

上机题:

☆1. 要求用邻接表实现无向图。根据一个无向图的顶点和无向边信息构造该无向图。输出该无向图的深度优先和广度优先序列。

☆2. 已知无向连通图 G 由顶点集 V 和边集 E 组成, $|E| > 0$, 当 G 中度为奇数的顶点个数为不大于 2 的偶数时, G 存在包含所有边且长度为 $|E|$ 的路径 (称为 EL 路径)。设图 G 采用邻接矩阵存储, 类型定义如下:

```
typedef struct{           //图的定义
    int numVertices,numEdges;
                                //图中实际的顶点数和边数
    char VerticesList[MAXV];
                                //顶点表。MAXV 为已定义常量
    int Edge[MAXV][MAXV];
                                //邻接矩阵
}MGraph;
```

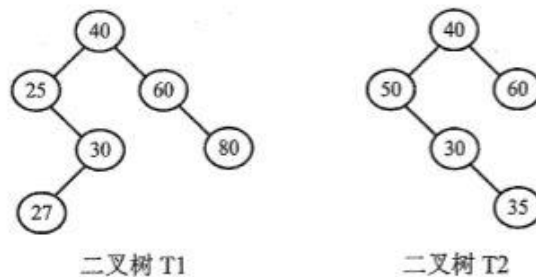
请设计算法 `int IsExistEL(MGraph G)`, 判断 G 是否存在 EL 路径, 若存在, 则返回 1, 否则返回 0。要求:

- 1) 给出算法的基本设计思想。
- 2) 根据设计思想, 采用 C 或 C++ 语言描述算法, 关键之处给出注释。
- 3) 说明你所设计算法的时间复杂度和空间复杂度。

☆3. 已知非空二叉树 T 的结点值均为正整数, 采用顺序存储方式保存, 数据结构定义如下:

```
typedef struct {           // MAX_SIZE 为已定义常量
    int SqBiTNode[MAX_SIZE]; // 保存二叉树结点值的数组
    int ElemNum;             // 实际占用的数组元素个数
} SqBiTree;
```

T 中不存在的结点在数组 `SqBiTNode` 中用 -1 表示。例如, 对于下图所示的两棵非空二叉树 T_1 和 T_2 ,



T_1 的存储结果如下:

<code>T1.SqBiTNode</code>	40	25	60	-1	30	-1	80	-1	-1	27		
---------------------------	----	----	----	----	----	----	----	----	----	----	--	--

`T1.ElemNum = 10`

T_2 的存储结果如下:

<code>T2.SqBiTNode</code>	40	50	60	-1	30	-1	-1	-1	-1	-1	35	
---------------------------	----	----	----	----	----	----	----	----	----	----	----	--

`T2.ElemNum = 11`

请设计一个尽可能高效的算法, 判定一棵采用这种方式存储的二叉树是否为二叉搜索树, 若是, 则返回 `true`, 否则, 返回 `false`。要求:

- 1) 给出算法的基本设计思想。
- 2) 根据设计思想, 采用 C 或 C++ 语言描述算法, 关键之处给出注释。