



## Torres de radio

Hay  $N$  torres de radio en Jakarta. Las torres están ubicadas a lo largo de una línea recta y numeradas de izquierda a derecha desde 0 hasta  $N - 1$ . Para cada  $i$  tal que  $0 \leq i \leq N - 1$ , la altura de la torre  $i$  es de  $H[i]$  metros. Las alturas de las torres son **distintas**.

Para algún valor de interferencia positivo  $\delta$ , dos torres  $i$  y  $j$  (donde  $0 \leq i < j \leq N - 1$ ) pueden comunicarse entre ellas, sí y solo sí existe una torre intermediaria  $k$  tal que

- la torre  $i$  está a la izquierda de la torre  $k$  y la torre  $j$  está a la derecha de la torre  $k$ , es decir,  $i < k < j$ , y
- las alturas de la torre  $i$  y la torre  $j$  son ambas de a lo sumo  $H[k] - \delta$  metros.

Pak Dengklek quiere alquilar algunas torres de radio para su nueva estación. Tu tarea es responder  $Q$  preguntas de Pak Dengklek, que son de la siguiente manera: dados los parámetros  $L, R$  y  $D$  ( $0 \leq L \leq R \leq N - 1$  y  $D > 0$ ), cuál es el máximo número de torres que Pak Dengklek puede alquilar, asumiendo que:

- Pak Dengklek solo puede alquilar torres con índices entre  $L$  y  $R$  (inclusive), y
- el valor de interferencia  $\delta$  es  $D$ , y
- cualquier par de torres de radio que Pak Dengklek alquile debe ser capaz de comunicarse entre ellas.

Ten en cuenta que dos torres alquiladas pueden comunicarse usando una torre intermediaria  $k$ , sin importar si la torre  $k$  está alquilada o no.

## Detalles de implementación

Debes implementar las siguientes funciones:

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

- $N$ : el número de torres de radio.
- $H$ : un arreglo de tamaño  $N$  describiendo las alturas de las torres.
- Esta función se llama solo una vez, antes de cualquier llamada a `max_towers`.

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

- $L, R$ : los límites de un rango de torres.

- $D$ : el valor de  $\delta$ .
- Esta función debe retornar el máximo número de torres de radio que Pak Dengklek puede alquilar para su nueva estación si solo tiene permitido alquilar torres entre la torre  $L$  y la torre  $R$  (inclusive) y si el valor de  $\delta$  de  $D$ .
- Esta función es llamada exactamente  $Q$  veces.

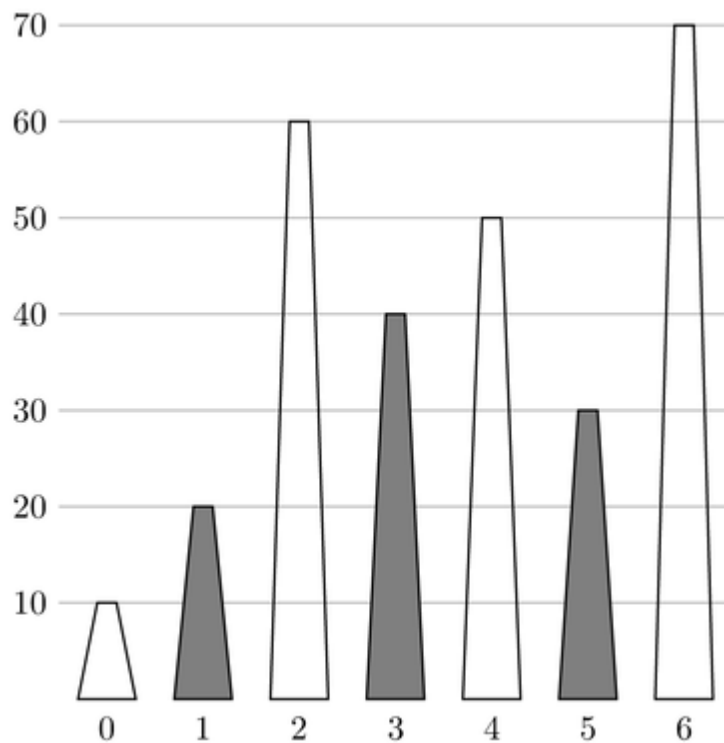
## Ejemplo

Considera las siguiente secuencia de llamadas:

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

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

Pak Dengklek puede alquilar las torres 1, 3, y 5. El ejemplo está ilustrado en la siguiente imagen, donde los trapecoides sombreados representan las torres alquiladas.



Las torres 3 y 5 pueden comunicarse usando la torre 4 como intermediaria, dado que  $40 \leq 50 - 10$  y  $30 \leq 50 - 10$ . Las torres 1 y 3 se pueden comunicar usando la torre 2 como intermediaria. Las torres 1 y 5 se pueden comunicar usando la torre 3 como intermediaria. No hay forma de alquilar más de 3 torres, por lo que la función debería retornar 3.

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

Solo hay 1 torre en el rango, así que Pak Dengklek solo puede alquilar 1 torre. Por lo tanto, la función debería retornar 1.

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

Pak Dengklek puede alquilar las torres 1 y 3. Las torres 1 y 3 se pueden comunicar usando la torre 2 como intermediaria, dado que  $20 \leq 60 - 17$  y  $40 \leq 60 - 17$ . No hay forma de alquilar más de 2 torres, por lo que la función debería retornar 2.

## Restricciones

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

## Subtareas

1. (4 puntos) Existe una torre  $k$  ( $0 \leq k \leq N - 1$ ) tal que
  - para cada  $i$  tal que  $0 \leq i \leq k - 1$ :  $H[i] < H[i + 1]$ , y
  - para cada  $i$  tal que  $k \leq i \leq N - 2$ :  $H[i] > H[i + 1]$ .
2. (11 puntos)  $Q = 1$ ,  $N \leq 2000$
3. (12 puntos)  $Q = 1$
4. (14 puntos)  $D = 1$
5. (17 puntos)  $L = 0$ ,  $R = N - 1$
6. (19 puntos) El valor de  $D$  es el mismo en todas las llamadas a `max_towers`.
7. (23 puntos) Sin restricciones adicionales.

## Evaluador local

El evaluador lee la entrada en el siguiente formato:

- línea 1:  $N$   $Q$
- línea 2:  $H[0]$   $H[1]$   $\dots$   $H[N - 1]$
- línea  $3 + j$  ( $0 \leq j \leq Q - 1$ ):  $L$   $R$   $D$  para la pregunta  $j$

El evaluador imprime tus respuestas en el siguiente formato:

- línea  $1 + j$  ( $0 \leq j \leq Q - 1$ ): el valor de retorno de `max_towers` para la pregunta  $j$