1476 Count Negative Numbers in a Sorted Matrix (link)

Description

Given a `m x n` matrix `grid` which is sorted in non-increasing order both row-wise and column-wise, return *the number of **negative** numbers in* `grid`.

Example 1:

```
Input: grid = [[4,3,2,-1],[3,2,1,-1],[1,1,-1,-2],[-1,-1,-2,-3]]
Output: 8
Explanation: There are 8 negatives number in the matrix.
```

Example 2:

```
Input: grid = [[3,2],[1,0]]
Output: 0
```

Constraints:

- `m == grid.length`
- `n == grid[i].length`
- `1 <= m, n <= 100`
- `-100 <= grid[i][j] <= 100`

Follow up: Could you find an `O(n + m)` solution?

(scroll down for solution)

Solution

Language: cpp

Status: Accepted

```cpp
class Solution {
public:
    int countNegatives(vector<vector<int>>& grid) {
        int m = grid.size();
        int n = grid[0].size();
        int count = 0;

        // Начинаем с правого верхнего угла
        int i = 0;
        int j = n - 1;

        while (i < m && j >= 0) {
            if (grid[i][j] < 0) {
                // Все элементы слева в той же строке отрицательные
                count += (m - i);
                j--; // Двигаемся влево в той же строке
            } else {
                // Двигаемся вниз на следующую строку
                i++;
            }
        }

        return count;
    }
};
```