

**数据结构实验报告**

**学 院** 信息工程学院

**专 业** 计算机类（大类招生）

**班 级** 2016240210

**姓 名** 瞿强鑫

**学 号** 2016902094

**指导教师** 史昕

**完成时间** 2017.6.15

1. **实验目的**

1.掌握图的逻辑结构；

2.验证图的邻接表存储结构；

3.验证邻接表的存储及遍历操作；

**二、实验设备（环境）及要求**

电脑一台，VC++环境下运行。

1. **实验内容（步骤）与方法**

内容：1.建立一个有向图邻接表存储结构

2.对建立的有向图进行深度优先遍历

3.对建立的有向图进行广度优先遍历

方法： 图的邻接表是一种顺序存储与链接存储结合的方法。先构建顶点表和边表分 别存放顶点数据和边，再将两个表按照图连接起来。通过深度优先遍历和广度优

先遍历两个算法实现对有向图的遍历操作。

//定义边表结点

**struct ArcNode**

{

int adjvex;

ArcNode <DataType> \* next;

};

//建立顶点表结点

**template<class DataType>**

**struct VertexNode**

{

DataType vertex;

ArcNode <DataType> \* firstedge;

};

const int MaxSize=5;

//定义邻接表类

**template <class DataType>**

**class ALGraph**

{

public:

ALGraph(DataType a[],int n,int e);

~ALGraph(){};

void DFSTraverse(int v);

void BFSTraverse(int v);

void newv();

private:

VertexNode <DataType> adjlist[MaxSize];

int vertexNum,arcNum;

int visited[MaxSize];

};

//邻接表构造函数

**template<class DataType>**

**ALGraph<DataType>::ALGraph(DataType a[],int n,int e)**

{

vertexNum=n;arcNum=e;

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

{

adjlist[i].vertex=a[i];

adjlist[i].firstedge=NULL;

}

int m,b;

ArcNode <DataType> \* s;

for(int k=0;k<arcNum;k++)

{

cout<<"输入边依附两顶点编号：";

cin>>m>>b;

s=new ArcNode<DataType>;s->adjvex=b;

s->next=adjlist[m].firstedge;

adjlist[m].firstedge=s;

}

}

//深度优先遍历

**template <class DataType>**

**void ALGraph<DataType>::DFSTraverse(int v)**

{

cout<<adjlist[v].vertex<<" ";

visited[v]=1;

ArcNode<DataType> \* p;

p=new ArcNode<DataType>;

p=adjlist[v].firstedge;

int j;

while(p!=NULL)

{

j=p->adjvex;

if(visited[j]==0) DFSTraverse(j);

p=p->next;

}

}

//广度优先遍历

**template <class DataType>**

**void ALGraph<DataType>::BFSTraverse(int v)**  {

int front=-1;

int rear=-1;

int Q[MaxSize];

int j;

ArcNode <DataType> \* p;

p=new ArcNode<DataType>;

cout<<adjlist[v].vertex<<" ";visited[v]=1;Q[++rear]=v;

while(front!=rear)

{

v=Q[++front];

p=adjlist[v].firstedge;

while(p!=NULL)

{

j=p->adjvex;

if(visited[j]==0)

{

cout<<adjlist[j].vertex<<" ";

visited[j]=1;

Q[++rear]=j;

}

p=p->next;

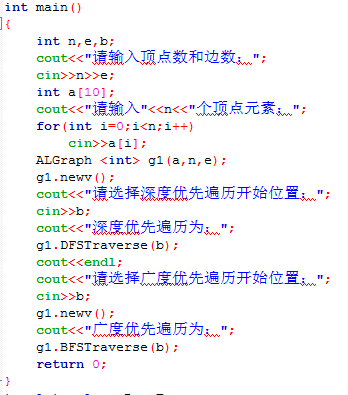
}

}

}

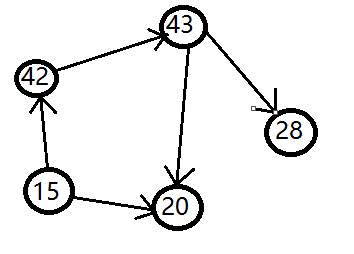
**四、实验结果与分析**

程序main函数如图1：



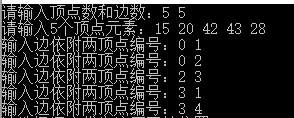
**图1**

程序运行，实现建立如图2所示有向图：



**图2**

建立过程如图3，建立成功：



**图3**

从第一个顶点7开始深度优先遍历，结果如图4，达到目标：



**图4**

从第二个顶点5开始广度优先遍历，结果如图5，达到目标：



**图5**

**五、实验心得与体会**

由于两次实验连着，第一个做完后第二个实验有向图邻接表显得略微仓促，写的很急，对程序的理解不够深刻，需要下来再看看。

然后说程序编程的时候，在深度优先遍历部分，我又一次深刻领教了递归算法的威力，对这个算法的领悟也加深了。而广度优先遍历部分，通过两个while的嵌套循环，我们能够较为容易的实现对所有连接着的结点进行遍历。

其次，是在使用有向图邻接表的时候，我们很显而易见的发现了存入数据的困难复杂性，需要一对一地输入，并且将数据在数组对应下标的输入必须准确把握，才能实现有向图的构建。这应该还有解决的办法吧，只是我们现在能力不够无法达到。

最后，希望自己数据结构能完美收官，这是最后一次实验，希望在期末取得好的成绩，争取不挂科，理解掌握这门学科上面学到的知识！