C++方向编程题答案

第八周

day44

> 题目ID：790 红与黑
>
> 链接：https://www.nowcoder.com/questionTerminal/5017fd2fc5c84f78bbaed4777996213a

[题目解析]

1. 输入的m和n就是代表输入后续会输入几行几列字符
2. 第二行开始，输入的字符就是我们的“行走矩阵”，其中“.”->黑色的瓷砖，“#”->白色的瓷砖，“@”->黑色的瓷砖，并且你站在这块瓷砖上
3. 这道题的核心问题是，从你站的位置开始，向周边任意位置走，你能直接走过的黑色瓷砖的总数是多少

[解题思路]

该问题可以采用深度优先遍历的方式，统计所有的可能性，具体细节见代码

```
/*
    DFS问题
*/
#include <iostream>
#include <fstream>
#include <vector>
#include <queue>
using namespace std;

struct pos { int x, y; };
int bfs(vector<vector<char> > & map, vector<vector<bool> > & visit, pos & start)
{
    const int dir[4][2] = { {-1,0},{1,0},{0,-1},{0,1} };
    queue<pos> que;
    int count = 0;
    int m = map.size(), n = map[0].size();

    que.push(start);
    visit[start.x][start.y] = true; ++count;
    while (!que.empty())
    {
        pos cur = que.front(), next;
        que.pop();
        for (int i = 0; i < 4; ++i)
        {
            next.x = cur.x + dir[i][0];
            next.y = cur.y + dir[i][1];
            if (next.x >= 0 && next.x < m && next.y >= 0 && next.y < n && \
                !visit[next.x][next.y] && map[next.x][next.y] == '.')

            {
```

```
                que.push(next);
                visit[next.x][next.y] = true;
                ++count;
            }
        }
    }

    return count;
}

int main()
{
    int m, n;
    while (cin >> m >> n && (m*n))
    {
        pos start;
        vector<vector<char> > map(m, vector<char>(n));
        vector<vector<bool> > visit(m, vector<bool>(n));
        for (int i = 0; i < m; ++i)
            for (int j = 0; j < n; ++j)
            {
                visit[i][j] = false;
                cin >> map[i][j];
                if (map[i][j] == '@')
                    start.x = i, start.y = j;
            }

        cout << bfs(map, visit, start) << endl;
    }

    return 0;
}
```

题目ID：25951 蘑菇阵

链接：https://www.nowcoder.com/questionTerminal/ed9bc679ea1248f9a3d86d0a55c0be10

[题目解析]

1.

[解题思路]

```
/*
要使用动态规划
*/
#include<iostream>
#include <iomanip>
#include<algorithm>
#include<vector>
using namespace std;

int main()
```

```
{
    int n, m, k;
    while(cin >> n >> m >> k){
        vector<vector<int> > table((n+1), vector<int>(m+1));//记录蘑菇
        vector<vector<double> > P((n+1), vector<double>(m+1));//P[i][j]表示不碰到蘑菇走到i,
j的概率
        int x, y;
        for(int i = 0; i < k; i++){
            cin >> x >> y;
            table[x][y] = 1;
        }
        P[1][1] = 1.0;      //起点概率为1
        for(int i = 1; i <= n; i++){
            for(int j = 1; j <= m; j++)
            {
                if(!(i == 1 && j ==1)){      //跳过起点
                    P[i][j] = P[i-1][j]*(j == m? 1 : 0.5) + P[i][j-1]*(i == n?1:0.5);   //
边界的时候，概率为1
                    if(table[i][j] == 1) P[i][j] = 0;        //如果该点有蘑菇，概率置为0
                }
            }
        }
        cout << fixed << setprecision(2) << P[n][m] << endl;
    }
}
```