Problem 1975: Maximum Matrix Sum

Problem Information

Difficulty: Medium
Acceptance Rate: 65.84%
Paid Only: No
Tags: Array, Greedy, Matrix

Problem Description

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` * `-105 <= matrix[i][j] <= 105`

Code Snippets

C++:

```
class Solution {
public:
long long maxMatrixSum(vector<vector<int>>& matrix) {

}
};
```

Java:

```
class Solution {
public long maxMatrixSum(int[][] matrix) {

}
}
```

Python3:

```
class Solution:
def maxMatrixSum(self, matrix: List[List[int]]) -> int:
```