# 《数据结构》实验报告六/七

|  |  |  |
| --- | --- | --- |
| 题目： 图的操作 | | 班级： 通信1601 |
| 学号： 29 |  | 姓名： 郑祖闯 |
| 日期： |  | 程序名： |

### 一、上机实验的问题和要求：

1. 图的存储结构的定义和图的创建。图的种类有：有向图、无向图、有向网、无向网。

图的存储结构可采用：邻接矩阵、邻接表。要求：分别给出邻接矩阵和邻接表在某一种图上的创建算法

1. 图的遍历：非递归的深度优先搜索算法、广度优先搜索算法。
2. 从键盘读入一个图的顶点信息和带路径权值的邻接矩阵， 然后从键盘输入任意的起点和终点，要求找出从起点到终点的最短路径 （包括中间结点和各点之间路径的权值），没有路径则输出路径为无穷大。
3. 将从迷宫入口到各点的最短路径的集合看作一棵树。用广度遍历的方法即可找到出口的最短路径。

### 二、程序设计的基本思想，原理和算法描述：

（包括程序的结构，数据结构，输入/输出设计，符号名说明等）

建立存储结构的一般步骤：

1. 输入参数:vexnum, arcnum, GraghKind
2. 输入顶点信息
3. 根据GraghKind，决定边是否要带权重
4. 采用某种形式逐条输入边，将它插入到存储结构中

### 三、源程序及注释：

Status CreateGragh(ALGraph &G){//建立邻接表

scanf(”%d%d%d”, &G.vexnum, &G.arcnum,&G.GraghKind);

if……

switch(G.GraghKind){

case DG: return CreateDG(G);

case DN: return CreateDN(G);

case UDG: return CreateUDG(G);

case UDN: return CreateUDN(G);

default: return ERROR;}

}

Status CreateDG(ALGraph &G){

for(i=0;i<G.vexnum;i++){

scanf( &data);

G.vertices[i].data = data

}//输入顶点信息

……

}

Status CreateDG(ALGraph &G){

……

for(i=0;i<G.arcnum;i++){//输入边的信息

scanf(”%d%d ”,&v, &w)//形式2

p= new arcnode;//建立结点

if(!p) return ERROR;

p.adjvex=w;

p.nextarc=G.vertices[v].firstarc;//顶点v的链表

G.vertices[v].firstarc=p;//添加到最左边

}

}

Status TranslateDG(ALGraph G1, MGraph &G2)

{ //设置参数

G2.GraghKind = G1.GraghKind;

G2.vexnum = G1.vexnum;

G2.arcnum = G1.arcnum//复制顶点

for(i=0;i<G1.vexnum;i++)G2.vexs[i] = G1.vertices[i].data;//复制弧

for(i=0;i<G2.vexnum;i++)

for(j=0;j<G2.vexnum;j++)

G2.arcs[i][j]=0;

}

Status TranlateDG(ALGraph G1, MGraph &G2)

{ //将G1转换为G2

//设置参数

G2.GraghKind = G1.GraghKind;

G2.vexnum = G1.vexnum;

G2.arcnum = G1.arcnum

//复制顶点

for(i=0;i<G1.vexnum;i++)G2.vexs[i] = G1.vertices[i].data;

//复制弧

for(i=0;i<G2.vexnum;i++)

for(j=0;j<G2.vexnum;j++)

G2.arcs[i][j]=0;

}

tatus TranlateDG(ALGraph G1, MGraph &G2){

……

for(i=0;i<G1.vexnum;i++){//复制G1每个顶点的邻接点

p=G1.vertices[i].firstarc;

while(p){

G2.arcs[i][p.adjvex]=1;

p=p->nextarc;

}

}

}

/\* 迷宫探路III（最短路径）\*/  
/\* DIJKSTRAMAZE.C \*/  
/\* 2003-8-26 \*/  
#include <stdlib.h>  
#include <time.h>  
#include <math.h>  
#include <stdio.h>  
#include <graphics.h>  
#define N 22  
#define M 22  
int bg[M][N];  
int aa[M][N];  
struct pace{  
 int pre;  
 int x;  
 int y;  
 int ri;  
 int rj;  
}road[M\*N];  
struct pace bestroad[M\*N];

void makebg(int,int);  
void drawbg(int[][],int,int,int,int,int);  
void drawman(int,int,int);  
void rect(int,int,int,int);

void main(){/\* main()开始 \*/  
int step=20;  
int len=10;  
int size=20;  
int x=0,y=0;  
int i=0,j=0;  
int gdriver=DETECT,gmode;  
char ch;  
int direc;  
int routelen=0;  
int dj[]={-1,1,0,0};  
int di[]={0,0,-1,1};  
int begin;  
int end;  
int k;  
int t;  
int num;  
int suc;  
int cnt;  
int x0;  
int y0;  
int le;  
int tmp;  
makebg(M,N);  
/\* registerbgidriver(EGAVGA\_driver);  
 initgraph(&gdriver,&gmode,\"c:\\\\turboc2\");\*/  
 initgraph(&gdriver,&gmode,\"c:\\\\tc20\\\\bgi\");  
cleardevice();  
setwritemode(XOR\_PUT);  
settextstyle(1,0,3);  
setcolor(GREEN);  
outtextxy(100,180,\"DIJKSTRA MAZE\");  
setcolor(BLUE);  
setfillstyle(LINE\_FILL,BLUE);

/\*drawbg(bg,M,N,size,0,0);\*/  
drawbg(aa,M,N,size,0,0);  
setcolor(WHITE);  
x+=len;y+=len;  
drawman(x,y,len);  
x0=x;y0=y;  
/\* 电脑控制 \*/  
i=j=0;  
aa[0][0]=1;  
begin=0;  
end=0;  
road[0].ri=0;  
road[0].rj=0;  
road[0].x=x0;  
road[0].y=y0;  
road[0].pre=-1;  
le=1;  
suc=0;  
while(!suc){  
cnt=0;  
le++;  
for(k=begin;k<=end;k++){  
 for(t=0;t<4;t++){  
 x=road[k].x+dj[t]\*step;  
 y=road[k].y+di[t]\*step ;  
 i=road[k].ri+di[t];  
 j=road[k].rj+dj[t];  
 if(i<M&&i>=0&&j<N&&j>=0&&aa[i][j]==0){  
 cnt++;  
 aa[i][j]=le;  
 road[end+cnt].pre=k;  
 road[end+cnt].x=x;  
 road[end+cnt].y=y;  
 road[end+cnt].ri=i;  
 road[end+cnt].rj=j;  
 if(i==N-1&&j==M-1){  
 suc=1;  
 break;  
 }  
 }  
 }  
 if(suc)break;  
}  
begin=end+1;  
end=end+cnt;  
}  
/\* output \*/  
i=end;j=0;  
while(i!=-1){  
 bestroad[j].x=road[i].x;  
 bestroad[j].y=road[i].y;  
 bestroad[j].ri=road[i].ri;  
 bestroad[j].rj=road[i].rj;  
 i=road[i].pre;  
 j++;  
}  
for(i=0;i<j/2;i++){  
 tmp=bestroad[i].x;  
 bestroad[i].x=bestroad[j-1-i].x;  
 bestroad[j-1-i].x=tmp;  
 tmp=bestroad[i].y;  
 bestroad[i].y=bestroad[j-1-i].y;  
 bestroad[j-1-i].y=tmp;  
}  
getch();  
drawman(x0,y0,len);  
for(i=0;i<j;i++){  
 drawman(bestroad[i].x,bestroad[i].y,len);  
 delay(80000);  
 drawman(bestroad[i].x,bestroad[i].y,len);  
}  
i--;  
drawman(bestroad[i].y,bestroad[i].x,len);  
getch();  
closegraph();  
}  
/\* main()结束 \*/

/\* 绘制小人 \*/  
void drawman(int x,int y,int len){  
 int r=len/4;  
 rect(x-r,y-len,x+r,y-len+2\*r);  
 line(x,y-len+2\*r,x,y);  
 line(x-len,y,x+len,y);  
 line(x,y,x-len,y+len);  
 line(x,y,x+len,y+len);  
}  
/\* 绘制迷宫地图 \*/  
void drawbg(int bg[][N],int a,int b,int size,int x,int y){  
 int startx=x;  
 int i,j;  
 for(i=0;i<a;i++){  
 for(j=0;j<b;j++){  
 if(bg[i][j]==-1)  
 rect(x,y,x+size-1,y+size-1);  
 x+=size;  
 }  
 x=startx;  
 y+=size;  
 }  
 rectangle(0,0,size\*b,size\*a);  
 line(0,0,size,0);line(0,0,0,size);  
 line(size\*b,size\*(a-1),size\*b,size\*a);  
 line(size\*(b-1),size\*a,size\*b,size\*a);  
}

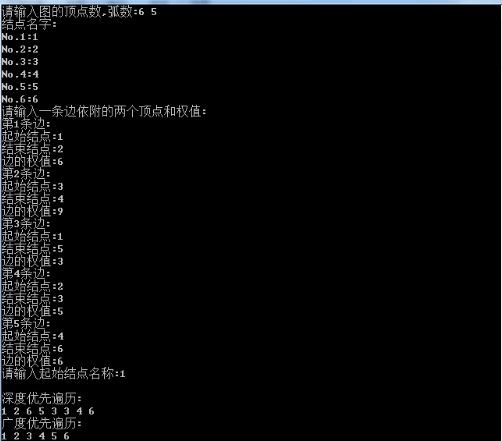
/\* 绘制实心矩形 \*/  
void rect(int x0,int y0,int x1,int y1){  
 int i,j;  
 for(i=x0;i<=x1;i++)  
 line(i,y0,i,y1);  
}

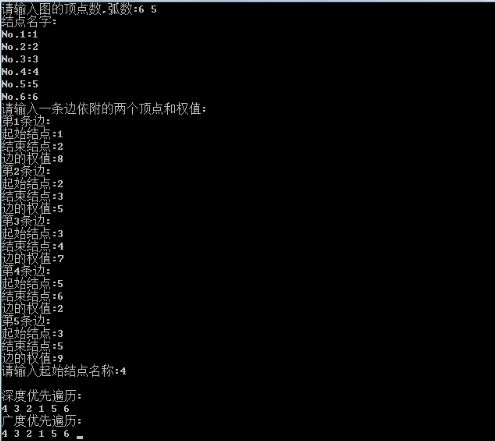
/\* 随机生成代表迷宫地图的数组 \*/  
void makebg(int a,int b){  
 int i,j;  
 int ran;  
 int direc;  
/\* 初始化迷宫地图 \*/  
 for(i=0;i<a;i++)  
 for(j=0;j<b;j++)  
 bg[i][j]=1;

/\* 随机生成迷宫通路 \*/  
 randomize();  
 i=j=0;direc=2;  
 while(1){

bg[i][j]=0;  
 if(i>=M-1&&j>=N-1)break;  
 ran=(int)rand()\*4;  
 if(ran<1){  
 if(direc!=1&&i<a-1){  
 i++;  
 direc=3;  
 }  
 }  
 else if(ran<2){  
 if(direc!=2&&j>0){  
 j--;  
 direc=0;  
 }  
 }  
 else if(ran<3){  
 if(direc!=3&&i>0){  
 i--;  
 direc=1;  
 }  
 }  
 else {  
 if(direc!=0&&j<b-1){  
 j++;  
 direc=2;  
 }  
 }  
 }  
/\* 随机生成迷宫其余部分 \*/  
 for(i=0;i<a;i++)  
 for(j=0;j<b;j++)  
 if(bg[i][j]==1){  
 ran=(int)rand()\*10;  
 if(ran<5)bg[i][j]=0;  
 }  
 for(i=0;i<a;i++)  
 for(j=0;j<b;j++){  
 if(bg[i][j]==1)aa[i][j]=-1;  
 else aa[i][j]=0;  
 }  
}

### 四、运行输出结果：





### 五、调试和运行程序过程中产生的问题及采取的措施：

调试结果可见，在程序存在一定的问题，主要是会导致结点可能会重复输出，但如果改变起始点，运行结果又正常，关键问题还是没能找到问题的所在，但会进一步研究，修改程序使其能正常工作

### 六、对算法的程序的讨论、分析，改进设想，其它经验教训：

图是一种非线性结构，它比线性表更为复杂，在图形结构中，结点之间的关系可以是任意的，图中任意两点数据元素之间都可能相关。

### 七、对实验方式、组织、设备、题目的意见和建议：

图的存储，遍历等在数据结构中是比较难的，虽然前面讲解过线性表、二叉树等结构，但这些都还比较容易，实现过程也不是很难，这次实验虽然做的是图的基本的遍历，但实验操作不是很容易。

最后对老师对我们电子班级同学的付出表示衷心的感谢！