

Giardino di Budrio (piante)

La provincia di Forlì-Cesena ha deciso di incentivare il rinnovamento del verde indicando un concorso per il miglior giardino del territorio.

Per questo Matteo ha comprato un giardino a Budrio e ora vuole renderlo il più rigoglioso di tutta la Romagna.

Il giardino di Matteo è composto da N piante in fila. L' i -esima pianta si trova in un vaso che dista H_i metri dal suolo.



Figura 1: Il nuovo giardino di Matteo

Purtroppo però, prendersi cura delle piante è un lavoro faticoso: infatti bisogna innaffiarle ogni giorno! Per semplificare il lavoro, Matteo ha collegato ogni coppia di vasi adiacenti con un canale che l'acqua può attraversare solo dal vaso più alto a quello basso. Se un canale è orizzontale (cioè collega due vasi alla stessa altezza) l'acqua ristagna e non riesce ad attraversarlo.

Così facendo, innaffiando un vaso direttamente, l'acqua può diffondersi anche ad altri vasi.

Matteo è pigro, aiutalo a determinare qual è il numero minimo di vasi che deve innaffiare direttamente per far sì che l'acqua raggiunga tutti i vasi.

Implementazione

Dovrai sottoporre un unico file, con estensione `.cpp` o `.py`.

📎 Tra gli allegati a questo task troverai dei template `piante.cpp` e `piante.py` con un esempio di implementazione.

Dovrai implementare la seguente funzione:

C++	<code>int pollice_verde(int N, vector<int> H);</code>
Python	<code>def pollice_verde(N: int, H: List[int]) -> int:</code>

- L'intero N rappresenta il numero di piante.
- L'array H , indicizzato da 0 a $N - 1$, contiene l'altezza a cui si trovano le piante.

- La funzione deve restituire il minimo numero di piante in cui versare dell'acqua per innaffiarle tutte.

Grader di prova

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che potete usare per testare le vostre soluzioni in locale. Il grader di esempio legge i dati da `stdin`, chiama la funzione che devi implementare e scrive su `stdout`, secondo il seguente formato.

Il file di input è composto da 2 righe, contenenti:

- Riga 1: l'intero N .
- Riga 2: gli interi H_0, \dots, H_{N-1} .

Il file di output è composto da un'unica riga, contenente il valore restituito dalla funzione `pollice_verde`.

Assunzioni

- $1 \leq N \leq 100\,000$.
- $1 \leq H_i \leq 10^9$ per ogni $0 \leq i < N$.

Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test che lo compongono.

- **Subtask 1** [0 punti]: Casi d'esempio.
- **Subtask 2** [3 punti]: $H_i < H_{i+1}$ per ogni $0 \leq i < N - 1$.
- **Subtask 3** [15 punti]: $N \leq 20$.
- **Subtask 4** [24 punti]: $N \leq 3000$.
- **Subtask 5** [58 punti]: Nessuna limitazione aggiuntiva.

Esempi di input/output

stdin	stdout
5 4 8 7 2 1	1
10 21 18 14 13 13 11 10 10 10 12	4

Spiegazione

Nel **primo caso d'esempio** la soluzione ottimale è quella di innaffiare direttamente solo la pianta 1, dalla quale l'acqua scorre attraverso i canali a tutte le altre piante. E' possibile dimostrare che non è possibile bagnare tutte le piante innaffiandone direttamente meno di una, e di conseguenza la risposta è 1.