

## 밀리 점프

수마트라의 열대 밀림에, 왼쪽부터 오른쪽으로 0부터  $N - 1$ 까지 번호가 매겨진  $N$  그루의 나무가 있다. 각각의 나무의 높이는 모두 다르다. 나무  $i$ 의 높이는  $H[i]$ 이다.

이 교수는 오랑우탄을 훈련시켜서 나무 사이를 점프해서 다니게 하고 있다. 한번 점프할 때, 오랑우탄은 현재 있는 나무의 꼭대기에서, 왼쪽 또는 오른쪽으로 현재 나무 높이보다 더 높은 가장 가까운 나무로 점프할 수 있다. 엄밀하게는, 현재 오랑우탄이 나무  $x$ 에 있다면 점프해서 이동하게 되는 나무가  $y$ 라는 것은 다음 두 조건 중 하나를 만족한다는 것과 동치이다.

- $y$ 는  $H[y] > H[x]$ 이자  $x$ 보다 작은 가장 큰 음이 아닌 정수이다. 또는
- $y$ 는  $H[y] > H[x]$ 이자  $x$ 보다 큰 가장 작은 음이 아닌 정수이다.

이 교수는 오랑우탄을 점프시킬  $Q$  가지의 계획을 가지고 있다. 각 계획은 네 정수  $A, B, C, D$  ( $A \leq B < C \leq D$ )로 표현된다. 이 교수는 오랑우탄이 어떤 나무  $s$  ( $A \leq s \leq B$ )에서 시작해서 점프를 통해서 최종적으로 나무  $e$  ( $C \leq e \leq D$ )에 도착할 수 있는 지 알고 싶다. 만약 가능하다면, 오랑우탄이 최소 횟수 점프를 해서 이 계획을 달성하게 하고 싶다.

## 상세 구현

다음 함수를 구현해야 한다.

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

- $N$ : 나무의 수
- $H$ : 길이  $N$ 인 배열로,  $H[i]$ 는 나무  $i$ 의 높이이다.
- 이 함수는 `minimum_jumps`를 호출하기 전 정확하게 한 번 호출된다.

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

- $A, B$ : 오랑우탄이 출발하는 나무의 범위
- $C, D$ : 오랑우탄이 최종 도착하는 나무의 범위
- 이 함수는 오랑우탄이 계획을 달성할 수 있는 최소 횟수의 점프를 리턴하거나, 계획을 달성하는 것이 불가능하다면  $-1$ 을 리턴해야 한다.
- 이 함수는 정확히  $Q$ 번 호출된다.

## 예제

다음 호출을 생각해보자.

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

초기화가 끝난 다음, 다음 호출을 생각해보자.

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

이는 오랑우탄이 4번 나무 (높이 4)에서 시작해서 6번 나무 (높이 7)에 최종 도착해야 한다는 뜻이다. 점프의 횟수를 최소로 하는 방법 중 하나는 먼저 3번 나무(높이 6)으로 점프한 다음, 6번 나무로 점프하는 것이다. 다른 방법은 5번 나무 (높이 5)로 점프한 다음, 6번 나무로 점프하는 것이다. 따라서, `minimum_jumps` 함수는 2를 리턴해야 한다.

다른 가능한 호출을 생각해보자.

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

이는 오랑우탄이 1번 나무 (높이 2), 2번 나무 (높이 1), 3번 나무 (높이 6) 중 하나에서 시작해서 5번 나무 (높이 5) 또는 6번 나무 (높이 7)에 최종 도착해야 한다는 뜻이다. 점프의 횟수를 최소로 하는 유일한 방법은 먼저 3번 나무에서 시작해서 6번 나무로 점프하는 것이다. 따라서, `minimum_jumps` 함수는 1를 리턴해야 한다.

다른 가능한 호출을 생각해보자.

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

이는 오랑우탄이 0번 나무 (높이 3), 1번 나무 (높이 2) 중 하나에서 시작해서 2번 나무 (높이 1)에 최종 도착해야 한다는 뜻이다. 2번 나무가 가장 높이가 낮은 나무이기 때문에, 다른 나무에서 이 나무로 점프할 수 없다. 따라서, `minimum_jumps` 함수는  $-1$ 을 리턴해야 한다.

## 제약 조건

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

## 부분 문제

1. (4 점)  $H[i] = i + 1$  (for all  $0 \leq i \leq N - 1$ )
2. (8 점)  $N \leq 200, Q \leq 200$
3. (13 점)  $N \leq 2000, Q \leq 2000$

4. (12 점)  $Q \leq 5$
5. (23 점)  $A = B, C = D$
6. (21 점)  $C = D$
7. (19 점) 추가적인 제약 조건이 없다.

## 샘플 그레이더

샘플 그레이더는 다음 양식으로 입력을 읽는다.

- line 1:  $N Q$
- line 2:  $H[0] H[1] \dots H[N - 1]$
- line  $3 + i$  ( $0 \leq i \leq Q - 1$ ): `minimum_jumps` 함수를  $i$ 번째 호출했을 때 입력  $A B C D$

샘플 그레이더는 다음 양식으로 당신의 답을 출력한다.

- line  $1 + i$  ( $0 \leq i \leq Q - 1$ ): `minimum_jumps` 함수를  $i$ 번째 호출했을 때 리턴값