

Maximum Matrix Sum

🔽 solved by	Senan
🔽 Platform	LeetCode
🔗 difficulty	Medium
# Serial	1975
≡ tags	Logic
🗨 language	C++
📅 solved on	@24/11/2024
🔗 link	https://leetcode.com/problems/maximum-matrix-sum/
☑ Completion	✓

Intuition

The task is to maximize the sum of the elements of the matrix after performing any number of sign flips. Flipping the signs of elements allows us to manipulate the total sum, but we need to account for constraints:

- The sum is maximized if all elements are positive.
- If there's an odd count of negative elements, one negative value will always remain after flipping. To minimize the penalty, we should flip the smallest absolute value element.

Approach

1. Traverse the matrix to calculate:
 - The total sum of the absolute values of all elements (`sum`).
 - The smallest absolute value (`mini`).
 - The count of negative elements (`evenCount`).
2. If the count of negative numbers (`evenCount`) is even, all negatives can be flipped to positive, so the sum is the maximum possible (`sum`).
3. If `evenCount` is odd, one negative will remain. Subtract twice the smallest absolute value (`mini`) from the total sum to account for the penalty.

Complexity

Time Complexity:

$O(m \times n)$ – The matrix is traversed once, where `m` is the number of rows and `n` is the number of columns.

Space Complexity:

$O(1)$ – Only a few variables are used for computation.

Code

```
class Solution {
public:
```

```
long long maxMatrixSum(vector<vector<int>>& matrix) {
    int m = matrix.size();
    int n = matrix[0].size();

    long long sum = 0;
    int evenCount = 0;
    int mini = INT_MAX;

    for(int i = 0; i < m; i++) {
        for(int j = 0; j < n; j++) {
            sum += abs(matrix[i][j]);
            mini = min(mini, abs(matrix[i][j]));
            if(matrix[i][j] < 0) evenCount++;
        }
    }

    return (evenCount % 2 == 0) ? sum : sum - 2 * mini;
};
```