求岛屿面积

1，2，1，0，0，0，0，0，2，3

3，0，2，0，1，2，1，0，1，2

4，0，1，0，1，2，3，2，0，1

3，2，0，0，0，1，2，4，0，0

0，0，0，0，0，0，1，5，3，0

0，1，2，1，0，1，5，4，3，0

0，1，2，3，1，3，6，2，1，0

0，0，3，4，8，9，7，5，0，0

0，0，0，3，7，8，6，0，1，2

0，0，0，0，0，0，0，0，1，0

在这张二维地图中，数字代表海拔，0代表大海，求某一个点（比如6，8）所在岛屿的面积使用广度优先遍历，从出发点开始，向周围尝试，碰到大海则尝试失败，继续尝试其他的位置。

#include "stdafx.h"

#include "iostream"

#define X 6

#define Y 8

using namespace std;

typedef struct Node

{

int x;

int y;

};

void main()

{

int map[11][11] = {

0,0,0,0,0,0,0,0,0,0,0,

0,1,2,1,0,0,0,0,0,2,3,

0,3,0,2,0,1,2,1,0,1,2,

0,4,0,1,0,1,2,3,2,0,1,

0,3,2,0,0,0,1,2,4,0,0,

0,0,0,0,0,0,0,1,5,3,0,

0,0,1,2,1,0,1,5,4,3,0,

0,0,1,2,3,1,3,6,2,1,0,

0,0,0,3,4,8,9,7,5,0,0,

0,0,0,0,3,7,8,6,0,1,2,

0,0,0,0,0,0,0,0,0,1,0 };

bool visited[11][11] = { false };

Node q[101];

int head=1, tail = 1;

int sum;

int tx=0, ty=0;

int next[4][2] = { 0,1,1,0,0,-1,-1,0 }；

q[tail].x = X;

q[tail].y = Y;

visited[X][Y] = true;

tail++;

if (map[X][Y] <= 0) { sum=0;cout << sum << endl;return; }

else sum = 1;

while (head < tail)

{

for (int i = 0;i <= 3;i++)

{

tx = q[head].x + next[i][0];

ty = q[head].y + next[i][1];

if (tx < 1 || tx>10 || ty < 1 || ty>10)continue;

if (map[tx][ty] > 0 && visited[tx][ty] == false)

{

q[tail].x = tx;

q[tail].y = ty;

visited[tx][ty] = true;

tail++;

sum++;

}

}

head++;

}

cout << sum << endl;

}