**题目11：关键路径寻找**

对于给定的一个工程施工图，该图以边为单位从键盘输入，编写能够找出该图的关键路径的程序。

**代码：**

#include"stdio.h"

#include"stdlib.h"

typedef struct Vexnode //顶点表

{

int adjvex; //邻接域

int dut; //记录权值

struct Vexnode \* next;

}vexnode;

typedef struct Enode

{

int indegree; //记录入度

int vertex; //顶点

int ee,el; //记录最迟开始时间和最早结束时间

struct Vexnode \* link;

}enode;

void print(enode dig[],int first,int len)

{

int i,j;

static int top=0,list[50];

vexnode \* q;

i=first;

q=dig[i].link;

list[top]=dig[i].vertex;

top++; //进栈

if (q==NULL)

{

printf("%d",list[len]);

for (i=1+len;i<top;i++) printf("->%d",list[i]);

printf("\n");

}

while (q!=NULL)

{

j=q->adjvex;

if (dig[j].ee==dig[j].el)

{

list[top]=dig[j].vertex;

print(dig,j,len);

////////

}

q=q->next;

}

top--;////////

}

int toposort(enode dig[],int e\_n,int stacktp[])

{

int top=0,bottom=0,i,j,len=0;

vexnode \*q;

for (i=1; i<=e\_n; i++)

{

if (dig[i].indegree==0) //入度为0则进栈

{

stacktp[top]=i;

top++;

}

len=top;

}

while (top>bottom) //拓扑排序

{

i=stacktp[bottom];

q=dig[i].link;

bottom++;

while (q!=NULL)

{

j=q->adjvex;

dig[j].indegree--; //入度--

if (dig[i].ee+q->dut>dig[j].ee) //求一下最早开始时间

dig[j].ee=dig[i].ee+q->dut;

if (dig[q->adjvex].indegree==0) //入度为0 则进栈

{ stacktp[top]=q->adjvex; top++; }

q=q->next;

}

}

if (top==e\_n) //表示没有环存在，则求出最迟结束时间

{

for (i=1; i<=e\_n; i++)

dig[i].el=dig[stacktp[top-1]].ee; //先初始化一下

bottom=0;

while (top>bottom) //栈非空

{

top--; //从最后的一个事件开始倒推

i=stacktp[top];

q=dig[i].link;

while (q!=NULL)

{

j=q->adjvex;

if (dig[j].el-q->dut<dig[i].el)

dig[i].el=dig[j].el-q->dut;

q=q->next;

}

}

for (i=0; i<len; i++)

print(dig,stacktp[i],i);

return 0;

}

else return 1;

}

int main()

{

enode dig[20]; //顶点表数组

vexnode \*q; //邻接点链表

char ch;

int e\_n,v\_n,m,i,j,u,v,stacktp[20];

printf("------------关键路径------------\n\n");

printf("请输入顶点个数和边的条数：");

scanf("%d%\*c%d",&e\_n,&v\_n);

for (i=1; i<=e\_n; i++) //初始化

{

dig[i].indegree=0;

dig[i].link=NULL;

dig[i].vertex=i;

dig[i].ee=dig[i].el=0;

}

for (i=0; i<v\_n; i++)

{

printf("请输入NO.%d条边的起始点以及该边的权：\n",i+1);

//printf("输入NO.%d")

scanf("%d%\*c%d%\*c%d",&u,&v,&j); //读入各边以及边的权值

q=(Vexnode \*)malloc(sizeof(Vexnode));

dig[v].indegree++;

q->adjvex=v;

q->dut=j;

q->next=dig[u].link;//将 q结点 放入u 邻接链表中

dig[u].link=q; //将 q结点 放入u 邻接链表中

}

m=toposort(dig,e\_n,stacktp); //拓扑排序

if (m) printf("图中存在环，不存在关键路径\n");

else

{

printf("需要各点明细查询吗[y/n]");

scanf("\n%c",&ch);;

if (ch=='y'||ch=='Y')

for (i=0; i<e\_n; i++)

printf("%d (%d %d) \n",stacktp[i],dig[stacktp[i]].ee,dig[stacktp[i]].el);

}

return 0;

}

**运行结果：**

