

Problem 3239: Minimum Number of Flips to Make Binary Grid Palindromic I

Problem Information

Difficulty: Medium

Acceptance Rate: 74.65%

Paid Only: No

Tags: Array, Two Pointers, Matrix

Problem Description

You are given an `m x n` binary matrix `grid`.

A row or column is considered **palindromic** if its values read the same forward and backward.

You can **flip** any number of cells in `grid` from `0` to `1`, or from `1` to `0`.

Return the **minimum** number of cells that need to be flipped to make **either** all rows **palindromic** or all columns **palindromic**.

Example 1:

Input: grid = [[1,0,0],[0,0,0],[0,0,1]]

Output: 2

Explanation:

Flipping the highlighted cells makes all the rows palindromic.

Example 2:

****Input:**** grid = [[0,1],[0,1],[0,0]]

****Output:**** 1

****Explanation:****

Flipping the highlighted cell makes all the columns palindromic.

****Example 3:****

****Input:**** grid = [[1],[0]]

****Output:**** 0

****Explanation:****

All rows are already palindromic.

****Constraints:****

* `m == grid.length` * `n == grid[i].length` * `1 <= m * n <= 2 * 105` * `0 <= grid[i][j] <= 1`

Code Snippets

C++:

```
class Solution {
public:
    int minFlips(vector<vector<int>>& grid) {
        }
};
```

Java:

```
class Solution {  
public int minFlips(int[][] grid) {  
}  
}  
}
```

Python3:

```
class Solution:  
def minFlips(self, grid: List[List[int]]) -> int:
```