# Радіовежі

У Джокарті є $N$ радіовеж. Вежі розташовані на прямій лінії і пронумеровані від $0$ до $N-1$ зліва направо. Для кожного $i$ такого, що $0 \leq i \leq N-1$, висота вежі $i$ становить $H[i]$ метрів. Усі вежі мають **різні** висоти.

Для деякого додатного числа $\delta$ пара веж $i$ та $j$ (де $0 \leq i < j \leq N-1$) може спілкуватись одна з одною тоді і тільки тоді, коли між ними є вежа $k$ така, що

- вежа $i$ розташована зліва від вежі $k$, а вежа $j$ розташована справа від вежі $k$, тобто, $i < k < j$, і
- висоти веж $i$ та $j$ не більші $H[k] - \delta$ метрів кожна.

Пак Денгклек хоче взяти в оренду кілька радіовеж для своєї нової радіомережі. Ваше завдання - відповісти на $Q$ запитань Пака Денгклека такої форми: задані параметри $L, R$ та $D$ ( $0 \leq L \leq R \leq N-1$ та $D > 0$), яку максимальну кількість веж може орендувати Пак Денгклек, припускаючи, що:

- Пак Денгклек може орендувати лише вежі з індексами від $L$ до $R$ (включно), і
- значення $\delta$ дорівнює $D$, і
- усі пари радіовеж, яку Пак Денгклек орендує, повинна мати можливість спілкуватися одна з одною.

Зауважте, що дві орендовані вежі можуть спілкуватися за допомогою проміжної вежі $k$, незалежно від того, орендована вежа $k$ чи ні.

## Деталі реалізації

Ви повинні реалізувати такі процедури:

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

- $N$: кількість радіовеж.
- $H$: масив довжини $N$, що описує висоту веж.
- Ця процедура викликається рівно один раз перед будь-якими викликами `max_towers`.

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

- $L$, $R$: межі діапазону веж.

- $D$: значення $\delta$.
- Ця процедура має повернути максимальну кількість радіовеж, які Пак Денгклек може орендувати для своєї нової радіомережі, якщо йому дозволено орендувати лише вежі між вежею $L$ і вежею $R$ (включно), а значення $\delta$ рівне $D$.
- Ця процедура викликається рівно $Q$ разів.

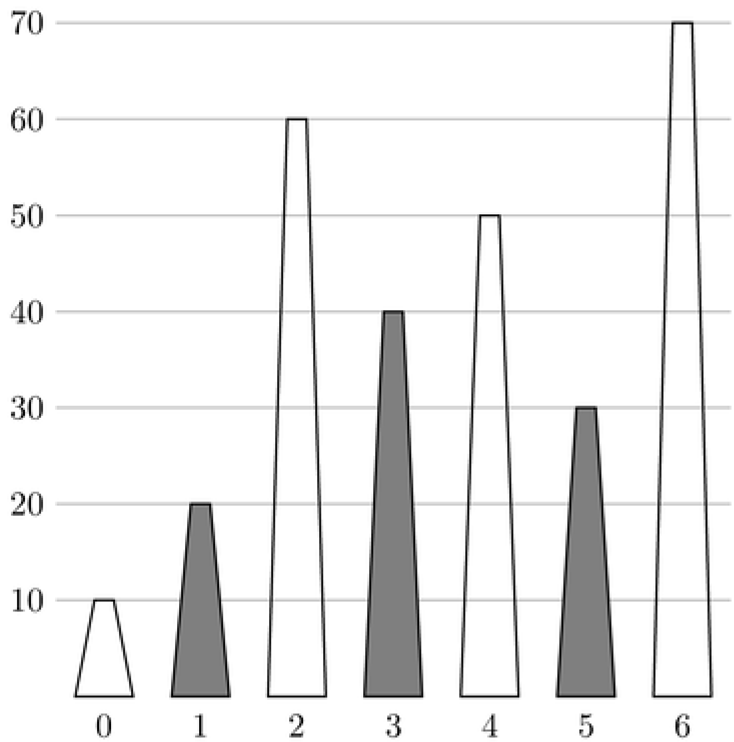
## Приклади

Розглянемо наступну послідовність викликів:

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

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

Пак Денгклек може орендувати вежі $1$, $3$, і $5$. Приклад показано на наступному малюнку, де заштриховані трапеції представляють орендовані вежі.




Вежі $3$ і $5$ можуть спілкуватись через вежу $4$ як посередника, оскільки $40 \leq 50 - 10$ і $30 \leq 50 - 10$. Вежі $1$ і $3$ можуть спілкуватися за допомогою вежі $2$ як посередника. Вежі $1$ і $5$ можуть спілкуватися за допомогою вежі $3$ як посередника. Немає способу орендувати більше ніж $3$ вежі, тому процедура має повернути $3$.

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

У діапазоні є лише вежа $1$, тому Пак Денгклек може орендувати лише вежу $1$. Тому процедура має повернути $1$.

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

Пак Денгклек може орендувати вежі $1$ і $3$. Вежі $1$ і $3$ можуть спілкуватися через вежу $2$ як посередника, оскільки $20 \leq 60 - 17$ і $40 \leq 60 - 17$. Немає способу орендувати більше ніж $2$ вежі, тому процедура має повернути $2$.

## Обмеження

- $1 \leq N \leq 100\ 000$
- $1 \leq Q \leq 100\ 000$
- $1 \leq H[i] \leq 10^9$ (для усіх $i$ таких, що $0 \leq i \leq N-1$)
- $H[i] \neq H[j]$ (для усіх $i$ і $j$ таких, що $0 \leq i < j \leq N-1$)
- $0 \leq L \leq R \leq N-1$
- $1 \leq D \leq 10^9$

## Підзадачі

1. (4 бали) Існує вежа $k$ ($0 \leq k \leq N-1$) така, що
   - для кожного $i$ такого, що $0 \leq i \leq k-1$: $H[i] < H[i+1]$, а також
   - для кожного $i$ такого, що $k \leq i \leq N-2$: $H[i] > H[i+1]$.
2. (11 балів) $Q = 1$, $N \leq 2000$
3. (12 балів) $Q = 1$
4. (14 балів) $D = 1$
5. (17 балів) $L = 0$, $R = N-1$
6. (19 балів) Значення $D$ однакове для усіх викликів `max_towers`.
7. (23 бали) Без додаткових обмежень.

## Приклад градера

Градер зчитує вхідні дані в такому форматі:

- 1-й рядок: $N\ Q$
- 2-й рядок: $H[0]\ H[1]\ \ldots\ H[N-1]$
- $(3+j)$-й рядок ($0 \leq j \leq Q-1$): $L\ R\ D$ для запитання $j$

Градер виводить ваші відповіді у такому форматі:

- $(1+j)$-й рядок ($0 \leq j \leq Q-1$): поверне значення `max_towers` для запитання $j$