Count Negative Numbers in a Column-Wise and Row-Wise Sorted Matrix

Find the number of negative numbers in a column-wise / row-wise sorted matrix M[][]. Suppose M has n rows and m columns.

Example:

```
Input:  M =  [-3, -2, -1,  1]
             [-2,  2,  3,  4]
             [4,   5,  7,  8]
Output : 4
We have 4 negative numbers in this matrix
```

We strongly recommend you to minimize your browser and try this yourself first.

Naive Solution

Here's a naive, non-optimal solution.

We start from the top left corner and count the number of negative numbers one by one, from left to right and top to bottom.

With the given example:

```
[-3, -2, -1,  1]
[-2,  2,  3,  4]
[4,   5,  7,  8]

Evaluation process

[→,  →,  →,  1]
[→,  2,  3,  4]
[4,  5,  7,  8]
```

Below is Python implementation of above idea.

```
# Python implementation of Naive method to count of
# negative numbers in M[n][m]

def countNegative(M, n, m):
    count = 0

    # Follow the path shown using arrows above
    for i in range(n):
        for j in range(m):

            if M[i][j] < 0:
                count += 1

            else:
                # no more negative numbers in this row
                break
    return count


# Driver code
M = [
      [-3, -2, -1,  1],
      [-2,  2,  3,  4],
      [ 4,  5,  7,  8]
    ]
print(countNegative(M, 3, 4))
```

Output :

```
4
```

In this approach we are traversing through the all the elements and therefore, in the worst case scenario (when all numbers are negative in the matrix), this takes O(n * m) time.

Optimal Solution

Here's a more efficient solution:

1. We start from the top right corner and find the position of the last negative number in the first row.
2. Using this information, we find the position of the last negative number in the second row.
3. We keep repeating this process until we either run out of negative numbers or we get to the last row.

```
With the given example:
[-3, -2, -1,  1]
[-2,  2,  3,  4]
[4,   5,  7,  8]

Here's the idea:
[-3, -2,  ↓,  ←] -> Found 3 negative numbers in this row
[ ↓,  ←,  ←,  4] -> Found 1 negative number in this row
[ ←,  5,  7,  8] -> No negative numbers in this row
```

```
# Python implementation of Efficient method to count of
# negative numbers in M[n][m]

def countNegative(M, n, m):

    count = 0 # initialize result

    # Start with top right corner
    i = 0
    j = m - 1

    # Follow the path shown using arrows above
    while j >= 0 and i < n:

        if M[i][j] < 0:

            # j is the index of the last negative number
            # in this row.  So there must be ( j+1 )
            count += (j + 1)

            # negative numbers in this row.
            i += 1

        else:
            # move to the left and see if we can
            # find a negative number there
             j -= 1
    return count

# Driver code
M = [
      [-3, -2, -1,  1],
      [-2,  2,  3,  4],
      [4,   5,  7,  8]
    ]
print(countNegative(M, 3, 4))
```

Output :

```
4
```

With this solution, we can now solve this problem in O(n + m) time.