



Fernsehtürme

In Jakarta befinden sich N Fernsehtürme. Alle Türme stehen in einer Reihe und sind durchnummeriert von 0 bis $N - 1$ von links nach rechts. Turm i ist $H[i]$ Meter hoch, für alle i mit $0 \leq i \leq N - 1$. Alle Höhen sind **unterschiedlich**.

Für einen positiven *Interferenzwert* δ können Türme i und j ($0 \leq i < j \leq N - 1$) genau dann miteinander kommunizieren, wenn ein Turm k dazwischen steht, sodass:

- Turm i steht links von Turm k , und Turm j steht rechts von Turm k , also $i < k < j$, sowie
- Türme i und j sind beide höchstens $H[k] - \delta$ Meter hoch.

Pak Dengklek möchte einige Fernsehtürme für sein neues Telekommunikationsnetz mieten. Deine Aufgabe ist es, Q Abfragen für Pak Dengklek zu beantworten, alle in folgender Form: Gegeben Parameter L, R und D ($0 \leq L \leq R \leq N - 1$ und $D > 0$), was ist die maximale Anzahl an Türmen, die Pak Dengklek mieten kann, unter den folgenden Bedingungen:

- Nur Türme mit Index von L bis inklusive R können gemietet werden.
- Der Interferenzwert δ ist D .
- Alle gemieteten Türme müssen paarweise miteinander kommunizieren können.

Beachte, dass zwei gemietete Türme über Turm k kommunizieren dürfen, unabhängig davon, ob Turm k gemietet ist oder nicht.

Implementierungsdetails

Implementiere folgende Funktionen:

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

- N : Die Anzahl an Fernsehtürmen.
- H : Ein Array der Länge N mit den Höhen der Türme.
- Die Funktion wird genau einmal aufgerufen, und zwar vor allen Aufrufen von `max_towers`.

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

- L, R : Nur Fernsehtürme von L bis inklusive R können gemietet werden.
- D : Der Wert von δ .

- Diese Funktion soll die maximale Anzahl an Fernsehtürmen zurückgeben, die Pak Dengklek für sein Telekommunikationsnetz mieten kann, wenn er nur Türme von L bis inklusive R mieten kann und δ den Wert D hat.
- Diese Funktion wird genau Q Mal aufgerufen.

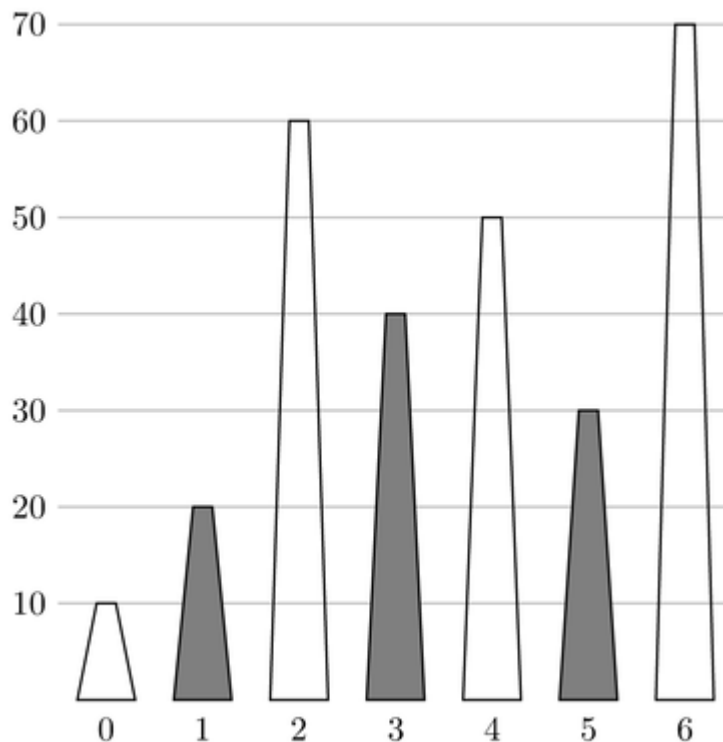
Beispiel

Betrachte die folgende Folge von Aufrufen:

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

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

Pak Dengklek kann die Türme 1, 3 und 5 mieten. Das Beispiel ist im folgenden Bild dargestellt, wobei die gemieteten Türme in dunkelgrau markiert sind.



Türme 3 und 5 können über Turm 4 kommunizieren, weil $40 \leq 50 - 10$ und $30 \leq 50 - 10$. Türme 1 und 3 können über Turm 2 kommunizieren. Türme 1 und 5 können über Turm 3 kommunizieren. Es ist nicht möglich, mehr als 3 Türme zu mieten, deshalb muss die Funktion 3 zurückgeben.

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

Es gibt nur 1 Turm im erlaubten Bereich, deshalb kann Pak Dengklek nur 1 Turm mieten. Die Funktion muss daher 1 zurückgeben.

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

Pak Dengklek kann Türme 1 und 3 mieten. Türme 1 und 3 können über 2 kommunizieren, weil $20 \leq 60 - 17$ und $40 \leq 60 - 17$. Es ist nicht möglich, mehr als 2 Türme zu mieten, deshalb muss die Funktion 2 zurückgeben.

Beschränkungen

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

Teilaufgaben

1. (4 Punkte) Es existiert ein Turm k ($0 \leq k \leq N - 1$) sodass
 - für alle i mit $0 \leq i \leq k - 1$ gilt $H[i] < H[i + 1]$, und
 - für alle i mit $k \leq i \leq N - 2$ gilt $H[i] > H[i + 1]$.
2. (11 Punkte) $Q = 1, N \leq 2000$
3. (12 Punkte) $Q = 1$
4. (14 Punkte) $D = 1$
5. (17 Punkte) $L = 0, R = N - 1$
6. (19 Punkte) Der Wert von D ist in allen Abfragen gleich.
7. (23 Punkte) Keine weiteren Beschränkungen.

Sample-Grader

Der Sample-Grader liest die Eingabe im folgenden Format:

- Zeile 1: $N Q$
- Zeile 2: $H[0] H[1] \dots H[N - 1]$
- Zeile $3 + j$ ($0 \leq j \leq Q - 1$): $L R D$ für Abfrage j

Der Sample-Grader gibt deine Antworten in folgendem Format aus:

- Zeile $1 + j$ ($0 \leq j \leq Q - 1$): der Rückgabewert von `max_towers` für Abfrage j