Treffen

N Berge, nummeriert von links nach rechts von 0 bis $N-1$, liegen in einer Reihe nebeneinander. Die Höhe des i-ten Berges ist $H_i (0 \leq i \leq N-1)$. Genau eine Person lebt auf dem Gipfel jedes Berges.

Du sollst Q Treffen abhalten, nummeriert von 0 bis $Q-1$. Das Treffen j $(0 \leq j \leq Q-1)$ wird von allen Personen besucht, die auf den Bergen von L_j bis inklusive R_j wohnen $(0 \leq L_j \leq R_j \leq N-1)$. Du sollst für dieses Treffen einen Berg x als Treffpunkt auswählen $(L_j \leq x \leq R_j)$. Die Kosten eines hängen von der Wahl von x ab und werden wie folgt berechnet:

- Die Kosten für einen Teilnehmer vom Berg y $(L_j \leq y \leq R_j)$ sind die maximale Höhe aller Berge zwischen den Bergen x und y inklusive.
- Insbesondere sind die Kosten des Teilnehmers vom Berg x daher H_x, die Höhe des Berges x.
- Die Kosten des Treffens sind die Summe der Kosten aller Teilnehmer.

Für jedes Treffen sollst du die kleinstmöglichen Kosten bestimmen.

Beachte: Alle Teilnehmer gehen nach jedem Treffen zu ihren eigenen Bergen zurück; daher werden die Kosten eines Treffens nicht von vorherigen Treffen beeinflusst.

Implementierungshinweise

Du musst die folgende Funktion implementieren:

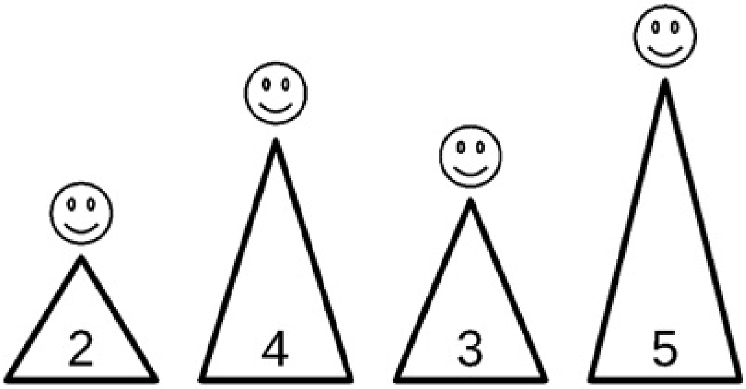
```
int64[] minimum_costs(int[] H, int[] L, int[] R)
```

- `H`: ein Array der Länge N, das die Höhen der Berge beschreibt.
- `L` und `R`: zwei Arrays der Länge Q, welche die Bereiche der Treffen-Teilnehmer beschreiben.
- Diese Funktion soll ein Array C der Länge Q zurückgeben: C_j $(0 \leq j \leq Q-1)$ muss die minimal möglichen Kosten angeben, um Treffen j abzuhalten.
- Beachte, dass N und Q die Längen der Arrays angeben. Diese können wie in den Implementierungshinweisen beschrieben abgefragt werden.

Beispiele

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

Der Grader ruft `minimum_costs([2, 4, 3, 5], [0, 1], [2, 3])` auf.



Für Treffen $j = 0$ ist $L_j = 0$ und $R_j = 2$. Somit wird dieses Treffen von den Teilnehmern in den Bergen Nummer 0, 1 und 2 besucht. Wenn Berg 0 als Treffpunkt ausgewählt wird, berechnen sich die Kosten wie folgt:

- Die Kosten für den Teilnehmer vom Berg 0 betragen $\max\{H_0\} = 2$.
- Die Kosten für den Teilnehmer vom Berg 1 betragen $\max\{H_0, H_1\} = 4$.
- Die Kosten für den Teilnehmer vom Berg 2 betragen $\max\{H_0, H_1, H_2\} = 4$.
- Daher betragen die Gesamtkosten für Treffen 0: $2 + 4 + 4 = 10$.

Es ist nicht möglich Treffen 0 zu niedrigeren Kosten als 10 abzuhalten.

Für Treffen $j = 1$ ist $L_j = 1$ und $R_j = 3$. Somit wird dieses Treffen von den Teilnehmern in den Bergen Nummer 1, 2 und 3 besucht. Wenn Berg 2 als Treffpunkt ausgewählt wird, berechnen sich die Kosten wie folgt:

- Die Kosten für den Teilnehmer vom Berg 1 betragen $\max\{H_1, H_2\} = 4$.
- Die Kosten für den Teilnehmer vom Berg 2 betragen $\max\{H_2\} = 3$.
- Die Kosten für den Teilnehmer vom Berg 3 betragen $\max\{H_2, H_3\} = 5$.
- Daher betragen die Gesamtkosten für Treffen 1: $4 + 3 + 5 = 12$.

Es ist nicht möglich Treffen 1 zu niedrigeren Kosten als 12 abzuhalten.

Die Dateien `sample-01-in.txt` und `sample-01-out.txt` im Zip-Archiv entsprechen diesen Beispielen. Andere Beispieleingaben und -ausgaben liegen ebenfalls in diesem Archiv.

Limits

- $1 \leq N \leq 750\,000$
- $1 \leq Q \leq 750\,000$
- $1 \leq H_i \leq 1\,000\,000\,000$ ($0 \leq i \leq N - 1$)
- $0 \leq L_j \leq R_j \leq N - 1$ ($0 \leq j \leq Q - 1$)

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

Teilaufgaben

1. (4 Punkte) $N \leq 3\,000$, $Q \leq 10$
2. (15 Punkte) $N \leq 5\,000$, $Q \leq 5\,000$
3. (17 Punkte) $N \leq 100\,000$, $Q \leq 100\,000$, $H_i \leq 2$ $(0 \leq i \leq N - 1)$
4. (24 Punkte) $N \leq 100\,000$, $Q \leq 100\,000$, $H_i \leq 20$ $(0 \leq i \leq N - 1)$
5. (40 Punkte) Keine weiteren Einschränkungen

Beispielgrader

Der Beispielgrader liest die Eingabe in folgendem Format ein:

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

Der Beispielgrader gibt den Rückgabewert von `minimum_costs` in folgendem Format aus:

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