Meetings

Existem N montanhas ao longo de uma linha horizontal, numeradas de 0 a $N-1$, da esquerda para a direita. A altura da montanha i é H_i ($0 \leq i \leq N-1$). Exatamente uma pessoa vive no topo de cada montanha.

Você vai organizar Q reuniões, numeradas de 0 a $Q-1$. A reunião j ($0 \leq j \leq Q-1$) terá a participação de todas as pessoas vivendo nas montanhas de L_j a R_j, inclusive ($0 \leq L_j \leq R_j \leq N-1$). Para esta reunião, você precisa escolher a montanha x como o local da reunião ($L_j \leq x \leq R_j$). Esta reunião tem um custo, que é calculado da seguinte forma:

- O custo da reunião é a soma dos custos de todos os participantes.
- O custo do participante da montanha y ($L_j \leq y \leq R_j$) é a altura máxima das montanhas entre a montanha x e y, inclusive.
- Em particular, o custo do participante da montanha x é H_x, a altura da montanha x.

Para cada reunião, você quer encontrar o custo mínimo possível para a organizar.

Note que todos os participantes voltam para suas próprias montanhas depois de cada reunião; portanto, o custo de uma reunião não é influenciado pelas reuniões anteriores.

Detalhes de Implementação

Você deve implementar a seguinte função:

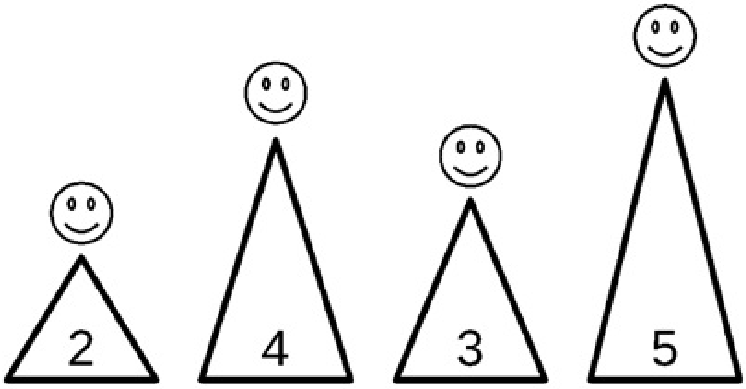
```
int64[] minimum_costs(int[] H, int[] L, int[] R)
```

- `H`: um array de comprimento N, representando as alturas das montanhas.
- `L` e `R`: arrays de comprimento Q, representando o intervalo dos participantes nas reuniões.
- Esta função deve retornar um array C de comprimento Q. O valor de C_j ($0 \leq j \leq Q-1$) deve ser o custo mínimo possível para organizar a reunião j.
- Note que os valores de N e Q são os comprimentos dos arrays e podem ser obtidos como indicado na nota de implementação.

Exemplo

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

O avaliador chama `minimum_costs([2, 4, 3, 5], [0, 1], [2, 3])`.



A reunião $j = 0$ tem $L_j = 0$ e $R_j = 2$, portanto terá a participação das pessoas vivendo nas montanhas 0, 1 e 2. Se a montanha 0 é escolhida como o local da reunião, o custo da reunião 0 será calculado da seguinte forma:

- O custo do participante da montanha 0 é $\max\{H_0\} = 2$.
- O custo do participante da montanha 1 é $\max\{H_0, H_1\} = 4$.
- O custo do participante da montanha 2 é $\max\{H_0, H_1, H_2\} = 4$.
- Portanto, o custo da reunião 0 é $2 + 4 + 4 = 10$.

É impossível organizar a reunião 0 com um custo menor, assim o custo mínimo da reunião 0 é 10.

A reunião $j = 1$ tem $L_j = 1$ e $R_j = 3$, portanto terá a participação das pessoas vivendo nas montanhas 1, 2, e 3. Se a montanha 2 é escolhida como o local da reunião, o custo da reunião 1 será calculado da seguinte forma:

- O custo do participante da montanha 1 é $\max\{H_1, H_2\} = 4$.
- O custo do participante da montanha 2 é $\max\{H_2\} = 3$.
- O custo do participante da montanha 3 é $\max\{H_2, H_3\} = 5$.
- Portanto, o custo da reunião 1 é $4 + 3 + 5 = 12$.

É impossível organizar a reunião 1 com um custo menor, assim o custo mínimo da reunião 1 é 12.

Os ficheiros `sample-01-in.txt` e `sample-01-out.txt` dentro do arquivo zip correspondem a este exemplo. Outros inputs/outputs de exemplo também estão disponíveis no arquivo.

Restrições

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

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

Subtarefas

1. (4 pontos) $N \leq 3\,000$, $Q \leq 10$
2. (15 pontos) $N \leq 5\,000$, $Q \leq 5\,000$
3. (17 pontos) $N \leq 100\,000$, $Q \leq 100\,000$, $H_i \leq 2$ ($0 \leq i \leq N-1$)
4. (24 pontos) $N \leq 100\,000$, $Q \leq 100\,000$, $H_i \leq 20$ ($0 \leq i \leq N-1$)
5. (40 pontos) Nenhuma restrição adicional

Avaliador de exemplo

O avaliador de exemplo lê o inpu no seguinte formato:

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

O avaliador de exemplo imprime o valor de retorno de `minimum_costs` no seguinte formato:

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