

## Прыжки в джунглях

В тропических джунглях Суматры находятся  $N$  деревьев в ряд, пронумерованных от 0 до  $N - 1$  слева направо. У всех деревьев **различные высоты**, дерево с номером  $i$  имеет высоту  $H[i]$ .

Пак Денгклек тренирует орангутана прыгать с дерева на дерево. За один прыжок орангутан может перепрыгнуть с верхушки дерева на верхушку ближайшего слева или справа дерева, высота которого выше, чем высота дерева, с которого осуществляется прыжок. Формально, если орангутан находится на дереве  $x$ , он может перепрыгнуть на дерево  $y$ , если и только если выполнено одно из двух условий:

- $y$  является максимальным неотрицательным целым числом, меньшим  $x$ , таким что  $H[y] > H[x]$ ; или
- $y$  является минимальным целым числом, большим  $x$ , таким что  $H[y] > H[x]$ .

Пак Денгклек подготовил  $Q$  планов прыжков, каждый из которых задается четырьмя целыми числами  $A, B, C$  и  $D$  ( $A \leq B < C \leq D$ ). Для каждого плана Пак Денгклек хочет выяснить, можно ли начать на каком-либо дереве  $s$  ( $A \leq s \leq B$ ) и закончить на каком-либо дереве  $e$  ( $C \leq e \leq D$ ), используя прыжки по описанным правилам для перемещения между деревьями. Если это возможно, Пак Денгклек хочет узнать, какое минимальное число прыжков потребуется орангутану для этого плана.

## Детали реализации

Вам необходимо реализовать следующие функции:

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

- $N$ : число деревьев.
- $H$ : массив длины  $N$ , где  $H[i]$  - это высота дерева с номером  $i$ .
- Эта функция будет вызвана ровно один раз, до вызовов функции `minimum_jumps`.

```
int minimum_jumps(int A, int B, int C, int D)
```

- $A, B$ : задают диапазон номеров деревьев, на котором орангутан может начать прыжки.

- $C, D$ : задают диапазон номеров деревьев, на котором орангутан может закончить прыжки.
- Эта функция должна вернуть минимальное число прыжков, которое необходимо совершить орангутану, чтобы выполнить план, или  $-1$ , если выполнить план невозможно.
- Эта функция будет вызывана ровно  $Q$  раз.

## Пример

Рассмотрим следующий вызов:

```
init(7, [3, 2, 1, 6, 4, 5, 7])
```

После выполнения инициализации, рассмотрим следующий вызов:

```
minimum_jumps(4, 4, 6, 6)
```

Он означает, что орангутан должен начать прыжки на дереве 4 (высота 4) и завершить прыжки на дереве 6 (высота 7). Один способ это сделать - прыгнуть на дерево 3 (высота 6), а затем на дерево 6. Другой способ - прыгнуть на дерево 5 (высота 5), а затем на дерево 6. Таким образом, функция `minimum_jumps` должна вернуть 2.

Рассмотрим еще один возможный вызов:

```
minimum_jumps(1, 3, 5, 6)
```

Это означает, что орангутан должен начать на одном из деревьев с номерами 1 (высота 2), 2 (высота 1) или 3 (высота 6) и закончить на одном из деревьев с номерами 5 (высота 5) или 6 (высота 7). Единственный способ это сделать за минимальное число прыжков - начать с дерева 3 и за один прыжок перепрыгнуть на дерево 6. Поэтому функция `minimum_jumps` должна вернуть 1.

Рассмотрим еще один возможный вызов:

```
minimum_jumps(0, 1, 2, 2)
```

Это означает, что орангутан должен начать на дереве 0 (высота 3) или дереве 1 (высота 2) и закончить на дереве 2 (высота 1). Поскольку высота дерева 2 меньше чем у обоих возможных начальных деревьев, невозможно выполнить план. Поэтому функция `minimum_jumps` должна вернуть  $-1$ .

## Ограничения

- $2 \leq N \leq 200\,000$
- $1 \leq Q \leq 100\,000$
- $1 \leq H[i] \leq N$  (для всех  $0 \leq i \leq N - 1$ )
- $H[i] \neq H[j]$  (для всех  $0 \leq i < j \leq N - 1$ )
- $0 \leq A \leq B < C \leq D \leq N - 1$

## Подзадачи

1. (4 балла)  $H[i] = i + 1$  (для всех  $0 \leq i \leq N - 1$ )
2. (8 баллов)  $N \leq 200, Q \leq 200$
3. (13 баллов)  $N \leq 2000, Q \leq 2000$
4. (12 баллов)  $Q \leq 5$
5. (23 балла)  $A = B, C = D$
6. (21 балл)  $C = D$
7. (19 баллов) Нет дополнительных ограничений.

## Грейдер участника

Грейдер участника читает ввод в следующем формате:

- строка 1:  $N Q$
- строка 2:  $H[0] H[1] \dots H[N - 1]$
- строка  $3 + i$  ( $0 \leq i \leq Q - 1$ ):  $A B C D$  для  $i$ -го вызова функции `minimum_jumps`

Грейдер участника выводит полученные результаты в следующем формате

- строка  $1 + i$  ( $0 \leq i \leq Q - 1$ ): возвращенное значение  $i$ -го вызова `minimum_jumps`