Problem 2174: Remove All Ones With Row and Column Flips II

Problem Information

Difficulty: Medium
Acceptance Rate: 67.11%
Paid Only: Yes
Tags: Array, Bit Manipulation, Breadth-First Search, Matrix

Problem Description

You are given a **0-indexed** `m x n` **binary** matrix `grid`.

In one operation, you can choose any `i` and `j` that meet the following conditions:

* `0 <= i < m` * `0 <= j < n` * `grid[i][j] == 1`

and change the values of **all** cells in row `i` and column `j` to zero.

Return _the**minimum** number of operations needed to remove all _`1` _' s from _`grid` _._

Example 1:

![](https://assets.leetcode.com/uploads/2022/02/13/image-20220213162716-1.png)

Input: grid = [[1,1,1],[1,1,1],[0,1,0]] **Output:** 2 **Explanation:** In the first operation, change all cell values of row 1 and column 1 to zero. In the second operation, change all cell values of row 0 and column 0 to zero.

Example 2:

![](https://assets.leetcode.com/uploads/2022/02/13/image-20220213162737-2.png)

Input: grid = [[0,1,0],[1,0,1],[0,1,0]] **Output:** 2 **Explanation:** In the first operation, change all cell values of row 1 and column 0 to zero. In the second operation, change all cell values of row 2 and column 1 to zero. Note that we cannot perform an operation using row 1 and column 1 because grid[1][1] != 1.

Example 3:

![](https://assets.leetcode.com/uploads/2022/02/13/image-20220213162752-3.png)

Input: grid = [[0,0],[0,0]] **Output:** 0 **Explanation:** There are no 1's to remove so return 0.

Constraints:

* `m == grid.length` * `n == grid[i].length` * `1 <= m, n <= 15` * `1 <= m * n <= 15` * `grid[i][j]` is either `0` or `1`.

Code Snippets

C++:

```
class Solution {
public:
int removeOnes(vector<vector<int>>& grid) {

}
};
```

Java:

```
class Solution {
public int removeOnes(int[][] grid) {

}
}
```

Python3:

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