实验题目：

深度优先搜索 C语言

#define NULL 0

#define N 4 /\* N表示图的顶点最大个数 \*/

typedef struct ArcNode{

int adjvex;

struct ArcNode \*nextarc;

}ArcNode;

typedef struct VNode{

int data; /\* 注意图的顶点信息是整型 \*/

ArcNode \*firstarc;

} VNode,Adjlist[N]; /\* 最多N个顶点 \*/

int visited[N]={0}; /\* 全局变量，初值是0，避免重复遍历 \*/

typedef struct{

Adjlist vertices;

int vexnum,arcnum;

}ALGraph; /\*邻接表\*/

void Create(ALGraph \*G){ /\* 创建一个有向图 \*/

int i,j,k;

ArcNode \*s;

printf("Please input vexnum and arcnum:");

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

/\* 输入顶点数(最多N)和边数 \*/

printf("Please input each vex message,vex message is int and from 0 start :\n");

for(i=0;i<G->vexnum;i++){

/\* 输入每个顶点信息，注意顶点类型是整数，顶点编号从0开始

顶点信息是顶点编号\*/

scanf("%d",&G->vertices[i].data);

G->vertices[i].firstarc=NULL;

}

printf("Please input the edge,edge message is int,int:\n");

for(k=0;k<G->arcnum;k++){

/\* 输入每个边信息，注意录入边的形式，例如顶点0，1之间有边， \*/

/\* 录入正确形式是：0,1 \*/

scanf("%d,%d",&i,&j);

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

s->adjvex=j;

s->nextarc=G->vertices[i].firstarc;

G->vertices[i].firstarc=s;

}

}

void DFS(ALGraph G,int v){

/\* 从顶点v出发进行深度优先遍历,注意参数G的类型 \*/

ArcNode \*w;

visited[v]=1;

printf("%3d",G.vertices[v].data);

for(w=G.vertices[v].firstarc;w;w=w->nextarc)

if(!visited[w->adjvex])

DFS(G, w->adjvex);

}

void DFSGraph(ALGraph G){ /\* 尝试从各点出发DFS \*/

int i;

for(i=0;i<G.vexnum;i++)

if(!visited[i]) {

printf("\nFrom %d DFS is:",i);

DFS(G,i); /\* 从i出发深度遍历 \*/

}

}

void main(){

ALGraph G;

Create(&G);

DFSGraph (G);

getch();

}

运行结果：

