**数据结构上机实习报告**

题目：2.9 迷宫问题

班级：1403013 学号：14030130018 姓名：邓楠 完成日期：2015.12.04

**一、需求分析**

1.**二、概要设计**

为了实现上述程序功能，我设计了一个队列。

基本操作：

void init(Queue \*A)

初始化一个空对象;

void add(Queue \*A,int val)

添加元素;

void del(Queue \*A)

删除队首元素;

void clear(Queue \*A)

清空队列;

int size(Queue \*A)

返回队列中的元素个数;

int front(Queue \*A)

返回队首元素;

**三、详细设计**

#include<iostream>

#include<cstdio>

#include<cstdlib>

#include<ctime>

using namespace std;

struct Node{

int val;

Node \*nex;

};

struct Queue{

Node \*head;

Node \*tail;

int sz;

};

void init(Queue \*A);

void add(Queue \*A,int val);

void clear(Queue \*A);

void del(Queue \*A);

int size(Queue \*A);

int front(Queue \*A);

void init\_maze();

bool solve(Queue \*A);

void print(int flag);

static const int MAXN=100;

static const int MAXM=100;

int maze[MAXN][MAXM];

int vis[MAXN][MAXM];

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

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

int n,m,s,e;

Queue Que;

void init(Queue \*A){

A->head=A->tail=NULL;

A->sz=0;

}

void add(Queue \*A,int \_val){

A->sz++;

Node \*tmp=new Node;

tmp->val=\_val;

tmp->nex=NULL;

if(A->head==NULL)

A->head=A->tail=tmp;

else{

A->tail->nex=tmp;

A->tail=tmp;

}

}

void clear(Queue \*A){

Node \*tmp;

while(A->head!=NULL){

del(A);

}

}

void del(Queue \*A){

if(A->head==NULL)

return ;

A->sz--;

Node \*tmp=A->head;

A->head=A->head->nex;

delete tmp;

}

int size(Queue \*A){

return A->sz;

}

int front(Queue \*A){

return A->head->val;

}

void init\_maze(){

srand((unsigned)time(0));

s=rand()%20+20;

do{

e=rand()%20+(n\*m-21);

}while(e==s);

maze[s/m][s%m]=2;

maze[e/m][e%m]=3;

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

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

vis[i][j]=-1;

if(maze[i][j]==2||maze[i][j]==3)

continue;

maze[i][j]=rand()%10;

maze[i][j]=maze[i][j]<8?1:0;

}

}

}

bool solve(){

vis[s/m][s%m]=2;

add(&Que,s);

while(size(&Que)){

int z=front(&Que);

del(&Que);

if(z==e)

return true;

int x=z%m;

int y=z/m;

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

int tx=x+dx[i];

int ty=y+dy[i];

if(vis[ty][tx]!=-1||!maze[ty][tx]||(tx<0||tx>=m||ty<0||ty>=n))

continue;

add(&Que,ty\*m+tx);

vis[ty][tx]=z;

}

}

return false;

}

void print(int flag){

vis[s/m][s%m]=2;

vis[e/m][s%m]=3;

if(flag)

for(int k=vis[e/m][e%m];k!=s;k=vis[k/m][k%m])

maze[k/m][k%m]=-1;

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

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

switch(maze[i][j]){

case -1:

printf("\*");break;

case 0:

printf("#");break;

case 1:

printf(".");break;

case 2:

printf("@");break;

case 3:

printf("^");break;

default:

break;

}

}

printf("\n");

}

printf("\n");

}

int main(){

printf("请输入迷宫的行数和列数: ");

scanf("%d%d",&n,&m);

init(&Que);

init\_maze();

printf("\n符号代表的意义:\n'.'=空地 '#'=墙 '\*'=答案 '@'=起点 '^'=终点\n");

print(0);

if(solve())

print(1);

else

printf("There is no answer\n");

}

**四、调试分析**

1.通过尾插法和头结点删除法建立了一个队列。

2.通过bfs求解出了最短的路径。

**五、用户手册**

本程序的运行环境为WIN10，执行文件为:棋盘问题.exe

**六、测试结果**

