Тик бурчтуктар

Тоолуу жер $n \times m$ өлчөмүндө көрсөтүлгөн . Анын саптары 0дөн $(n-1)$ге чейин ,жана мамычалары 0дөн $(m-1)$ге чейин номерленген. i'инчи сап жана j'инчи мамычада $(0 \leq i \leq n-1, 0 \leq j \leq m-1)$ клетка (i,j) аркылуу белгиленет. Ар бир (i,j) - клетканын бийиктиги $a[i][j]$.

Бул тоолуу жерде **жакшы тик бурчтуктуу** чепти куруш керек. **жакшы тик бурчтуктуу** чепти тоолуу жердин четтерине кура албайбыз. Ошондуктан, r_1, r_2, c_1, жана c_2 $(1 \leq r_1 \leq r_2 \leq n-2$ жана $1 \leq c_1 \leq c_2 \leq m-2)$, төрт бүтүн санды тандаш керек, алар **жакшы тик бурчтуктуу** чептин (i,j)-клеткаларын ($r_1 \leq i \leq r_2$ жана $c_1 \leq j \leq c_2$) билдирет.

жакшы тик бурчтуктуу чеп болуш үчүн ошол тик бурчтуктун ар бир $(i, j) клеткасы төмөндөгү шартка туура келиш керек:

- $(i,j) клетканын бийиктиги (i, c_1-1)-клетканын, (i, c_2+1)) -клетканын (r_1-1, j) - клетканын жана (r_2+1, j)) - клетканын бардык бийиктиктеринен кичине болуш керек.

Канча мындай **жакшы тик бурчтуктуу** чепти жасай алабыз?

Implementation details

You should implement the following procedure:

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

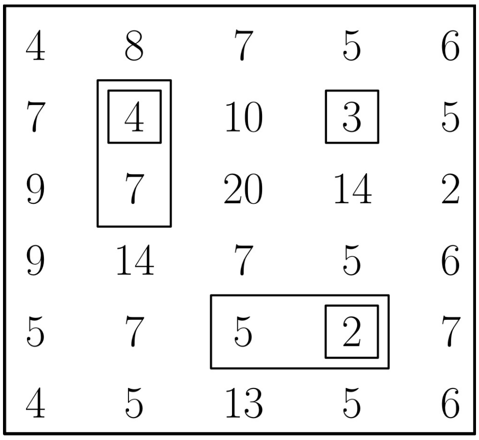
- a: a two-dimensional n by m array of integers representing the heights of the cells.
- This procedure should return the number of valid areas for the fortress.

Examples

Example 1

Consider the following call.

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



There are 5 valid areas, listed below:

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

For example $r_1 = 1, r_2 = 2, c_1 = c_2 = 1$ is a valid area because both following conditions hold:

- $a[1][1] = 4$ is strictly smaller than $a[0][1] = 8$, $a[3][1] = 14$, $a[1][0] = 7$, and $a[1][2] = 10$.
- $a[2][1] = 7$ is strictly smaller than $a[0][1] = 8$, $a[3][1] = 14$, $a[2][0] = 9$, and $a[2][2] = 20$.

Constraints

- $1 \leq n, m \leq 2500$
- $0 \leq a[i][j] \leq 7\,000\,000$ (for all $0 \leq i \leq n-1, 0 \leq j \leq m-1$)

Subtasks

1. (8 points) $n, m \leq 30$
2. (7 points) $n, m \leq 80$
3. (12 points) $n, m \leq 200$

4. (22 points) $n, m \leq 700$
5. (10 points) $n \leq 3$
6. (13 points) $0 \leq a[i][j] \leq 1$ (for all $0 \leq i \leq n-1, 0 \leq j \leq m-1$)
7. (28 points) No additional constraints.

Sample grader

The sample grader reads the input in the following format:

- line 1: $n\ m$
- line $2+i$ (for $0 \leq i \leq n-1$): $a[i][0]\ \ a[i][1]\ \ \ldots\ \ a[i][m-1]$

The sample grader prints a single line containing the return value of `count_rectangles`.