习题课（枚举）

TEACHING COURSEWARE POWPOINT

授课时间：2025.07.26

启航舱
中小学拔尖创新人才培养中心

目录

01

题目描述

TEACHING
COURSEWARE

TEACH

题目描述

涂条纹

题目描述

输入格式

第一行是两个整数 N, M。

接下来 N 行是一个矩阵，矩阵的每一个小方块是 `W`（白），`B`（蓝），`R`（红）中的一个。

输出格式

一个整数，表示至少需要涂多少块。

只要一个由 $N \times M$ 个小方块组成的旗帜符合如下规则，就是合法的图案。

- 从最上方若干行（至少一行）的格子全部是白色的；
- 接下来若干行（至少一行）的格子全部是蓝色的；
- 剩下的行（至少一行）全部是红色的；

现有一个棋盘状的布，分成了 N 行 M 列的格子，每个格子是白色蓝色红色之一，小 a 希望把这个布改成合法图案，方法是在一些格子上涂颜料，盖住之前的颜色。

小 A 很懒，希望涂最少的格子，使这块布成为一个合法的图案。

题目描述

涂条纹

样例输入

```
4 5
WRWRW
BWRWB
WRWRW
RWBWR
```

样例输出

```
11
```

说明/提示

样例解释

目标状态是：

```
WWWWW
BBBBB
RRRRR
RRRRR
```

启航舱
中小学拔尖创新人才培养中心

02

知识回归

TEACHING
COURSEWARE

TEACH

知识回归

“在计算机科学中，枚举算法（也叫‘穷举法’）是指：明确问题的所有可能解的范围（即‘解空间’），逐一列举所有候选解，对每个候选解验证是否满足问题的条件，最终得到所有有效解的算法。

简单说，就是让计算机‘按顺序把所有可能的情况过一遍，符合要求的就记下来。”

核心思想：

“枚举算法的核心可以浓缩成两个动作：

第一个是‘遍历’：系统地、不重不漏地列出解空间里的所有候选解；

第二个是‘验证’：对每个候选解，用问题的条件检查它是否有效。

启航舱
中小学拔尖创新人才培养中心

03

题目解答

TEACHING
COURSEWARE

TEACH

题目解答

枚举白与蓝，蓝与红的分界

启航舱
中小学拔尖创新人才培养中心

启航舱
中小学拔尖创新人才培养中心

04

标准程序

TEACHING
COURSEWARE

TEACH

标准程序

启航舱
中小学拔尖创新人才培养中心

```cpp
#include <iostream>
#include <climits>
using namespace std;

int main() {
    int n, m;
    char a[55][55];
    int ans = INT_MAX;
    int sum;
    cin >> n >> m;
    for(int i = 1; i <= n; i++) {
        for(int j = 1; j <= m; j++) {
            cin >> a[i][j];
        }
    }
```

标准程序

```
for(int i = 1; i <= n-2; i++) {  // 白色至少1行，留至少2行给蓝红
    for(int j = i+1; j <= n-1; j++) {  // 蓝色至少1行，红色至少1行
        sum = 0;
        // 白色区域：1到i行
        for(int x = 1; x <= i; x++) {
            for(int y = 1; y <= m; y++) {
                if(a[x][y] != 'W') sum++;
            }
        }
        // 蓝色区域：i+1到j行
        for(int x = i+1; x <= j; x++) {
            for(int y = 1; y <= m; y++) {
                if(a[x][y] != 'B') sum++;
            }
        }
        // 红色区域：j+1到n行
        for(int x = j+1; x <= n; x++) {
            for(int y = 1; y <= m; y++) {
                if(a[x][y] != 'R') sum++;
            }
        }
        if(sum < ans) ans = sum;
    }
}
```

i：白色区域结束行（含i），白色区域为[1，i]
j：蓝色区域结束行（含j），蓝色区域为[i+1，j]
红色区域为[j+1，n]，需保证各区域至少1行

Question：
如何优化呢？

标准程序

```
for(int y = 1; y <= m; y++) {
    if(a[x][y] != 'W') sum++;
for(int y = 1; y <= m; y++) {
    if(a[x][y] != 'B') sum++;
}
for(int y = 1; y <= m; y++) {
    if(a[x][y] != 'R') sum++;
}
```

启航舱
中小学拔尖创新人才培养中心

标准程序

启航舱
中小学拔尖创新人才培养中心

```
// 预处理：计算每行涂成W、B、R所需的修改次数
for (int i = 1; i <= n; i++) {  // 遍历每行（1~n）
    for (int j = 1; j <= m; j++) {  // 遍历每列（1~m）
        char c = a[i][j];
        // 统计当前行涂成白色需要修改的格子
        if (c != 'W') W[i]++;
        // 统计当前行涂成蓝色需要修改的格子
        if (c != 'B') B[i]++;
        // 统计当前行涂成红色需要修改的格子
        if (c != 'R') R[i]++;
    }
}
```

标准程序

启航舱
中小学拔尖创新人才培养中心

```
// 累加白色区域的修改次数（1~i行）
for (int x = 1; x <= i; x++) {
    sum += W[x];
}
// 累加蓝色区域的修改次数（i+1~j行）
for (int x = i + 1; x <= j; x++) {
    sum += B[x];
}
// 累加红色区域的修改次数（j+1~n行）
for (int x = j + 1; x <= n; x++) {
    sum += R[x];
}
```