5835. Maximum Matrix Sum

My Submissions (/contest/biweekly-contest-59/problems/maximum-matrix-sum/submissions/)

Back to Contest (/contest/biweekly-contest-59/)

User Accepted:	1
User Tried:	1
Total Accepted:	1
Total Submissions:	1
Difficulty:	Medium

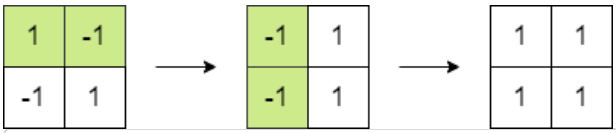
You are given an `n x n` integer `matrix`. You can do the following operation **any** number of times:

- Choose any two **adjacent** elements of `matrix` and **multiply** each of them by `-1`.

Two elements are considered **adjacent** if and only if they share a **border**.

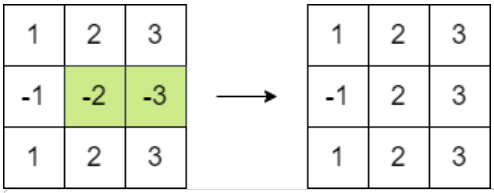
Your goal is to **maximize** the summation of the matrix's elements. Return *the **maximum** sum of the matrix's elements using the operation mentioned above.*

Example 1:




```
Input: matrix = [[1,-1],[-1,1]]
Output: 4
Explanation: We can follow the following steps to reach sum equals 4:
- Multiply the 2 elements in the first row by -1.
- Multiply the 2 elements in the first column by -1.
```

Example 2:




```
Input: matrix = [[1,2,3],[-1,-2,-3],[1,2,3]]
Output: 16
Explanation: We can follow the following step to reach sum equals 16:
- Multiply the 2 last elements in the second row by -1.
```

Constraints:

- `n == matrix.length == matrix[i].length`
- `2 <= n <= 250`
- `-10^5 <= matrix[i][j] <= 10^5`

JavaScript

```
const maxMatrixSum = (g) => {
    let n = g.length, m = g[0].length, sum = 0, negative = 0, min = Number.MAX_SAFE_INTEGER;
    for (let i = 0; i < m; i++) {
        for (let j = 0; j < n; j++) {
            sum += Math.abs(g[i][j]);
            if (g[i][j] < 0) negative++;
            min = Math.min(min, Math.abs(g[i][j]));
        }
    }
```

```
10      if (negative & 1) sum -= 2 * min;
11      return sum;
12  };
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/541941394/) | More Details > (/submissions/detail/541941394/)

Share your acceptance!