 1$ ,  $N \leq 2000$ .
3. (12 punti)  $Q = 1$ .
4. (14 punti)  $D = 1$ .
5. (17 punti)  $L = 0$ ,  $R = N - 1$ .
6. (19 punti) Il valore di  $D$  è lo stesso in tutte le chiamate a `max_towers`.
7. (23 punti) Nessuna limitazione aggiuntiva.

## Grader di esempio

Il grader di esempio legge l'input secondo il seguente formato:

- riga 1:  $N$   $Q$
- riga 2:  $H[0]$   $H[1]$  ...  $H[N - 1]$
- riga  $3 + j$  ( $0 \leq j < Q$ ):  $L$   $R$   $D$  per la domanda  $j$

Il grader di esempio stampa l'output secondo il seguente formato:

- riga  $1 + j$  ( $0 \leq j < Q$ ): il valore restituito da `max_towers` per la domanda  $j$