



Consessi eremitici

consesso s. m. [dal lat. *consessus -us*, der. di *considerēre* «star seduto»]. – Adunanza di persone ragguardevoli: *sono lieto di prendere parte a questo c. di dotti, a questo nobile c.*;

Nella prefettura di Ibaraki ci sono N montagne in fila, numerate da 0 a $N - 1$ da sinistra a destra, ciascuna di altezza H_i (per $0 \leq i \leq N - 1$). In cima ad ogni montagna vive esattamente un eremita.

Devi organizzare Q consessi eremitici (numerati da 0 a $Q - 1$), tali per cui il consesso j ($0 \leq j \leq Q - 1$) deve riunire tutti gli eremiti che vivono nelle montagne da L_j a R_j incluse ($0 \leq L_j \leq R_j \leq N - 1$). Per ognuno di quei consessi, devi selezionare una montagna x in cui tenerlo (con $L_j \leq x \leq R_j$) in maniera da minimizzare la **fatica** totale del consesso, che si calcola come:

- La fatica totale di un consesso è la somma delle fatiche che ogni suo partecipante spende per raggiungere il luogo del consesso.
- La fatica spesa dall'eremita nella montagna y per raggiungere la montagna x è pari alla massima altezza tra le montagne comprese tra x e y (incluse).
- In particolare, la fatica spesa dall'eremita x che ospita il consesso è pari ad H_x , l'altezza della sua montagna.

Nota bene che ogni partecipante torna indietro alla sua montagna dopo ogni consesso, per cui la fatica totale di ciascuno di questi non è mai influenzata dagli altri.

Dettagli di implementazione

Devi implementare la seguente funzione:

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

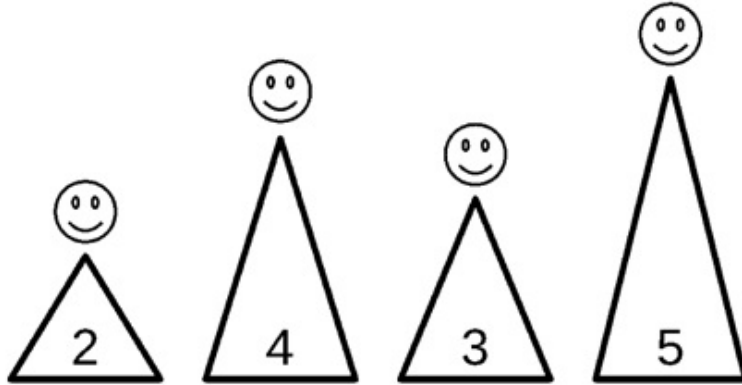
- H : array di lunghezza N , rappresenta le altezze delle montagne.
- L ed R : array di lunghezza Q , rappresentano gli estremi dell'intervallo di eremiti partecipanti a ciascun consesso.
- La funzione deve restituire un array C di lunghezza Q , per cui C_j ($0 \leq j \leq Q - 1$) sia la fatica minima possibile per tenere il consesso j .

Nota che le lunghezze N e Q non vengono fornite come parametri perché si possono ottenere (come indicato nelle "Note di implementazione") dagli array stessi.

Esempio

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

Poniamo che il grader chiami `minimum_costs([2, 4, 3, 5], [0, 1], [2, 3])`.



Il consesso $j = 0$ ha $L_j = 0$ e $R_j = 2$, e quindi sarà presenziato dagli eremiti che vivono nelle montagne 0, 1, e 2. Se la montagna 0 è scelta come luogo d'incontro, la fatica totale del consesso 0 è calcolata come segue:

- La fatica dell'eremita della montagna 0 è $\max\{H_0\} = 2$.
- La fatica dell'eremita della montagna 1 è $\max\{H_0, H_1\} = 4$.
- La fatica dell'eremita della montagna 2 è $\max\{H_0, H_1, H_2\} = 4$.
- Quindi, la fatica totale del consesso 0 è $2 + 4 + 4 = 10$.

Non è possibile tenere il consesso 0 con minore fatica totale, e quindi la minima fatica del consesso 0 è 10.

Il consesso $j = 1$ ha $L_j = 1$ e $R_j = 3$, e quindi sarà presenziato dagli eremiti che vivono nelle montagne 1, 2, e 3. Se la montagna 2 è scelta come luogo d'incontro, la fatica totale del consesso 1 è calcolata come segue:

- La fatica dell'eremita della montagna 1 è $\max\{H_1, H_2\} = 4$.
- La fatica dell'eremita della montagna 2 è $\max\{H_2\} = 3$.
- La fatica dell'eremita della montagna 3 è $\max\{H_2, H_3\} = 5$.
- Quindi, la fatica totale del consesso 1 è $4 + 3 + 5 = 12$.

Non è possibile tenere il consesso 1 con minore fatica totale, e quindi la minima fatica del consesso 1 è 12.

I file `sample-01-in.txt` e `sample-01-out.txt` nell'archivio compresso in allegato corrispondono a questo esempio. Altri input/output di esempio sono inoltre disponibili in questo archivio.

Assunzioni

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

Subtask

1. (4 punti) $N \leq 3\,000$, $Q \leq 10$
2. (15 punti) $N \leq 5\,000$, $Q \leq 5\,000$
3. (17 punti) $N \leq 100\,000$, $Q \leq 100\,000$, $H_i \leq 2$ ($0 \leq i \leq N - 1$)
4. (24 punti) $N \leq 100\,000$, $Q \leq 100\,000$, $H_i \leq 20$ ($0 \leq i \leq N - 1$)
5. (40 punti) Nessuna limitazione aggiuntiva

Grader di esempio

Il grader di esempio legge l'input nel formato seguente:

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

Il grader di esempio stampa il valore restituito da `minimum_costs` nel seguente formato:

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