

P1605 迷宫

题目描述

给定一个 $N \times M$ 方格的迷宫，迷宫里有 T 处障碍，障碍处不可通过。给定起点坐标和终点坐标，问：每个方格最多经过1次，有多少种从起点坐标到终点坐标的方案。在迷宫中移动有上下左右四种方式，每次只能移动一个方格。数据保证起点上没有障碍。

输入格式

第一行 N 、 M 和 T ， N 为行， M 为列， T 为障碍总数。第二行起点坐标 SX, SY ，终点坐标 FX, FY 。接下来 T 行，每行为障碍点的坐标。

输出格式

给定起点坐标和终点坐标，问每个方格最多经过1次，从起点坐标到终点坐标的方案总数。

输入样例

```
2 2 1
1 1 2 2
1 2
```

输出样例

```
1
```

求方案总数，非常经典的深搜问题，直接套用深搜模板即可。

编码

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

using namespace std;
//globalMap为总地图，vis记录是否访问
bool globalMap[15][15], vis[15][15];
//上下左右四方向
// -1, 0
// 1, 0
// 0, -1
// 0, 1
int n, m, d[5] = {-1, 0, 1, 0, -1};
//起点和终点坐标，方案数
int nx, ny, ex, ey, cnt;

//nx,ny起点坐标;ex,ey终点坐标,cnt路径条数
void dfs(int x, int y) {
    //如果到终点
```

```

if (x == ex && y == ey) {
    cnt++; //路径加一
    return; //回去继续查找
}
for (int k = 0; k < 4; k++) {
    int newX = x + d[k];
    int newY = y + d[k + 1];
    //未越界
    //不是障碍点
    //没有访问过
    if (newX >= 1 && newY >= 1 && newX <= n && newY <= m
        && !globalMap[newX][newY] && !vis[newX][newY]) {
        vis[newX][newY] = true; //标记为已访问
        dfs(newX, newY);
        vis[newX][newY] = false; //回溯
    }
}
return;
}

int main() {
    int t, zx, zy;
    cin >> n >> m >> t >> nx >> ny >> ex >> ey;
    //将起点标记为已经访问
    vis[nx][ny] = true;
    //读入障碍点
    while (t--) {
        cin >> zx >> zy;
        //记录障碍点
        globalMap[zx][zy] = true;
    }
    //从起点开始搜索
    dfs(nx, ny);
    cout << cnt;
    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。

