**1351. Count Negative Numbers in a Sorted Matrix :-**

Easy Accepted: 290.2K Submissions: 378.5K Acceptance Rate: 76.7%

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?

**Code :-**

class Solution {

public:

    int find(vector<int> &arr, int ind){

        int start = 0, end = ind, ans;

        if(arr[0] < 0)

            return 0;

        else if(arr[arr.size()-1] >= 0)

            return arr.size();

        while(start <= end){

            int mid = (start + end) / 2;

            if(arr[mid] < 0){

                ans = mid;

                end = mid - 1;

            }

            else

                start = mid + 1;

        }

        return ans;

    }

    int countNegatives(vector<vector<int>>& grid) {

        int ans=0;

        if(grid[0][0] < 0){

            for(auto arr:grid){

                ans = ans + arr.size();

            }

            return ans;

        }

        int ind=grid[0].size()-1;

        for(auto arr:grid){

            ind = find(arr, ind);

            if(ind == arr.size())

                continue;

            ans = ans + (arr.size() - ind);

        }

        return ans;

    }

};

**T.C :- O(m \* log(n))**

**S.C :- O(1)**