



# Rettangoli

A metà del 18-esimo secolo il sovrano Panah-Ali Khan ordinò di costruire una fortezza sull'altopiano intorno alla città di Shusha. L'altopiano è una griglia  $n \times m$  di celle quadrate, per cui la cella nella riga  $i$  e colonna  $j$  (con  $0 \leq i \leq n - 1, 0 \leq j \leq m - 1$ ) viene indicata come  $C(i, j)$ . Ogni cella  $C(i, j)$  ha un'altezza specifica  $a[i][j]$ .

Panah-Ali Khan ha chiesto ai suoi architetti di scegliere un'area **rettangolare** in cui costruire la fortezza. Quest'area consiste delle celle  $C(i, j)$  con  $r_1 \leq i \leq r_2$  e  $c_1 \leq j \leq c_2$ , e non deve contenere nessuna cella dei bordi della griglia, per cui  $1 \leq r_1 \leq r_2 \leq n - 2$  e  $1 \leq c_1 \leq c_2 \leq m - 2$ .

Inoltre, per essere considerata **valida**, un'area deve rispettare la seguente condizione per ogni sua cella  $C(i, j)$ :

- Considera le due celle prima e dopo l'area selezionata sulla riga  $i$  (celle  $C(i, c_1 - 1)$  e  $C(i, c_2 + 1)$ ) e sulla colonna  $j$  (celle  $C(r_1 - 1, j)$  e  $C(r_2 + 1, j)$ ). L'altezza della cella  $C(i, j)$  deve essere **strettamente minore** delle altezze di tutte e quattro queste celle.

Il tuo compito è di aiutare gli architetti a trovare il numero di aree valide per la fortezza, cioè il numero di diverse scelte per  $(r_1, r_2, c_1, c_2)$  che definiscono un'area valida.

## Dettagli di implementazione

Devi implementare la seguente funzione:

```
int64 count_rectangles(int[][] a)
```

- $a$ : un array bidimensionale  $n \times m$  di interi che rappresentano le altezze delle celle.
- La funzione deve restituire il numero di aree valide per la costruzione della fortezza.

# Esempi

## Esempio 1

Considera la chiamata seguente.

```
count_rectangles([[4, 8, 7, 5, 6],
                  [7, 4, 10, 3, 5],
                  [9, 7, 20, 14, 2],
                  [9, 14, 7, 3, 6],
                  [5, 7, 5, 2, 7],
                  [4, 5, 13, 5, 6]])
```

4	8	7	5	6
7	4	10	3	5
9	7	20	14	2
9	14	7	3	6
5	7	5	2	7
4	5	13	5	6

There are 6 valid areas, listed below:

- $r_1 = r_2 = c_1 = c_2 = 1$
- $r_1 = 1, r_2 = 2, c_1 = c_2 = 1$
- $r_1 = r_2 = 1, c_1 = c_2 = 3$
- $r_1 = r_2 = 4, c_1 = 2, c_2 = 3$
- $r_1 = r_2 = 4, c_1 = c_2 = 3$
- $r_1 = 3, r_2 = 4, c_1 = c_2 = 3$

Per esempio  $r_1 = 1, r_2 = 2, c_1 = c_2 = 1$  è un'area valida perché valgono entrambe le seguenti condizioni:

- $a[1][1] = 4$  è strettamente minore di  $a[0][1] = 8$ ,  $a[3][1] = 14$ ,  $a[1][0] = 7$ , e  $a[1][2] = 10$ .
- $a[2][1] = 7$  è strettamente minore di  $a[0][1] = 8$ ,  $a[3][1] = 14$ ,  $a[2][0] = 9$ , e  $a[2][2] = 20$ .

## Assunzioni

- $1 \leq n, m \leq 2500$ .
- $0 \leq a[i][j] \leq 7\,000\,000$  (per ogni  $0 \leq i \leq n - 1, 0 \leq j \leq m - 1$ ).

## Subtask

1. (8 punti)  $n, m \leq 30$ .
2. (7 punti)  $n, m \leq 80$ .
3. (12 punti)  $n, m \leq 200$ .
4. (22 punti)  $n, m \leq 700$ .
5. (10 punti)  $n \leq 3$ .
6. (13 punti)  $0 \leq a[i][j] \leq 1$  (per ogni  $0 \leq i \leq n - 1, 0 \leq j \leq m - 1$ ).
7. (28 punti) Nessuna limitazione aggiuntiva.

## Grader di esempio

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

- riga 1:  $n m$
- righe  $2 + i$  (per  $0 \leq i \leq n - 1$ ):  $a[i][0] a[i][1] \dots a[i][m - 1]$

Il grader di esempio stampa un'unica riga contenente il valore restituito da `count_rectangles`.