

c++-DFS 搜索-有一个大小是 $N \times M$ 的战场地图

题目描述:

有一个大小是 $N \times M$ 的战场地图, 被墙壁'#'分隔成大小不同的区域, 上下左右四个方向相邻的空地 '.' 属于同一个区域, 只有空地上可能存在敌人'E', 请求出地图上总共有多少区域里的敌人数小于 K 。

输入描述:

第一行输入为 N, M, K ;

N 表示地图的行数, M 表示地图的列数, K 表示目标敌人数量 $N, M \leq 100$;

之后为一个 $N \times M$ 大小的字符数组。

输出描述:

敌人数小于 K 的区域数量

补充说明:

示例 1

输入:

3 5 2

..#EE

E.#E.

###..

输出:

1

说明:

地图被墙壁分为两个区域, 左边区域有 1 个敌人, 右边区域有 3 个敌人, 符合条件的区域数量是 1

```
#include <iostream>
```

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

```
using namespace std;
```

```
int n, m, k, cou;
```

```
bool vis[105][105];
```

```
char g[105][105];
```

```
int X[4] = {0, 0, 1, -1};
```

```
int Y[4] = {1, -1, 0, 0};
```

```
bool check(int x, int y){
```

```
    if(x >= n || x < 0 || y >= m || y < 0) return false;
```

```
    if(vis[x][y]) return false;
```

```
    if(g[x][y] == '#') return false;
```

```
    return true;
```

```
}
```

```
void dfs(int x, int y){
```

```
    if(g[x][y] == 'E') cou++;
```

```
    vis[x][y] = true;
```

```
    for(int i = 0; i < 4; i++){
```

```
        if(check(x + X[i], y + Y[i])){
```

```
            dfs(x + X[i], y + Y[i]);
```

```
        }
```

```

    }
}
int main() {
    memset(vis, false, sizeof(vis));
    cin >> n >> m >> k;
    for(int i = 0; i < n; i++){
        for(int j = 0; j < m; j++){
            cin >> g[i][j];
        }
    }
    int ans = 0;
    for(int i = 0; i < n; i++){
        for(int j = 0; j < m; j++){
            if(!vis[i][j] && g[i][j] != '#'){
                cou = 0;
                dfs(i, j);
                if(cou < k) ans++;
            }
        }
    }
    cout << ans << endl;
}
// 64 位输出请用 printf("%lld")

```