

[实验题目五] 图的应用

一、实验目的

- 1、理解图的结构特点。
- 2、掌握存储图的方法（邻接表）。
- 3、掌握图的遍历方法和基本应用。

二、预习内容

- 1、教材 7.2.2、7.3 和 7.5.1；
- 2、递归算法

三、实验内容

图的应用有以下 2 个实验内容：

1. 建立一个有向图，对其实施深度优先遍历

(1) 建立一个有向图

- A、定义图的数据结构，要求用邻接表；
- B、定义一个函数建立一个有向图。

(2) 对上边有向图进行深度优先遍历

具体算法思想可参考教材算法 7.4 和算法 7.5

(3) 图的程序编制上较难，所以实验内容 1 给出全部参考算法，做验证性实验。

```
无标题 - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

#include <stdlib.h>
#define N 20 // N表示图的顶点个数
typedef enum{DG, DN, UDG, UDN} GraphKind; //{有向图, 有向网, 无向图, 无向网}
typedef struct ArcNode{
    int adjvex; //该弧指向顶点位置
    struct ArcNode *nextarc; //指向下一条弧的指针
}ArcNode;
typedef struct VNode{
    int data; // 注意图的顶点信息是整型
    ArcNode *firstarc; //指向第一条邻接该顶点 (以该顶点为弧尾) 的弧的指针
} VNode, Adjlist[N]; //最多N个顶点
typedef struct{
    Adjlist vertices;
    int vexnum, arcnum; //分别存图的顶点数和弧数
    GraphKind kind; //图的种类标志
}ALGraph; //邻接表
void CreateDG(ALGraph *G){ // 创建一个有向图
    int i, j, k;
    ArcNode *s;
    G->kind=DG; //有向图
    printf("分别输入顶点数和弧数:");
    scanf("%d%d", &G->vexnum, &G->arcnum);
    printf("输入顶点信息, 从0开始, 编号连续 :\\n");
    for(i=0; i<G->vexnum; i++){
        scanf("%d", &G->vertices[i].data);
        G->vertices[i].firstarc=NULL;
    }
    printf("输入%d条弧, 弧的信息格式: 顶点信息, 顶点信息\\n", G->arcnum);
    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;
    }
}
```

```
无标题 - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

int visited[N]={0}; //全局变量, 初值是0, 避免重复遍历
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); //输出顶点信息, 在屏幕占3个宽度
    //找顶点v的邻接点, 如果未被访问, 则从相应的邻接点出发深度遍历
    for(w=G.vertices[v].firstarc; w; w=w->nextarc){
        if(!visited[w->adjvex])
            DFS(G, w->adjvex); //递归
    }
}
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("\\n从顶点 %d 出发, 深度遍历的结果:", i);
            DFS(G, i); // 从i出发深度遍历
            putchar('\\n'); //换行
        }
    }
}
void main(){
    ALGraph G;
    CreateDG(&G); //创建一个有向图
    DFSGraph(G); //对图G深度优先遍历
}
```

例：输入数据依次是：4 3 回车

0 1 2 3 回车

0,1 0,2 2,3 回车

要求：在实验报告上画出按上面输入数据所创建的有向图的邻接表；

2. 求图中结点度数的应用

(1) 求图中某个顶点（地址编号为 i）的出度

```
int OutputDegree(ALGraph G, int i){  
    /*返回编号是 i 的顶点的出度*/  
    .....  
}
```

(2) 求图中某个顶点（地址编号为 i）的入度(选做)

```
int InDegree(ALGraph G, int i){  
    /*返回编号是 i 的顶点的入度*/  
    .....  
}
```

3. 图的广度优先遍历(选做)

四、实验报告要求

- 1、写好实验题目、学号、姓名和实验日期；
- 2、写上实验的内容及程序源代码；
- 3、写清程序的输入实例和输出结果，并画出相应的邻接表；
- 4、写个实验小结。