Rectangles

In the early 19th century, the ruler Hoseyngulu Khan Sardar ordered a palace (මාලිගාව) to be built on a plateau (සානුව - උස් බිමක ඇති තැනිතලා ප්‍රදේශය) overseeing a beautiful river. The plateau is modeled as an $n \times m$ grid of square cells. The rows of the grid are numbered 0 through $n-1$, and the columns are numbered 0 through $m-1$. We refer to the cell in row i and column j ($0 \le i \le n-1, 0 \le j \le m-1$) as cell (i, j). Each cell (i, j) has a specific height, denoted by $a[i][j]$.

Hoseyngulu Khan Sardar asked his architects (ගෘහ නිර්මාණ ශිල්පීන්) to choose a rectangular **area** to build the palace. The area should not contain any cell from the grid boundaries (row 0, row $n-1$, column 0, and column $m-1$). Hence, the architects should choose four integers r_1, r_2, c_1, and c_2 ($1 \le r_1 \le r_2 \le n-2$ and $1 \le c_1 \le c_2 \le m-2$), which define an area consisting of all cells (i, j) such that $r_1 \le i \le r_2$ and $c_1 \le j \le c_2$.

In addition, an area is considered **valid**, if and only if for every cell (i, j) in the area, the following condition holds:

- Consider the two cells adjacent to the area in row i (cell $(i, c_1 - 1)$ and cell $(i, c_2 + 1)$) and the two cells adjacent to the area in column j (cell $(r_1 - 1, j)$ and cell $(r_2 + 1, j)$). The height of cell (i, j) should be strictly smaller than the heights of all these four cells.
  [cell (i, j) කොටුවෙහි උස cell $(i, c_1 - 1)$, cell $(i, c_2 + 1)$, cell $(r_1 - 1, j)$ සහ cell $(r_2 + 1, j)$ කොටුවල උසට වඩා අඩු විය යුතුයි.]

Your task is to help the architects find the number of valid areas for the palace (i.e., the number of choices of r_1, r_2, c_1 and c_2 that define a valid area).

Implementation details

You should implement the following procedure:

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

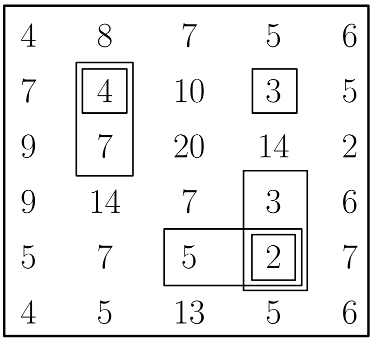
- a: a two-dimensional n by m array of integers representing the heights of the cells.
- This procedure should return the number of valid areas for the palace.

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,  3,  6],
                  [5,  7,  5,  2,  7],
                  [4,  5,  13, 5,  6]])
```



There are 6 valid areas, listed below:

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

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

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

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`.