

## P1596 [USACO10OCT]Lake Counting S

## 题目描述

由于近期的降雨，雨水汇集在农民约翰的田地不同的地方。我们用一个  $N \times M$  ( $1 \leq N \leq 100$ ;  $1 \leq M \leq 100$ ) 网格图表示。每个网格中有水('W')或是旱地('.')。一个网格与其周围的八个网格相连，而一组相连的网格视为一个水坑。约翰想弄清楚他的田地已经形成了多少水坑。给出约翰田地的示意图，确定当中有多少水坑。

## 输入格式

第1行：两个空格隔开的整数：N 和 M 第2行到第N+1行：每行M个字符，每个字符是'W'或'.'，它们表示网格图中的一排。字符之间没有空格。

## 输出格式

一行：水坑的数量

## 输入样例

```
10 12
W.....WW.
. WWW.....WWW
....WW...WW.
.....WW.
.....W..
..W.....W..
.W.W.....WW.
W.W.W.....W.
.W.W.....W.
..W.....W.
```

## 输出样例

3

## 解析

本题推荐使用深度优先搜索，同时使用染色法进行优化。

在一本通上的《细胞》一题中，我们使用了vis数组来进行某个点是否被访问过，在这里我们可以将那个步骤进行省略，改变成只要走过了就被标记成为“.”，这样的操作，将会让代码的效率更高。

在明确了上文的操作之后，对于本题，我们则可以进行如此判断：每当遇到一个w，便意味着遇到了一个全新的水坑，因为原本连接在一起的w，经过搜索算法之后，都被染成了“.”，因此不会存在任何的重复问题。

## 编码

```
#include<bits/stdc++.h>

using namespace std;
//方向数组
int forwardX[8] = { -1, -1, -1, 0, 0, 1, 1, 1}; //x方向
int forwardY[8] = { -1, 0, 1, -1, 1, -1, 0, 1}; //y方向
//行列数，连通块数
int n, m, ans;
//原始的地图数据
char maps[105][105];

void dfs(int x, int y) {
    int r, c;
    //将当前点标记为不可作为起点，防止被重复计算
    maps[x][y] = '.';
    //分别向剩下的八方向移动
    for (int i = 0; i < 8; i++) {
        r = x + forwardX[i];
        c = y + forwardY[i];
        //判断是否出界
        if (r < 1 || r > n || c < 1 || c > m || maps[r][c] == '.') {
            continue;
        }
        //将走过的点都标记为不可作为起点
        maps[r][c] = '.';
        dfs(r, c);
    }
}

int main() {
    //读入行列数
    scanf("%d %d\n", &n, &m);
    //读入初始地图数据
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++)
```

```

        cin >> maps[i][j];
    }
    //连通块的个数
    ans = 0;
    //遍历所有的节点
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= m; j++) {
            //找到了一个起点
            if (maps[i][j] == 'W') {
                //记录连通块
                ans++;
                //以这个点作为起点
                dfs(i, j);
            }
        }
    }
    printf("%d\n", ans);
    return 0;
}

```

逻辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

