



二维数组Bfs走迷宫模板。

#include<iostream>

#include<string.h>

#include<algorithm>

#include<stdio.h>

#include<queue>

using namespace std;

#define x first

#define y second

typedef pair<int,int> PII;

const int num = 210;

#define LL long long

char re[num][num];

int juage[num][num];

int t , r ,c;

PII start , End;

LL sum[num][num];

int bfs(){

memset(juage, 0 ,sizeof(juage));

memset(sum,0,sizeof(sum));//记得对判断数组再次重置

juage[start.x][start.y] =1;

queue<PII> Q;

Q.push(start);

int lenx [] = {-1,1,0, 0 };

int leny [] = {0,0,-1, 1 };

//遍历数组来确定上下左右

while(!Q.empty()){

PII tem = Q.front();

Q.pop();

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

int temx = tem.x + lenx[i],temy =tem.y+leny[i] ;

if(temx >=r || temx < 0 || temy >=c || temy < 0 || re[temx][temy] =='#' ){//向周围bfs搜索

continue;

}

if(juage[temx][temy]==1){//如果之前访问过该坐标则跳过对其的处理

continue;

}

juage[temx][temy]=1;//记录表示访问了该坐标了

sum[temx][temy] = sum[tem.x][tem.y]+1;//表示到当前位置走了多少步

if(End == make\_pair(temx,temy)){//搜索先将判断放前面

return sum[temx][temy];

}

Q.push(make\_pair(temx,temy));

}

}

return -1;//表示没有搜索到

}

int main(void){

scanf("%d",&t);

while(t--){

scanf("%d %d" ,&r,&c);

for(int i = 0 ; i < r ; i++){

scanf("%s" , re[i]);

}

for(int i = 0 ; i < r ; i++){

for(int j = 0 ; j < c ; j++){

if(re[i][j] == 'S'){

start= {i,j};

}

else if(re[i][j] == 'E') {

End = {i,j};

}

}

}

LL ans =bfs();

if(ans == -1){

printf("oop!\n");

}

else{

printf("%lld\n",ans);

}

}

return 0 ;

}



#include<iostream>

#include<cstring>

#include<stdio.h>

#include<algorithm>

#include<queue>

using namespace std;

#define x first

#define y second

typedef pair<int,int> PII ;

const int Num = 40;

//本题先用bfs 来 实现 bfs实现后再用dfs来实现

char re[Num][Num];

char End[Num];

int juage[Num][Num];

int sum ;

int x ,y;

PII start;

int bfs(){

    int lx[]= {-1,1,0,0};

    int ly[]= {0,0,-1,1};

    sum = 0;

    memset(juage,0,sizeof (juage));

    juage[start.y][start.x] = 1;

    queue<PII> Q;

    Q.push(start);

    while(!Q.empty()){//bfs肯定得利用queue数据结构先进先出的特点来实现

        PII tem = Q.front();//出队列

        Q.pop();

        for(int i= 0 ; i < 4 ; i++){

            int nx = tem.x+lx[i],ny = tem.y+ly[i];

            if(nx<0 || nx >=x || ny < 0 || ny >=y || re[ny][nx]=='#'){

                continue;

            }

            if(juage[ny][nx]==1){

                continue;

            }

            juage[ny][nx]=1;

            sum++;

            Q.push(make\_pair(nx,ny));//进队列

        }

    }

    return sum ;

}

int main(void){

      scanf("%d %d",&x , &y);

    while(x!=0 && y!=0){

    for(int i = 0 ; i < y ; i++){

        scanf("%s",re[i]);

    }

    for(int i = 0  ;i < y ; i++){

        for(int j = 0 ; j < x ; j++){

            if(re[i][j]=='@'){//得到bfs的起点坐标

                start.x = j;

                start.y = i ;

                break;

            }

        }

    }

    cout <<bfs()+1 <<endl;

    scanf("%d %d",&x , &y);

    }

    return  0;

}