

```

1  /*
2  有一个二维矩阵 A 其中每个元素的值为 0 或 1 。
3
4  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。
5
6  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。
7
8  返回尽可能高的分数。
9
10
11
12  示例：
13
14  输入: [[0,0,1,1],[1,0,1,0],[1,1,0,0]]
15  输出: 39
16  解释:
17  转换为 [[1,1,1,1],[1,0,0,1],[1,1,1,1]]
18  0b1111 + 0b1001 + 0b1111 = 15 + 9 + 15 = 39
19
20
21
22  提示：
23
24      1 <= A.length <= 20
25      1 <= A[0].length <= 20
26      A[i][j] 是 0 或 1
27
28  来源：力扣（LeetCode）
29  链接：https://leetcode-cn.com/problems/score-after-flipping-matrix
30  著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
31  */

```

分析:

- 通过行变换，把第一列全部变成1.
- 通过列变换，使得每一列1的个数大于或等于列数的一半.

方法一:C++_行列变换

```

1  class solution
2  {
3      private:
4          void __reverse_row(vector<vector<int>>& A , int row )
5          {
6              for(int i = 0 ; i < A[row].size() ; i++)
7              {
8                  A[row][i] ^= 0x01;
9              }
10         }
11
12         void __reverse_col(vector<vector<int>>& A , int col )

```

```

13     {
14         for(int i = 0 ; i < A.size() ; i++)
15         {
16             A[i][col] ^= 0x01;
17         }
18     }
19
20 public:
21     int matrixScore(vector<vector<int>>& A)
22     {
23         int ret_val = 0;
24         int count = 0;
25
26         /*处理第一列*/
27         for(int i = 0; i < A.size();i++)
28         {
29             if(A[i][0]==0)
30             {
31                 __reverse_row(A,i);
32             }
33         }
34
35         /*处理后续的列*/
36         for(int i = 1 ; i < A[0].size() ; i++)
37         {
38             count = 0;
39             for(int j = 0 ; j < A.size(); j++)
40             {
41                 count += A[j][i];
42             }
43
44             if(count*2 < A.size())
45             {
46                 __reverse_col(A,i);
47             }
48         }
49
50         /* 计算结果*/
51         for(int i = 0 ; i < A.size(); i++)
52         {
53             count = 0 ;
54             for(int j = 0; j < A[0].size(); j++)
55             {
56                 count *= 2;
57                 count += A[i][j];
58             }
59             ret_val += count;
60         }
61
62         return ret_val;
63     }
64 };

```