Meetings

There are N mountains lying in a horizontal row, numbered from 0 through $N-1$ from left to right. The height of the mountain i is H_i ($0 \leq i \leq N-1$). Exactly one person lives on the top of each mountain.

You are going to hold Q meetings, numbered from 0 through $Q-1$. The meeting j ($0 \leq j \leq Q-1$) will be attended by all the people living on the mountains from L_j to R_j, inclusive ($0 \leq L_j \leq R_j \leq N-1$). For this meeting, you must select a mountain x as the meeting place ($L_j \leq x \leq R_j$). The cost of this meeting, based on your selection, is then calculated as follows:

- The cost of the participant from each mountain y ($L_j \leq y \leq R_j$) is the maximum height of the mountains between the mountains x and y, inclusive. In particular, the cost of the participant from the mountain x is H_x, the height of the mountain x.
- The cost of the meeting is the sum of the costs of all participants.

For each meeting, you want to find the minimum possible cost of holding it.

Note that all participants go back to their own mountains after each meeting; so the cost of a meeting is not influenced by the previous meetings.

Implementation details

You should implement the following function:

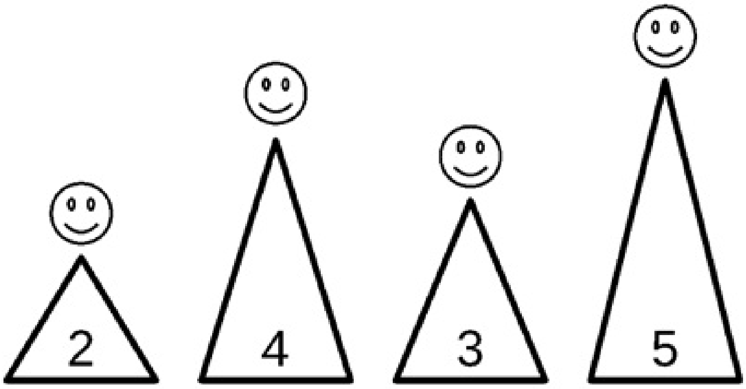
```
int64[] minimum_costs(int[] H, int[] L, int[] R)
```

- `H`: an array of length N, representing the heights of the mountains.
- `L` and `R`: arrays of length Q, representing the range of the participants in the meetings.
- This function should return an array C of length Q. The value of C_j ($0 \leq j \leq Q-1$) must be the minimum possible cost of holding the meeting j.
- Note that the values of N and Q are the lengths of the arrays, and can be obtained as indicated in the implementation notice.

Example

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

The grader calls `minimum_costs([2, 4, 3, 5], [0, 1], [2, 3])`.



The meeting $j = 0$ has $L_j = 0$ and $R_j = 2$, so will be attended by the people living on the mountains 0, 1, and 2. If the mountain 0 is chosen as the meeting place, the cost of the meeting 0 is calculated as follows:

- The cost of the participant from the mountain 0 is $\max\{H_0\} = 2$.
- The cost of the participant from the mountain 1 is $\max\{H_0, H_1\} = 4$.
- The cost of the participant from the mountain 2 is $\max\{H_0, H_1, H_2\} = 4$.
- Therefore, the cost of the meeting 0 is $2 + 4 + 4 = 10$.

It is impossible to hold the meeting 0 at a lower cost, so the minimum cost of the meeting 0 is 10.

The meeting $j = 1$ has $L_j = 1$ and $R_j = 3$, so will be attended by the people living on the mountains 1, 2, and 3. If the mountain 2 is chosen as the meeting place, the cost of the meeting 1 is calculated as follows:

- The cost of the participant from the mountain 1 is $\max\{H_1, H_2\} = 4$.
- The cost of the participant from the mountain 2 is $\max\{H_2\} = 3$.
- The cost of the participant from the mountain 3 is $\max\{H_2, H_3\} = 5$.
- Therefore, the cost of the meeting 1 is $4 + 3 + 5 = 12$.

It is impossible to hold the meeting 1 at a lower cost, so the minimum cost of the meeting 1 is 12.

The files `sample-01-in.txt` and `sample-01-out.txt` in the zipped attachment package correspond to this example. Other sample inputs/outputs are also available in the package.

Constraints

- $1 \leq N \leq 750\,000$
- $1 \leq Q \leq 750\,000$

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

Subtasks

1. (4 points) $N \le 3\,000$, $Q \le 10$
2. (15 points) $N \le 5\,000$, $Q \le 5\,000$
3. (17 points) $N \le 100\,000$, $Q \le 100\,000$, $H_i \le 2$ ($0 \le i \le N-1$)
4. (24 points) $N \le 100\,000$, $Q \le 100\,000$, $H_i \le 20$ ($0 \le i \le N-1$)
5. (40 points) No additional constraints

Sample grader

The sample grader reads the input in the following format:

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

The sample grader prints the return value of `minimum_costs` in the following format:

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