**实验报告（七）**

**图**

班级： 2015211307 姓名：尹子长 学号：2015211912 分工： 写文档

班级： 2015211307 姓名：赵如茵 学号：2015211334 分工： 算法设计与写代码

1. 需求分析

已知某图是边带权(权值为正数)的有向无环图,采用邻接表存储,求出图中距离最远的两个结点。

1. 概要设计

先遍历，在该有向无环图中寻找入度为0的节点并在bool数组中表示为True, bool数组初始化为全True

再用递归的方式寻找每种路径的最后一个节点并计算路径长度

在带权有向无环图中寻找距离最远的两个点，起始节点入度必为0，结束节点出度必为0

1. 详细设计

数据结构：

图：

typedef struct {

VNode AdjList[MAX\_VERTEX\_NUM];//顶点数组

int vexnum,arcnum;//图的当前顶点数和弧数

int kind;//图的种类标志

}ALGraph;

结点：

typedef struct VNode{

VertexType data;//顶点信息

ArcNode \*firstarc;//指向第一条依附于该顶点的弧的指针

}VNode;

边：

typedef struct ArcNode {

int adjvex;//该弧指向的顶点位置

struct ArcNode \*nextarc;//指向下一条弧的指针

InfoType info;//原来是该弧相关信息的指针，这里改为一个int型变量表示该弧的权值

}ArcNode;

函数模块：

void isHead (int n,VNode l[],bool \*b);//寻找入度为0的节点

void findFurthest (VNode l[],int x,int prelen,int \*head,int \*tail,int \*maxlen,int ifHead); 寻找最长路径，l顶点数组，x当前节点在数组中的位置，prelen在到达此节点之前累计走

过的路径长度，tail最长路径尾节点所在位置，maxlen最长路径,ifhead储存当前正在计算

的路径的起始节点，注意此处head，tail，ifhead都是节点在数组中的位置，输出时数值需要+1s ←v ←

void isHead (int n,VNode l[],bool \*b){

ArcNode \*p;

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

p=l[i].firstarc;

while(p!=NULL){

\*(b+p->adjvex)=false;//将入度不为0的数组的对应的布尔变量置为false

p=p->nextarc;

}

}

}

void findFurthest (VNode l[],int x,int prelen,int \*head,int \*tail,int \*maxlen,int ifHead){

ArcNode \*p;

if(l[x].firstarc==NULL){//若此时指向的节点出度为0，将累计的路径长度与maxlen比较，决定是否要修改

if(prelen>\*maxlen){

\*maxlen=prelen;

\*tail=x;

\*head=ifHead;

}

else;

}

else{

p=l[x].firstarc;

while(p!=NULL){

findFurthest(l,p->adjvex,prelen+(p->info),head,tail,maxlen,ifHead);

p=p->nextarc;

}

}

}

1. 调试分析报告

算法时间复杂度：O（en2）

空间复杂度: O(1)

1. 用户使用说明

Step 1：输入图中节点的数目；

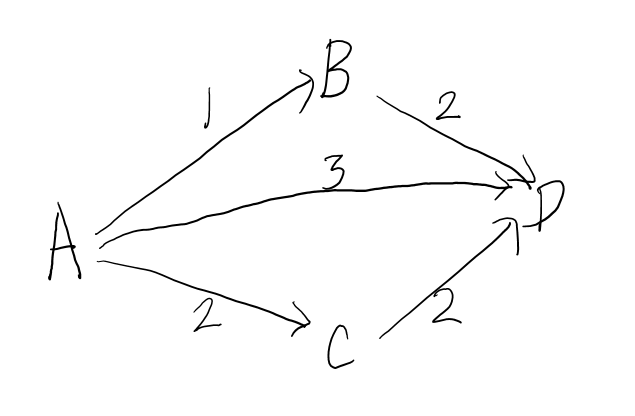
Step 2：输入从一个节点出发的弧的条数；

依次输入这些弧的指向和权值；

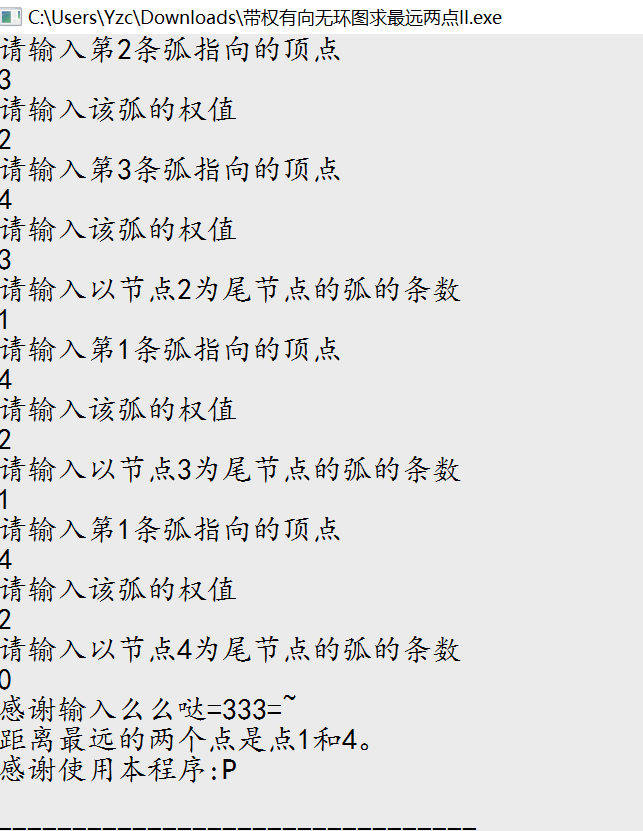
Step 3：输入下一节点，转向2；

Step 4：输出结果；

1. 测试结果

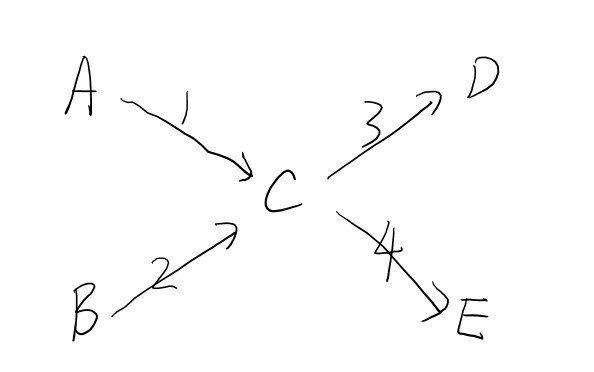
Case 1:

输出样式：

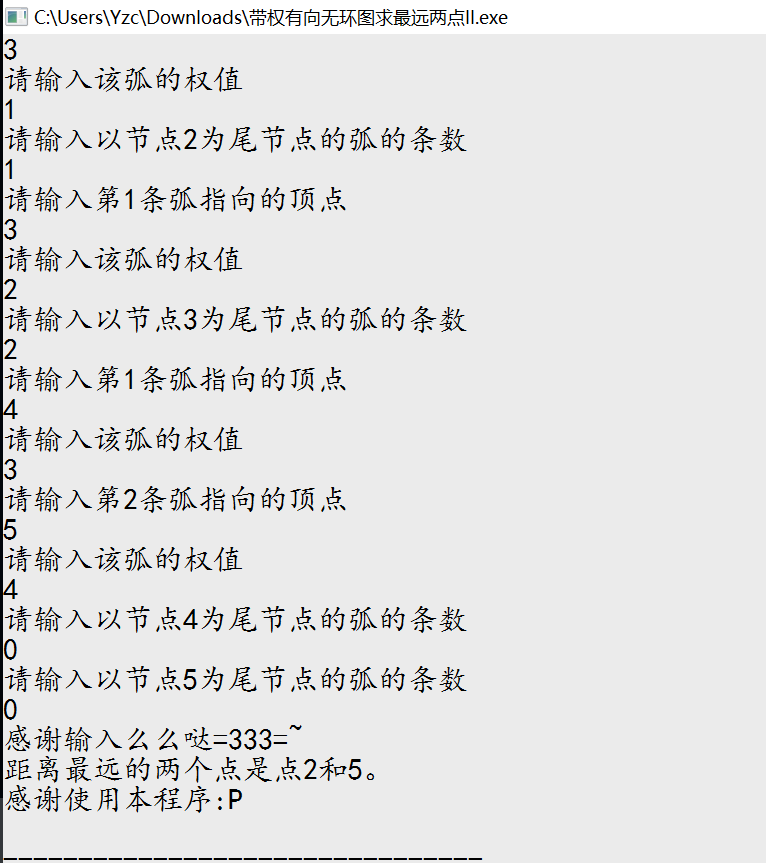


与预期一致。

Case 2:



输出样式：



与预期一致。