

美团笔试-换座位

```
// 班级换座位 n行 m大列 每个大列中有a个小列
// 换座位规则：在第i<n行的同学，新位置在第i+1行，若i=n，则新位置在第1行
// 在第j<m大列的同学，新位置在第j+1大列，若j=m，则新位置在第1大列
// 对于坐在第k小列的同学，新位置仍然在第k小列
// 现在换了一批新桌椅，可以调节桌椅高度
// 想知道，换一次位置后，有多少同学需要重新调整桌子高度
// 输入规则如下：
// 输入第一行包含三个数 n, m, a，意义如题目所示。
// 接下来 n 行，每行 m 个长度为 a 的连续01 字符串，表示目前小团班上的桌子情况。其中 0 表示这个位置未调节桌子高度，1 表示已调节桌子高度。
// 输出一个数，表示换一次位置后，需要重新调整桌子高度的同学数。

//输入样例如下：
/**
    • 3 3 2
    • 01 10 00
    • 10 00 11
    • 01 00 00
    • */
```

```
1  /*换座位问题*/
2  #include <iostream>
3  #include <string>
4  using namespace std;
5
6  // 真的鸡贼 老团子
7
8  int main()
9  {
10     // 先构造输入输出
```

```

11     int n, m, a;
12     cin >> n >> m >> a;
13     int arr[n][m][a];
14     int count = 0; // 最终返回count 即为需要重新调整桌椅位置
    的同学人数
15
16     // 输入数据
17     for (int i = 0; i < n; i++)
18     {
19         for (int j = 0; j < m; j++)
20         {
21             string temp;
22             cin >> temp; // 输入的是一个01字符串
23             for (int k = 0; k < a; k++)
24             {
25                 if (temp[k] == '0')
26                     arr[i][j][k] = 0;
27                 else
28                     arr[i][j][k] = 1;
29             }
30         }
31     }
32
33     // 算法 其实就是暴力比较每一次的新位置和当前的是否一致
34     // 不一致则count++
35     // 三重循环 希望不会超时
36     for (int i = 0; i < n; i++)
37     {
38         for (int j = 0; j < m; j++)
39         {
40             for (int k = 0; k < a; k++)
41             {
42                 if (i < n - 1 && j < m - 1)
43                 {
44                     if (arr[i + 1][j + 1][k] != arr[i]
    [j][k])
45                     {
46                         count++;
47                     }
48                 }

```

```

49         else if (i < n - 1 && j == m - 1)
50         {
51             if (arr[i + 1][0][k] != arr[i][j]
[k])
52             {
53                 count++;
54             }
55         }
56         else if (i == n - 1 && j < m - 1)
57         {
58             if (arr[0][j + 1][k] != arr[i][j]
[k])
59             {
60                 count++;
61             }
62         }
63         else
64         {
65             if (arr[0][0][k] != arr[i][j][k])
66             {
67                 count++;
68             }
69         }
70     }
71 }
72 }
73 cout << count << endl;
74 return 0;
75 }

```