

奇数值单元格的数目

给你一个 $m \times n$ 的矩阵，最开始的时候，每个单元格中的值都是 0。

另有一个二维索引数组 `indices`，`indices[i] = [ri, ci]` 指向矩阵中的某个位置，其中 `ri` 和 `ci` 分别表示指定的行和列（从 0 开始编号）。

对 `indices[i]` 所指向的每个位置，应同时执行下述增量操作：

1. `ri` 行上的所有单元格，加 1。
2. `ci` 列上的所有单元格，加 1。

给你 `m`、`n` 和 `indices`。请你在执行完所有 `indices` 指定的增量操作后，返回矩阵中 **奇数值单元格** 的数目。

示例 1：

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 & 1 \\ 1 & 3 & 1 \end{bmatrix}$$

输入：`m = 2, n = 3, indices = [[0,1],[1,1]]`

输出：6

解释：最开始的矩阵是 `[[0,0,0],[0,0,0]]`。

第一次增量操作后得到 `[[1,2,1],[0,1,0]]`。

最后的矩阵是 `[[1,3,1],[1,3,1]]`，里面有 6 个奇数。

示例 2：

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix}$$

输入：`m = 2, n = 2, indices = [[1,1],[0,0]]`

输出：0

解释：最后的矩阵是 `[[2,2],[2,2]]`，里面没有奇数。

高

```
class Solution {
public:
    int oddCells(int m, int n, vector<vector<int>>& indices) {

        //在二维数组的基础上根据题意操作 多次 for 循环，不仅代码冗余，而且复杂度

        vector<vector<int>> res(m,vector<int>(n));
        for(auto& x :indices)
        {
            for(int i=0;i<n;i++)
                res[x[0]][i]++;
        }
        for(auto& x :indices)
        {
            for(int i=0;i<m;i++)
                res[i][x[1]]++;
        }
        int sum=0;
        for(int i=0;i<m;i++)
        {
            for(int j=0;j<n;j++)
            {
                if((res[i][j])%2!=0)
                    sum++;
            }
        }
        return sum;
    }
};
```

执行结果： **通过** [显示详情 >](#)

[添加备注](#)

执行用时： **8 ms** ，在所有 C++ 提交中击败了 **31.23%** 的用户

内存消耗： **8 MB** ，在所有 C++ 提交中击败了 **8.50%** 的用户

炫耀一下：



[写题解，分享我的解题思路](#)

```
class Solution {
public:
    int oddCells(int m, int n, vector<vector<int>>& indices) {
        //将一个二维数组分割为行 列 以两个角度完成操作，最终求和构建二维新数组
        vector<int> row(m),col(n);
        for(auto& x :indices)
        {
            row[x[0]]++;
            col[x[1]]++;
        }
        int sum=0;
        for(int i=0;i<m;i++)
        {
            for(int j=0;j<n;j++)
            {
                if((row[i]+col[j])%2!=0)
                    sum++;
            }
        }
        return sum;
    }
};
```

执行结果： **通过** [显示详情](#) >

[添加备注](#)

执行用时： **4 ms** ，在所有 C++ 提交中击败了 **78.85%** 的用户

内存消耗： **7.6 MB** ，在所有 C++ 提交中击败了 **47.43%** 的用户

炫耀一下：



[写题解，分享我的解题思路](#)