



## Torres de Radio

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

Para algún valor de interferencia positivo  $\delta$ , un par de torres  $i$  y  $j$  (donde  $0 \leq i < j \leq N - 1$ ) pueden comunicarse una con otra si y solo si 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  $k$ , esto es,  $i < k < j$ , y
- las alturas de la torre  $i$  y la torre  $j$  son ambas a lo más  $H[k] - \delta$  metros.

Pak Dengklek quiere colocar algunas torres de radio para su nueva red de radio. Tu tarea es responder  $Q$  consultas de Pak Dengklek las cuáles son de la siguiente forma: dados parámetros  $L, R$  y  $D$  ( $0 \leq L \leq R \leq N - 1$  and  $D > 0$ ), cuál es el máximo número de torres que Pak Dengklek puede colocar, asumiendo que:

- Pak Dengklek puede únicamente colocar 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 coloca debe poderse comunicar con las otras que coloca.

Nota que dos torres colocadas pueden comunicarse usando una torre intermediaria  $k$ , independientemente de si la torre  $k$  es colocada o no.

## Detalles de Implementación

Tu debes implementar las siguientes funciones:

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

- $N$ : el número de torres de radio.
- $H$ : un vector de longitud  $N$  describiendo las alturas de las torres.
- Este procedimiento es llamado exactamente una vez, antes de cualquier llamada a `max_towers`.

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

- $L, R$ : los límites del rango de torres.
- $D$ : el valor de  $\delta$ .
- Esta función debe retornar el máximo número de torres de radio que Pak Dengklek puede colocar para su nueva red de radio, si solo se le permite colocar torres que se encuentren entre la torre  $L$  y la torre  $R$  (inclusive) y el valor de  $\delta$  es  $D$ .
- Este procedimiento es llamado exactamente  $Q$  veces.

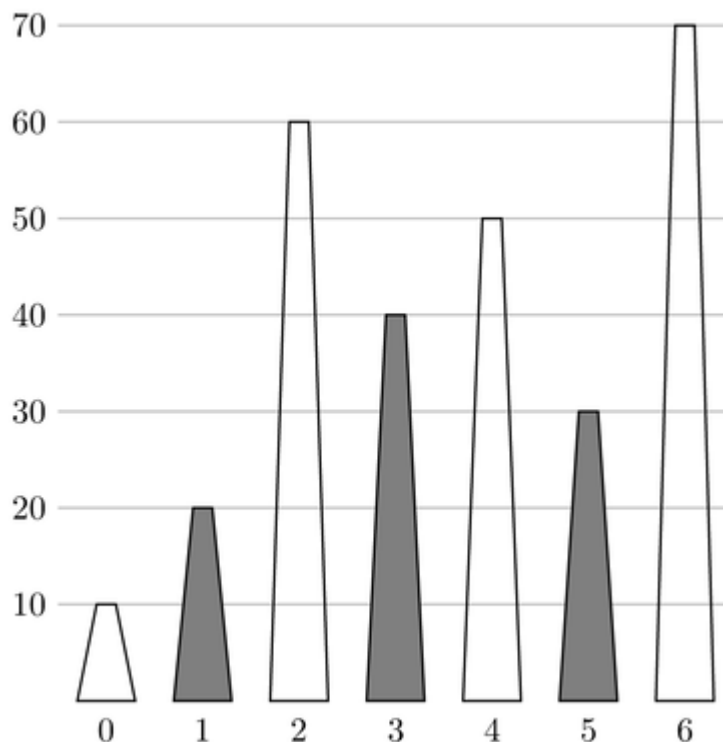
## Ejemplo

Considera la siguiente secuencia de llamadas:

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

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

Pak Dengklek puede colocar las torres 1, 3, y 5. El ejemplo es ilustrado en la siguiente imagen, donde los trapecios sombreados representan las torres colocadas.



Las torres 3 y 5 pueden comunicarse usando la torre 4 como intermediaria, ya que  $40 \leq 50 - 10$  y  $30 \leq 50 - 10$ . Las torres 1 y 3 pueden comunicarse usando la torre 2 como intermediaria. Las torres 1 y 5 pueden comunicarse usando la torre 3 como intermediaria. No existe forma de colocar más de 3 torres y que todas estén comunicadas entre sí, así que la función debe retornar 3.

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

Existe una sola torre en el rango, así que Pak Dengklek puede únicamente colocar una torre. Así que la función debe retornar 1.

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

Pak Dengklek puede colocar torres 1 y 3. Las torres 1 y 3 pueden comunicarse usando la torre 2 como intermediaria, ya que  $20 \leq 60 - 17$  y  $40 \leq 60 - 17$ . No existe forma de colocar más de 2 torres, así que la función debe 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 para todas las llamadas de `max_towers`.
7. (23 puntos) Sin restricciones adicionales.

## Grader de ejemplo

El grader de ejemplo 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 consulta  $j$

El grader de ejemplo 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 consulta  $j$

