西安财经大学 信息学院

姓名：程驰

学号：1931054041

班级：计本1902

指导教师：李薇

成绩：

数据结构 实验报告

# 实验名称:图的创建和遍历

实验日期：2020年11月30日

**一、实验目的：**

1、熟悉图的存储结构；

2、掌握对图的深度优先和广度优先遍历算法；

**二、实验内容：**

1、键盘输入数据，建立一个有向图的存储结构（可选邻接矩阵或者邻接表）；

2、计算各顶点的度；

2、用递归算法对图进行深度优先和广度优先遍历。

三、实验要求：

编写实现图的建立和遍历的基本算法的函数，并在此基础上设计一个主程序完成如下功能：

⑴键盘输入数据，建立一个有向图（可选邻接矩阵或者邻接表）；

⑵计算各顶点的度；

⑶用递归算法对图进行深度优先和广度优先遍历；

⑷求距离给定顶点v0的路径长度为k的所有顶点。

四、实验步骤：

1. 确定实验环境：Windows10 中文版 、Dev—cpp

2.定义数据结构

**typedef struct ArcNode{**

**int adjvex; //该弧所指向的顶点的位置**

**struct ArcNode \*nextarc; //指向下一条弧的指针**

**}ArcNode;**

**typedef struct VNode{**

**char data; //顶点信息**

**ArcNode \*firstarc; //指向第一条依附该顶点的弧的指针**

**}VNode,AdjList[max];**

**typedef struct{**

**AdjList vertices;**

**int vexnum,arcnum;**

**}ALGraph;**

**typedef struct{**

**int \*base;**

**int \*top;**

**int stacksize;**

**}SqStack;**

3.选择邻接表建立有向图

void CreatGraph(ALGraph &G){

int i,s,e;

ArcNode \*p,\*q;

printf("输入结点数目和边数：");

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

getchar();

printf("输入顶点信息：\n");

for(i=1;i<=G.vexnum;i++){

scanf("%c",&G.vertices[i].data);

G.vertices[i].firstarc=NULL;

}

getchar();

printf("输入一条边依附的起点序号和终点序号：\n");

for(i=1;i<=G.arcnum;i++){

scanf("%d%d",&s,&e);

getchar();

p=(ArcNode \*)malloc(sizeof(ArcNode));

p->nextarc=NULL;

p->adjvex=e;

q=G.vertices[s].firstarc;

if(!q)

G.vertices[s].firstarc=p;

else{

while(q->nextarc){

q=q->nextarc;

}

q->nextarc=p;

}

}

}

4.计算各顶点的度，求距离给定顶点v0的路径长度为k的所有顶点

void FindInDegree(ALGraph G,int indegree[]){

//求顶点入度

ArcNode \*p;

int i;

for(i=1;i<=G.vexnum;i++) //初始化

indegree[i]=0;

for(i=1;i<=G.vexnum;i++){

p=G.vertices[i].firstarc;

while(p){

indegree[p->adjvex]++;

p=p->nextarc;

}

}

}

void TopologicalSort(ALGraph G){

//拓扑排序

ArcNode \*p;

SqStack S;

int count;

int i,k,indegree[max];

FindInDegree(G,indegree); //求入度

InitStack(S);

for(i=1;i<=G.vexnum;i++)// 将入度为零的点压入栈

if(!indegree[i]) Push(S,i);

count=0;

while(!StackEmpty(S)){

Pop(S,i);

printf("%c",G.vertices[i].data);

count++;

for(p=G.vertices[i].firstarc;p;p=p->nextarc){

k=p->adjvex;

if(!(--indegree[k])) Push(S,k);

}

}

if(count<G.vexnum){

printf("该图有回路\n");

exit(0);

}

}

5.用递归算法对图进行深度优先和广度优先遍历；

int visited1[max];

void Dfs(ALGraph G,int v){

ArcNode \*p;

int w;

visited1[v]=1;

printf("%c",G.vertices[v].data);

p=G.vertices[v].firstarc;

while(p){

w=p->adjvex;

if(!visited1[w])

Dfs(G,w);

p=p->nextarc;

}

}

void DfsTraverse(ALGraph G){

//深度优先遍历递归算法

int v;

for(v=1;v<=G.vexnum;v++){

visited1[v]=0;

}

for(v=1;v<=G.vexnum;v++){

if(!visited1[v])

Dfs(G,v);

}

}

void DfsTraverse2(ALGraph G,int v){

//从第v个顶点进行深度优先遍历（非递归）

int visited2[max],n,i;

for(i=1;i<=G.vexnum;i++)

visited2[i]=0;

ArcNode \*p;

SqStack S;

InitStack(S);

visited2[v]=1;

printf("%c",G.vertices[v].data);

Push(S,v);

while(!StackEmpty(S)){

GetTop(S,v);

p=G.vertices[v].firstarc;

while(p)

if(visited2[p->adjvex])

p=p->nextarc;

else{

printf("%c",G.vertices[p->adjvex].data);

visited2[p->adjvex]=1;

Push(S,p->adjvex);

p=G.vertices[p->adjvex].firstarc;

}

if(!StackEmpty(S)){

Pop(S,n);

}

}

}

int q[max],f=0,r=0;

int visited2[max];

void BfsTraverse(ALGraph G,int s){

//广度优先遍历的递归算法

ArcNode \*p;

if(!visited2[s]){

printf("%c",G.vertices[s].data);

visited2[s]=1;

p=G.vertices[s].firstarc;

while(p){

if(!visited2[p->adjvex])

q[r++]=p->adjvex;

p=p->nextarc;

}

}

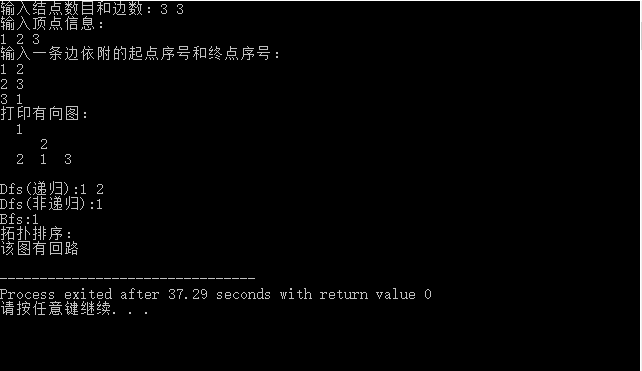
while(f<r){

BfsTraverse(G,q[f++]);

}

}

五、实验结果：



六、实验总结：

在输入顶点数和边数时，要添加空格，不然程序默认为一个数据。

打印有向图时，直接按照顶点的度打印了顶点，不够形象，这是需要改进的。