



모임들

N 개의 산들이 존재하고 산들의 바닥은 수평선 위에 놓여있다. 산들에는 왼쪽에서 오른쪽으로 0부터 $N - 1$ 의 번호가 붙어있다. 산 i 의 높이는 H_i 이다 ($0 \leq i \leq N - 1$). 각 산의 꼭대기에 정확히 한 사람이 산다.

당신은 Q 개의 모임들을 개최할 것이다. 여기서 모임들은 0부터 $Q - 1$ 로 나타낸다. 모임 j ($0 \leq j \leq Q - 1$)에는 산 번호 L_j 이상 R_j 이하의 산들에 살고 있는 모든 사람들이 참석해야 한다 ($0 \leq L_j \leq R_j \leq N - 1$).

모임 j 에 대해서, 당신은 모임 장소로 산 x 를 선택해야 한다 ($L_j \leq x \leq R_j$). 이 선택에 따라 모임 j 의 비용이 다음과 같이 계산된다:

- 산 y ($L_j \leq y \leq R_j$)에 사는 참석자의 비용은 x 와 y 를 포함해서 x 와 y 사이 산들의 최대 높이이다. 특별히 산 x 에 사는 참석자의 비용은 산 x 의 높이인 H_x 이다.
- 모임의 비용은 모든 참석자들의 비용의 합이다.

각 모임에 대해서, 당신은 그 모임을 개최하는 최소 비용을 찾고 싶다.

각 모임 후에 모든 참석자들은 그들이 사는 산으로 돌아가므로 모임의 비용은 이전 모임들에 영향받지 않음에 주목하자.

Implementation details

다음 함수를 구현해야 한다:

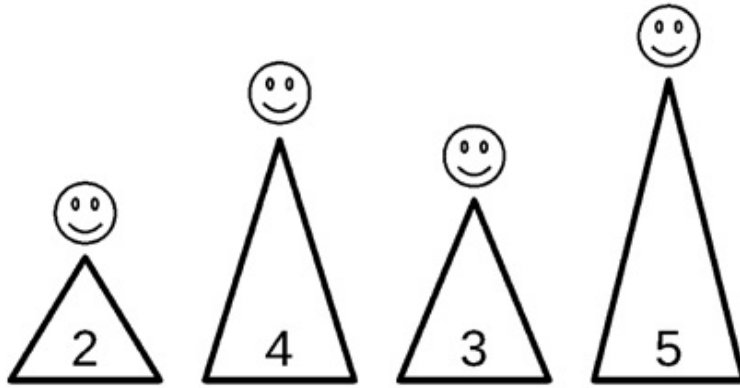
```
int64[] minimum_costs(int[] H, int[] L, int[] R)
```

- H : 산들의 높이를 나타내는 길이 N 의 배열.
- L, R : 모임 참석자들의 범위를 나타내는 길이 Q 의 배열.
- 이 함수는 길이 Q 의 배열 C 를 반환해야 한다. C_j ($0 \leq j \leq Q - 1$)의 값은 모임 j 를 개최하는 가능한 최소 비용이어야 한다.
- N 과 Q 의 값은 배열의 길이이고 구현 명세에서 알려진 방법을 통해서 얻을 수 있다는데 주의하자.

Example

$N = 4, H = [2, 4, 3, 5], Q = 2, L = [0, 1], R = [2, 3]$ 이라고 하자.

그레이더는 `minimum_costs([2, 4, 3, 5], [0, 1], [2, 3])` 을 호출한다.



모임 $j = 0$ 은 $L_j = 0$ 이고 $R_j = 2$ 이다. 따라서 산 0, 1, 2에 사는 사람들이 이 모임에 참석할 것이다. 산 0이 모임 장소로 선택되면 모임 0의 비용은 다음과 같이 계산된다:

- 산 0에 사는 참석자의 비용은 $\max\{H_0\} = 2$ 이다.
- 산 1에 사는 참석자의 비용은 $\max\{H_0, H_1\} = 4$ 이다.
- 산 2에 사는 참석자의 비용은 $\max\{H_0, H_1, H_2\} = 4$ 이다.
- 따라서 모임 0의 비용은 $2 + 4 + 4 = 10$ 이다.

더 작은 비용으로 모임 0을 개최하는 것은 불가능하므로 모임 0을 개최하는 최소 비용은 10이다.

모임 $j = 1$ 은 $L_j = 1$ 이고 $R_j = 3$ 이다. 따라서 산 1, 2, 3에 사는 사람들이 이 모임에 참석할 것이다. 산 2가 모임 장소로 선택되면 모임 1의 비용은 다음과 같이 계산된다:

- 산 1에 사는 참석자의 비용은 $\max\{H_1, H_2\} = 4$ 이다.
- 산 2에 사는 참석자의 비용은 $\max\{H_2\} = 3$ 이다.
- 산 3에 사는 참석자의 비용은 $\max\{H_2, H_3\} = 5$ 이다.
- 따라서 모임 1의 비용은 $4 + 3 + 5 = 12$ 이다.

더 작은 비용으로 모임 1을 개최하는 것은 불가능하므로 모임 1을 개최하는 최소 비용은 12이다.

압축된 첨부 패키지 파일의 `sample-01-in.txt`와 `sample-01-out.txt`는 이 예제에 대응한다. 다른 입출력 예제도 이 패키지에 포함되어 있다.

Constraints

- $1 \leq N \leq 750\,000$
- $1 \leq Q \leq 750\,000$
- $1 \leq H_i \leq 1\,000\,000\,000$ ($0 \leq i \leq N - 1$)
- $0 \leq L_j \leq R_j \leq N - 1$ ($0 \leq j \leq Q - 1$)
- $(L_j, R_j) \neq (L_k, R_k)$ ($0 \leq j < k \leq Q - 1$)

Subtasks

1. (4 points) $N \leq 3\,000$, $Q \leq 10$
2. (15 points) $N \leq 5\,000$, $Q \leq 5\,000$

3. (17 points) $N \leq 100\,000$, $Q \leq 100\,000$, $H_i \leq 2$ ($0 \leq i \leq N - 1$)
4. (24 points) $N \leq 100\,000$, $Q \leq 100\,000$, $H_i \leq 20$ ($0 \leq i \leq N - 1$)
5. (40 points) 추가적인 제약조건 없음

Sample grader

샘플 그레이더는 다음과 같은 형식으로 입력을 받는다.

- line 1: N Q
- line 2: H_0 H_1 \cdots H_{N-1}
- line $3 + j$ ($0 \leq j \leq Q - 1$): L_j R_j

샘플 그레이더는 다음과 같은 형식으로 `minimum_costs` 의 반환값을 출력한다.

- line $1 + j$ ($0 \leq j \leq Q - 1$): C_j