



Allenamento su ChinaForces (allenamento)

Come sanno tutti, per ottenere buoni risultati alle gare di informatica Filippo si allena sul sito segreto *ChinaForces*. Filippo ha già partecipato a N contest, e nell' i -esimo di questi ha ottenuto una performance A_i . Le performance sono tutte distinte.



Figura 1: Il logo di *ChinaForces*

Filippo vuole analizzare le sue performance nei contest, in particolare il suo miglioramento nel tempo. Filippo decide allora di definire un intervallo di contest A_l, A_{l+1}, \dots, A_r **migliorante** se e solo se verifica la seguente condizione:

- Esistono due indici x, y ($l \leq x < y \leq r$) tali che $A_x = \min(A_l, A_{l+1}, \dots, A_r)$ e $A_y = \max(A_l, A_{l+1}, \dots, A_r)$.

Aiuta Filippo a determinare quanti sono gli intervalli miglioranti.

Implementazione

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

📖 Tra gli allegati a questo task troverai un template `allenamento.cpp` con un esempio di implementazione.

Dovrai implementare la seguente funzione:

```
C++ | long long conta(int N, vector<int> A);
```

La funzione viene chiamata durante l'esecuzione del programma con i seguenti parametri:

- L'intero N rappresenta il numero di contest a cui Filippo ha partecipato.
- L'array A , indicizzato da 0 a $N - 1$, descrive le performance nei vari contest. In particolare, per ogni $0 \leq i < N$, A_i indica la performance nell' i -esimo contest.

La funzione `conta` deve restituire il numero di intervalli miglioranti di A .

Grader di prova

Nella directory relativa a questo problema è presente una versione semplificata del grader usato durante la correzione, che puoi usare per testare le tue 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.

L'input è composto da 2 righe, contenenti:

- Riga 1: l'intero N .
- Riga 2: N interi A_0, A_1, \dots, A_{N-1} .

L'output è composto da un'unica riga:

- Riga 1: il valore restituito dalla funzione `conta`.

Assunzioni

- $1 \leq N \leq 10^7$.
- $1 \leq A_i \leq 10^9$.
- Gli A_i sono distinti.

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 relativi ad esso.

- **Subtask 1** [0 punti]: Casi d'esempio.
- **Subtask 2** [12 punti]: $N \leq 20$
- **Subtask 3** [17 punti]: $N \leq 400$
- **Subtask 4** [12 punti]: $N \leq 1500$
- **Subtask 5** [17 punti]: $N \leq 5000$
- **Subtask 6** [40 punti]: $N \leq 2 \cdot 10^5$.
- **Subtask 7** [2 punti]: Nessuna limitazione specifica.

Esempi di input/output

stdin	stdout
6 5 2 1 4 7 3	9
8 3 27 86 95 419 852 1473 6461	28

Spiegazioni

Nel **primo caso d'esempio** gli intervalli miglioranti sono $[1, 5]$, $[1, 6]$, $[2, 4]$, $[2, 5]$, $[2, 6]$, $[3, 4]$, $[3, 5]$, $[3, 6]$, $[4, 5]$.
Ad esempio:

- $[4, 5]$ è migliorante perché $\min(A_4, A_5) = A_4$, $\max(A_4, A_5) = A_5$, quindi $x = 4$, $y = 5$: dunque vale $x < y$.
- $[2, 5]$ è migliorante perché $\min(A_2, A_3, A_4, A_5) = A_3$, $\max(A_2, A_3, A_4, A_5) = A_5$, quindi $x = 3$, $y = 5$: dunque vale $x < y$.

Nel **secondo caso d'esempio** tutti gli intervalli di lunghezza almeno 2 sono miglioranti.