

```
1  ///////////////////////////////////////////////////
2  //2005级信息安全2班 张文
3  // 200532530040
4  ///////////////////////////////////////////////////
5  #include<stdio.h>
6  #include<stdlib.h>
7  #include<string.h>
8
9  #define MAXV 10//最大顶点数
10 ///////////////////////////////////////////////////邻接表
11 typedef struct ANode
12 {
13     int adjvex;
14     struct ANode *nextarc;
15 }ArcNode;
16
17 typedef struct VNode_type
18 {
19     int node;
20     ArcNode *firstarc;
21 }VNode;
22
23 typedef struct VNode_type AdjList[MAXV];
24
25 typedef struct
26 {
27     AdjList adjlist;
28     int n,e;
29 }ALGraph;
30
31 ///////////////////////////////////////////////////函数申明
32 void initGraph(ALGraph *);
33 void travelALG(ALGraph *,int);
34
35 ///////////////////////////////////////////////////主程序
36 int main()
37 {
38     ALGraph MyGraph;
39     initGraph(&MyGraph);
40     travelALG(&MyGraph,0);
41     return 0;
42 }
43
44 ///////////////////////////////////////////////////初始化建图
45 void initGraph(ALGraph *ThisGraph)
46 {
47     int Node[2];
48     int i,j;
49     ArcNode *ThisNode,*PreNode;
50
51     printf("输入节点总数: ");
52     scanf("%d",&(ThisGraph->n));
53
54     for(i=0;i<MAXV;i++)
55     {
56         ThisGraph->adjlist[i].firstarc=NULL;
57         ThisGraph->adjlist[i].node=i;
58     }
59
60     i=0;
61     while(1)
62     {
63         printf("输入每条边的两个端点,用空格间隔(输入两个-1结束输入): ");
64         scanf("%d%d",&Node[0],&Node[1]);
65         if(Node[0]==-1 && Node[1]==-1)break;
66
67         for(j=0;j<2;j++)
```

```

68         {
69             PreNode=ThisGraph->adjlist[Node[j]].firstarc;
70             ThisNode=(ArcNode *)malloc(sizeof(ArcNode));
71             ThisNode->adjvex=Node[(j+1)%2];
72             ThisNode->nextarc=NULL;
73
74             if(PreNode==NULL)ThisGraph->adjlist[Node[j]].firstarc=ThisNode;
75             else
76             {
77                 while(PreNode->nextarc!=NULL)PreNode=PreNode->nextarc;
78                 PreNode->nextarc=ThisNode;
79             }
80         }
81
82         i++;
83     }
84     ThisGraph->e=i;
85 }
86
87 void travelALG(ALGraph *ThisGraph,int StartNode)//非递归深度优先遍历
88 {
89     int VisitedFlag[MAXV];
90     ArcNode *ArcStack[MAXV];
91     VNode *NodeStack[MAXV];
92     int i,j;
93     ArcNode *ThisArc;
94     VNode *ThisNode;
95
96     memset(VisitedFlag,0,sizeof(VisitedFlag));
97     for(i=0;i<MAXV;i++)
98     {
99         ArcStack[i]=NULL;
100        NodeStack[i]=NULL;
101    }
102    ThisNode=&(ThisGraph->adjlist[StartNode]);
103    ThisArc=ThisNode->firstarc;
104    i=-1;j=0;
105    while(j<ThisGraph->n)
106    {
107        if(ThisArc!=NULL)
108        {
109            NodeStack[++i]=ThisNode;
110            ThisArc=ThisNode->firstarc;
111            VisitedFlag[ThisNode->node]=1;
112            j++;
113        }
114        else if(i>=0)//回溯
115        {
116            ArcStack[ThisNode->node]=NULL;
117            NodeStack[i]=NULL;
118            ThisNode=NodeStack[--i];
119            ThisArc=ArcStack[ThisNode->node];
120        }
121        else break;
122        printf("Visited Node %d\n",ThisNode->node);
123        while(ThisArc!=NULL)
124            if(VisitedFlag[ThisArc->adjvex])ThisArc=ThisArc->nextarc;
125            else
126            {
127                ArcStack[ThisNode->node]=ThisArc;
128                ThisNode=&(ThisGraph->adjlist[ThisArc->adjvex]);
129                break;
130            }
131    }
132 }

```