题目难度：MEDIUM

有一个二维矩阵 A 其中每个元素的值为 0 或 1 。

移动是指选择任一行或列，并转换该行或列中的每一个值：

将所有 0 都更改为 1，将所有 1 都更改为 0。

在做出任意次数的移动后，将该矩阵的每一行都按照二进制数来解释，矩阵的得分就是这些数字的总和。

返回尽可能高的分数。

示例：

输入：[[0,0,1,1],[1,0,1,0],[1,1,0,0]]

输出：39

解释：

转换为 [[1,1,1,1],[1,0,0,1],[1,1,1,1]]

0b1111 + 0b1001 + 0b1111 = 15 + 9 + 15 = 39

提示：

1 <= A.length <= 20

1 <= A[0].length <= 20

A[i][j] 是 0 或 1

思路：模拟

遍历每一行数组的第一项，如果出现0就把对应行的值调换（这样才会让这行值最大）

然后遍历从第二列开始的每列，累计0的个数

如果大于数组的长度的一半，那么就把该列的值调换（值的和增加）

之后累计值清零（WA点：少了这一步）

然后倒序遍历数组按1取和，结束后每次临时值（初始化1）乘2取和返回结果即可

时间击败84%（时间不错）

空间击败33%（空间不咋地）

上代码：

class Solution {

public:

int sum = 0, cur = 1, res = 0;

int matrixScore(vector<vector<int>>& A) {

for(int i = 0; i < A.size(); i++) {

if(A[i][0] == 0) {

for(int j = 0; j < A[i].size(); j++) {

if(A[i][j] == 0) A[i][j] = 1;

else A[i][j] = 0;

}

}

}

for(int i = 1; i < A[0].size(); i++) {

for(int j = 0; j < A.size(); j++) if(A[j][i] == 0) sum++;

if(sum > A.size() / 2) {

for(int j = 0; j < A.size(); j++) {

if(A[j][i] == 0) A[j][i] = 1;

else A[j][i] = 0;

}

}

sum = 0;

}

for(int i = A[0].size() - 1; i >= 0; i--) {

for(int j = 0; j < A.size(); j++) if(A[j][i] == 1) res += cur;

cur \*= 2;

}

return res;

}

};