



## Torres de Rádio

Existem  $N$  torres de rádio em Jakarta. As torres estão distribuídas ao longo de uma linha reta e são numeradas de 0 a  $N - 1$ , da esquerda para a direita. Para cada  $i$  tal que  $0 \leq i \leq N - 1$ , a altura da torre  $i$  é  $H[i]$  metros. As alturas de cada torre são **distintas**.

Para algum valor positivo de interferência  $\delta$ , um par de torres  $i$  e  $j$  (onde  $0 \leq i < j \leq N - 1$ ) podem comunicar entre si se e só se existir uma torre intermediária  $k$  tal que

- a torre  $i$  está à esquerda da torre  $k$  e a torre  $j$  está à direita da torre  $k$ , isto é,  $i < k < j$ , e
- as alturas das torres  $i$  e  $j$  são ambas no máximo  $H[k] - \delta$  metros.

O Pak Dengklek quer alugar algumas torres de rádio para a sua nova rede de rádio. A tua tarefa é responder a  $Q$  questões do Pak Dengklek que são da seguinte forma: dados os parâmetros  $L$ ,  $R$  e  $D$  ( $0 \leq L \leq R \leq N - 1$  e  $D > 0$ ), qual é o número máximo de torres que o Pak Dengklek pode alugar, assumindo que

- o Pak Dengklek pode apenas alugar torres com os índices entre  $L$  e  $R$  (inclusive), e
- o valor de interferência  $\delta$  é  $D$ , e
- qualquer par de torres de rádio que o Pak Dengklek alugue deve ser capaz de comunicar entre si.

Nota que duas torres alugadas podem comunicar entre si utilizando uma torre intermediária  $k$  independentemente da torre  $k$  estar alugada ou não.

## Detalhes de implementação

Deves implementar as seguintes funções:

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

- $N$ : o número de torres de rádio.
- $H$ : um array de tamanho  $N$  que descreve a altura das torres.
- A função é chamada exatamente uma vez, antes de qualquer chamada a `max_towers`.

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

- $L$ ,  $R$ : os limites de um intervalo de torres.
- $D$ : o valor de  $\delta$ .

- Esta função deve devolver o número máximo de torres de rádio que o Pak Dengklek pode alugar para a sua nova rede de rádio se apenas for autorizado a alugar torres entre a torre  $L$  e a torre  $R$  (inclusive) e o valor de  $\delta$  for  $D$ .
- Esta função é chamada exatamente  $Q$  vezes.

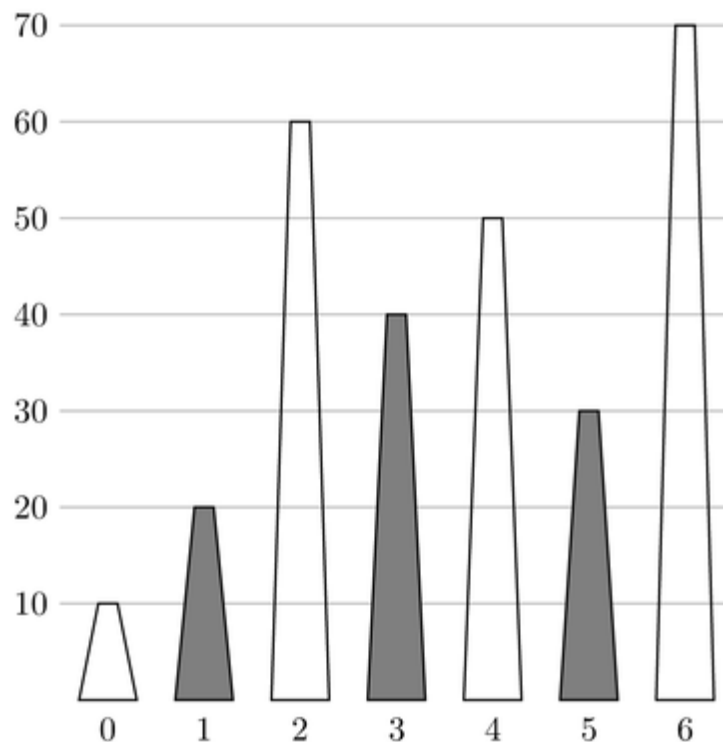
## Exemplo

Considera a seguinte sequência de chamadas:

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

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

O Pak Dengklek pode alugar as torres 1, 3, e 5. O exemplo é ilustrado na figura seguinte, onde os trapézios sombreados representam as torres alugadas.



As torres 3 e 5 podem comunicar utilizando a torre 4 como intermediária, já que  $40 \leq 50 - 10$  e  $30 \leq 50 - 10$ . As torres 1 e 3 podem comunicar utilizando a torre 2 como intermediária. As torres 1 e 5 podem comunicar utilizando a torre 3 como intermediária. Não existe qualquer forma de alugar mais que 3 torres, portanto a função deve devolver 3.

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

Existe apenas 1 torre no intervalo, então o Pak Dengklek pode apenas alugar 1 torre. Portanto a função deve devolver 1.

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

O Pak Dengklek pode alugar as torres 1 e 3. As torres 1 e 3 podem comunicar utilizando a torre 2 como intermediária, já que  $20 \leq 60 - 17$  e  $40 \leq 60 - 17$ . Não existe qualquer forma de alugar mais que 2 torres, portanto a função deve devolver 2.

## Restrições

- $1 \leq N \leq 100\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq H[i] \leq 10^9$  (para cada  $i$  tal que  $0 \leq i \leq N - 1$ )
- $H[i] \neq H[j]$  (para cada  $i$  e  $j$  tal que  $0 \leq i < j \leq N - 1$ )
- $0 \leq L \leq R \leq N - 1$
- $1 \leq D \leq 10^9$

## Subtarefas

1. (4 pontos) Existe uma torre  $k$  ( $0 \leq k \leq N - 1$ ) tal que
  - Para cada  $i$  tal que  $0 \leq i \leq k - 1$ :  $H[i] < H[i + 1]$ , e
  - Para cada  $i$  tal que  $k \leq i \leq N - 2$ :  $H[i] > H[i + 1]$ .
2. (11 pontos)  $Q = 1$ ,  $N \leq 2000$
3. (12 pontos)  $Q = 1$
4. (14 pontos)  $D = 1$
5. (17 pontos)  $L = 0$ ,  $R = N - 1$
6. (19 pontos) O valor de  $D$  é o mesmo para todas as chamadas a `max_towers`.
7. (23 pontos) Sem restrições adicionais.

## Avaliador exemplo

O avaliador exemplo lê o input no seguinte formato:

- linha 1:  $N$   $Q$
- linha 2:  $H[0]$   $H[1]$   $\dots$   $H[N - 1]$
- linha  $3 + j$  ( $0 \leq j \leq Q - 1$ ):  $L$   $R$   $D$  para a questão  $j$

O avaliador exemplo imprime as tuas respostas no seguinte formato:

- linha  $1 + j$  ( $0 \leq j \leq Q - 1$ ): o valor devolvido por `max_towers` para a questão  $j$