

数据结构（C语言版）（第2版）

树和二叉树

二叉树的性质和存储结构

主讲教师：汪红松



教学内容 Contents

1 树和二叉树的定义

2 二叉树的性质和存储结构

3 遍历二叉树

4 线索二叉树

5 树和森林

6 哈夫曼树及其应用

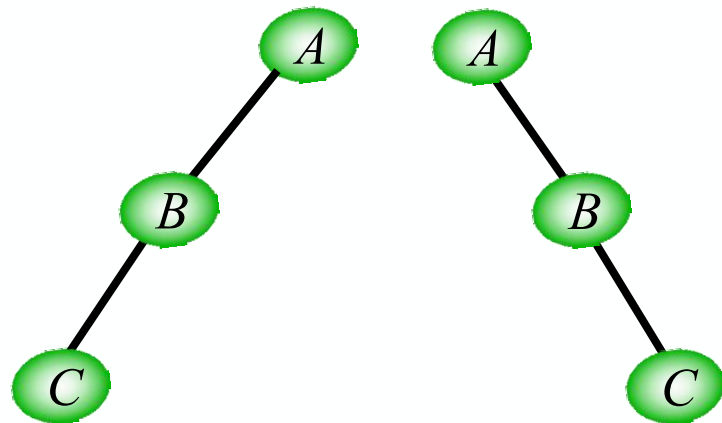
▶▶▶ 一、特殊形态的二叉树

1. 斜树

(1) 所有结点都只有左子树的二叉树称为**左斜树**;

(2) 所有结点都只有右子树的二叉树称为**右斜树**;

(3) 左斜树和右斜树统称为**斜树**。



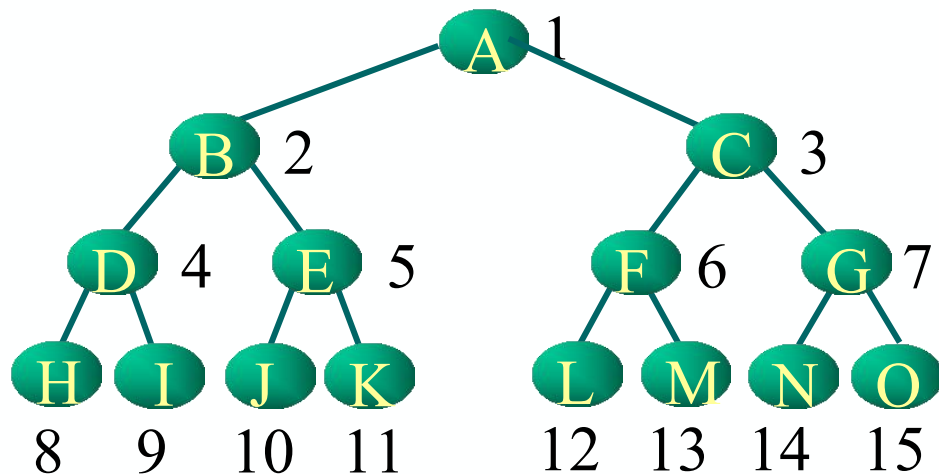
斜树的特点:

- ① 在斜树中，每一层只有一个结点;
- ② 斜树的结点个数与其深度相同。

▶▶▶ 一、特殊形态的二叉树

2. 满二叉树

在一棵二叉树中，如果所有分支结点都存在左子树和右子树，并且所有叶子都在同一层上。

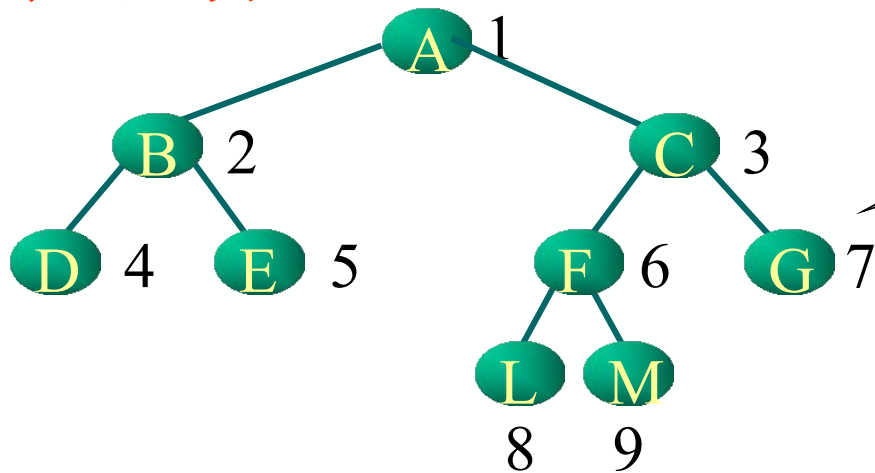


满二叉树的特点：

- ① 叶子只能出现在最下一层；
- ② 只有度为0和度为2的结点。

一、特殊形态的二叉树

2. 满二叉树



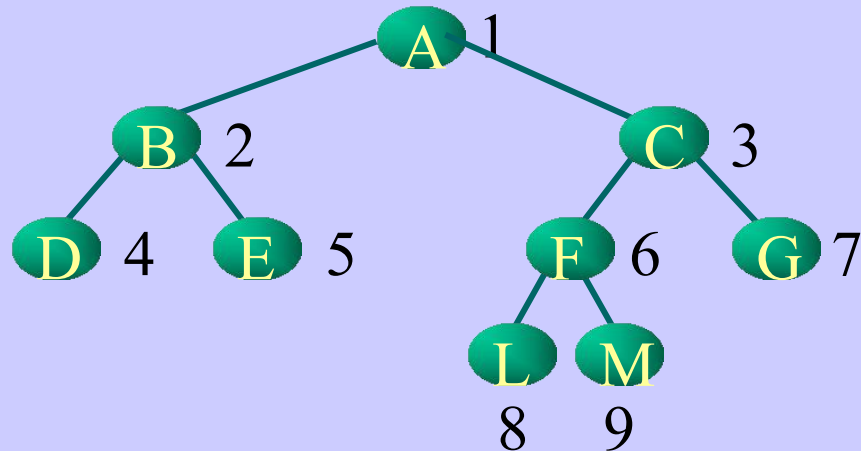
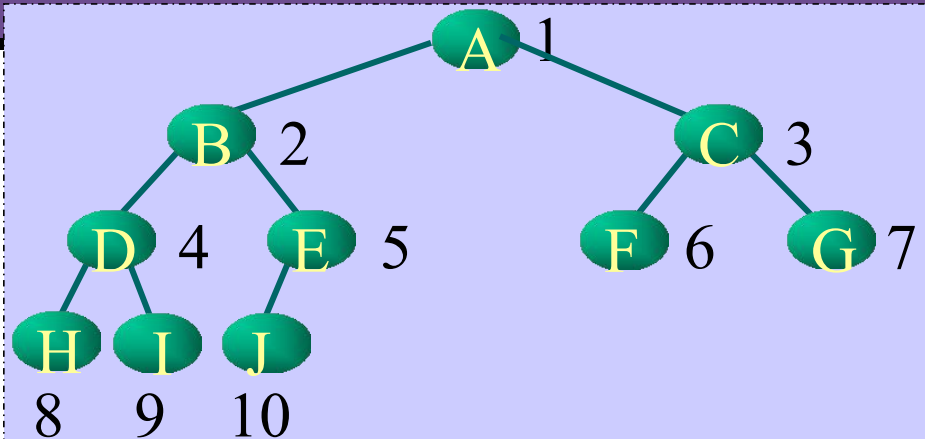
不是满二叉树，虽然所有分支结点都有左右子树，但叶子不在同一层上。

满二叉树在同样深度的二叉树中**结点**个数最多；
满二叉树在同样深度的二叉树中**叶子结点**个数最多。

一、特殊形态的二叉树

3. 完全二叉树

对一棵具有 n 个结点的二叉树按层序编号，如果编号为 i ($1 \leq i \leq n$) 的结点与同样深度的满二叉树中编号为 i 的结点在二叉树中的位置完全相同。



▶▶▶ 一、特殊形态的二叉树

3. 完全二叉树

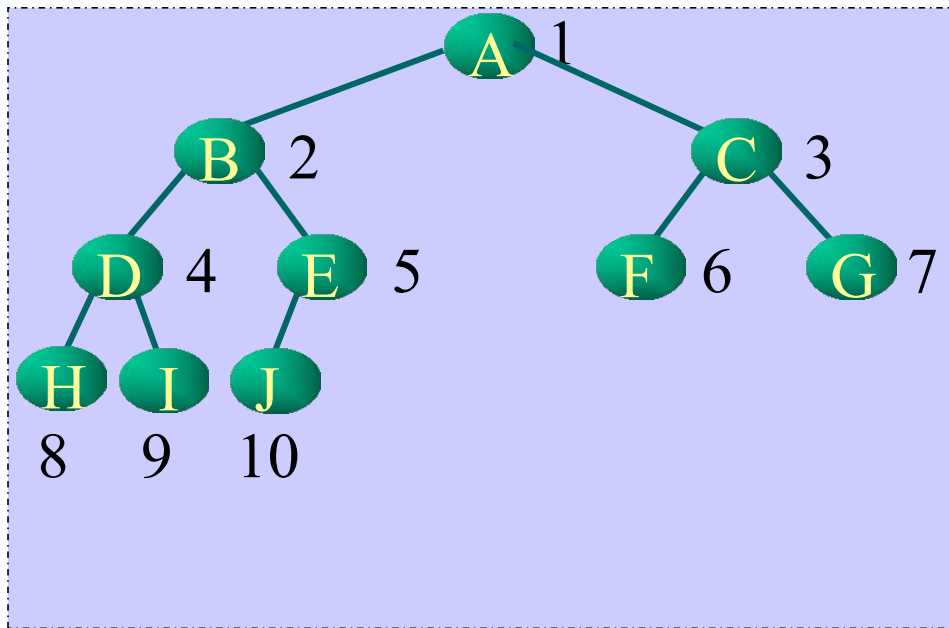
完全二叉树的特点

(1) 叶子结点只能出现在最下两层且最下层的叶子结点都集中在二叉树的左面；

(2) 完全二叉树中如果有度为1的结点，只可能有一个，且该结点只有左孩子。

(3) 深度为 k 的完全二叉树在 $k-1$ 层上一定是满二叉树。

(4) 在同样结点个数的二叉树中，完全二叉树的深度最小。



二、二叉树的性质和存储结构

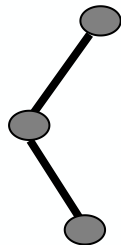
性质1: 在二叉树的第 i 层上至多有 2^{i-1} 个结点

注意：第 i 层上至少有1个结点

性质2: 深度为 k 的二叉树至多有 2^k-1 个结点

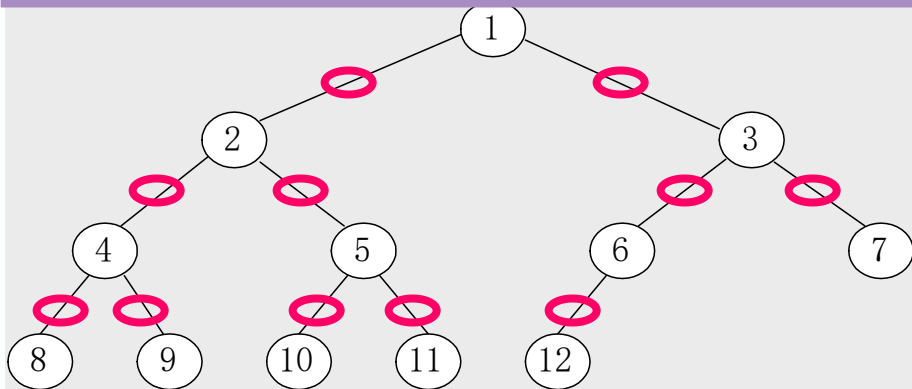
注意：深度为 k 时至少有 k 个结点

深度为 k 且具有 2^k-1 个结点的二叉树**一定是**满二叉树，
深度为 k 且具有 k 个结点的二叉树**不一定**是斜树。

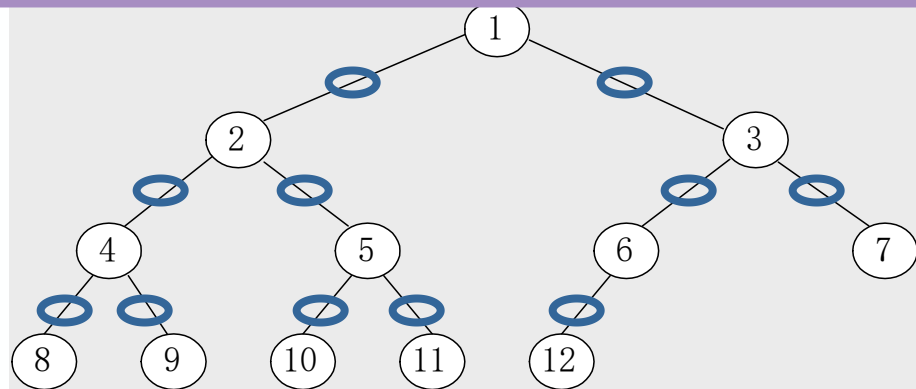


二、二叉树的性质和存储结构

性质3: 对于任何一棵二叉树，若2度的结点数有 n_2 个，则叶子数 n_0 必定为 $n_2 + 1$ （即 $n_0 = n_2 + 1$ ）



$$B = n - 1$$

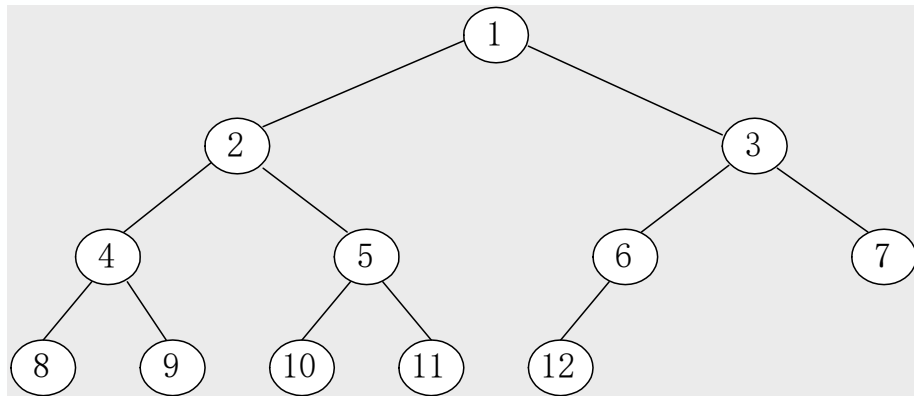


$$B = n_2 \times 2 + n_1 \times 1$$

$$n = n_2 \times 2 + n_1 \times 1 + 1 = n_2 + n_1 + n_0$$

二、二叉树的性质和存储结构

性质4: 具有 n 个结点的完全二叉树的深度必为 $\lfloor \log_2 n \rfloor + 1$



因此, $2^{k-1} \leq n < 2^k$
对不等式取对数, 有:

$$k-1 \leq \log_2 n < k$$

由于 k 是整数, 故必有 $k = \lfloor \log_2 n \rfloor + 1$ 。

最少结点数

k-1层

$$2^{k-1} - 1$$

k层

$$2^k - 1$$

最多结点数

二、二叉树的性质和存储结构

性质5：对一棵具有 n 个结点的完全二叉树中从1开始按层序编号，则对于任意的序号为 i ($1 \leq i \leq n$) 的结点（简称为结点 i ），有：

（1）如果 $i > 1$ ，则结点 i 的双亲结点的序号为 $i/2$ ；如果 $i = 1$ ，则结点 i 是根结点，无双亲结点。

（2）如果 $2i \leq n$ ，则结点 i 的左孩子的序号为 $2i$ ；
如果 $2i > n$ ，则结点 i 无左孩子。

（3）如果 $2i+1 \leq n$ ，则结点 i 的右孩子的序号为 $2i+1$ ；如果 $2i+1 > n$ ，则结点 i 无右孩子。

▶▶▶ 三、二叉树的顺序存储

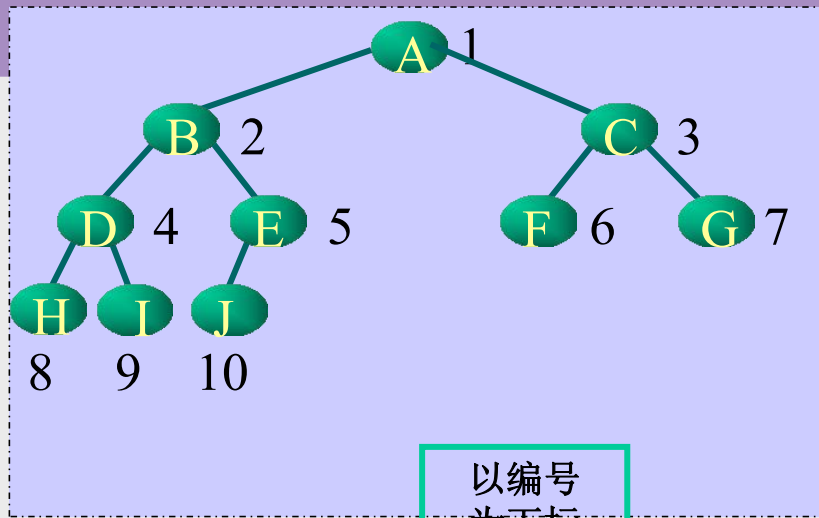
二叉树的顺序存储结构就是用一维数组存储二叉树中的结点，并且结点的**存储位置**（下标）应能体现结点之间的**逻辑关系**——父子关系。

```
#define MAXTSIZE 100 //二叉树的最大结点数
typedef TElemType SqBiTree [MAXTSIZE];
//0号单元存储根结点

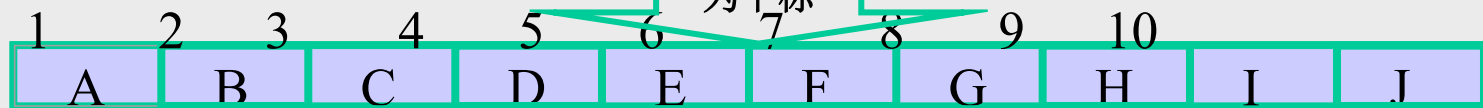
SqBiTree bt ;
```

三、二叉树的顺序存储

实现：按**满二叉树**的结点层次编号，依次存放二叉树中的数据元素。



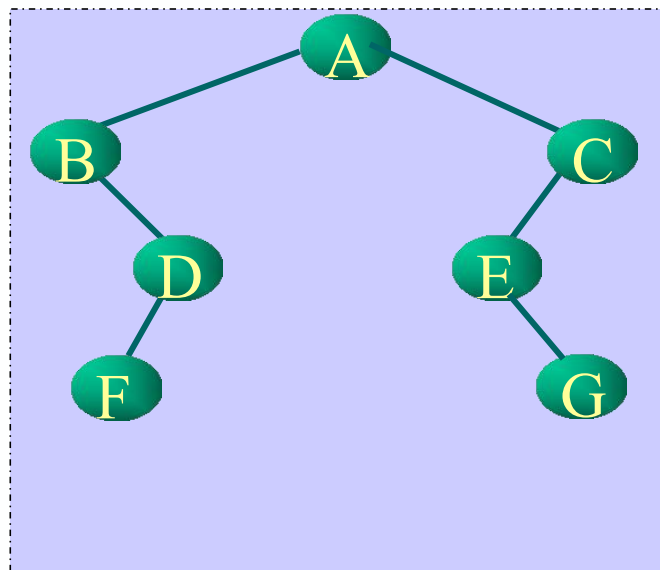
数组下标



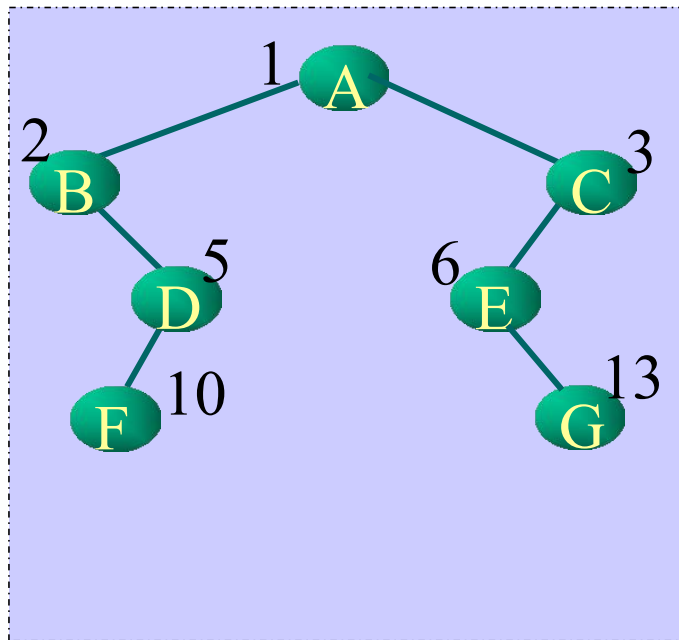
完全二叉树和**满二叉树**中结点的层序编号可以唯一地反映出结点之间的逻辑关系。

三、二叉树的顺序存储

存储
二叉树的顺序



按照完全
二叉树编号



数组下标

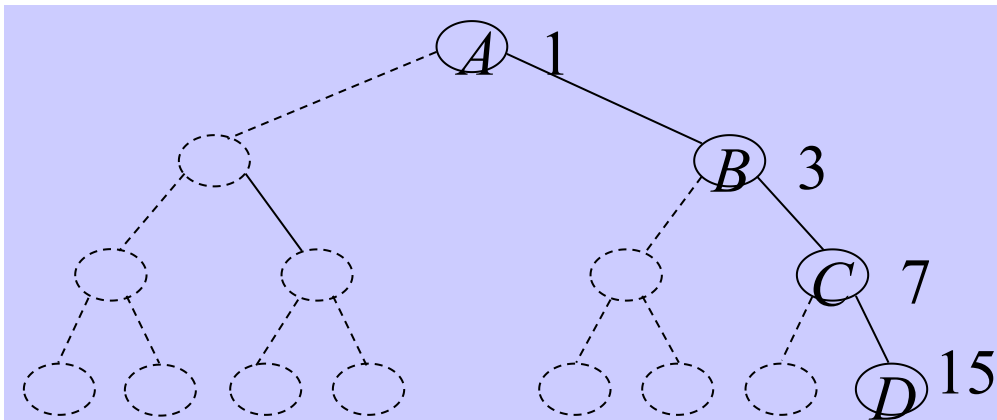
0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	0	D	E	0	0	0	F	0	0	G

三、二叉树的顺序存储

一棵二叉树改造后成完全二叉树形态，需增加很多空结点，造成存储空间的浪费。

一棵斜树的顺序存储会怎样呢？

深度为 k 的右斜树， k 个结点需分配 $2^k - 1$ 个存储单元。



二叉树的顺序存储结构

▶▶▶ 四、二叉树的链式存储

1. 二叉链表

基本思想： 令二叉树的每个结点对应一个链表结点，链表结点除了存放与二叉树结点有关的数据信息外，还要设置指示左右孩子的指针。

结点结构：

lchild	data	rchild
--------	------	--------

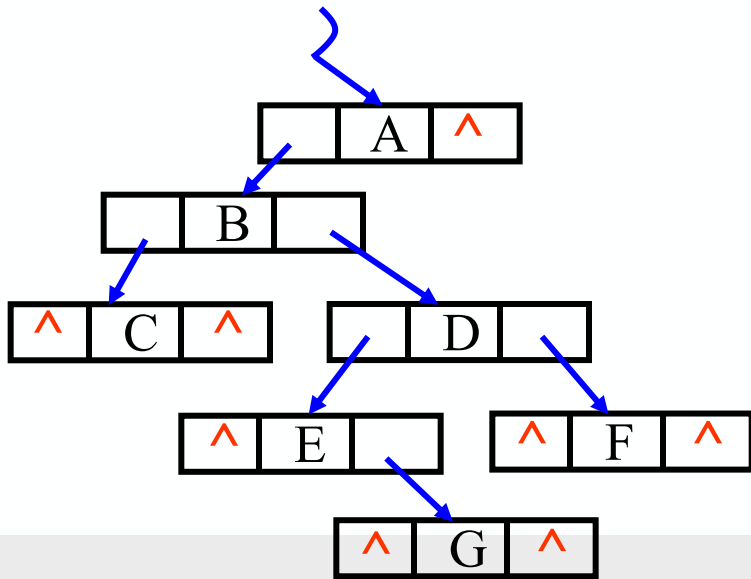
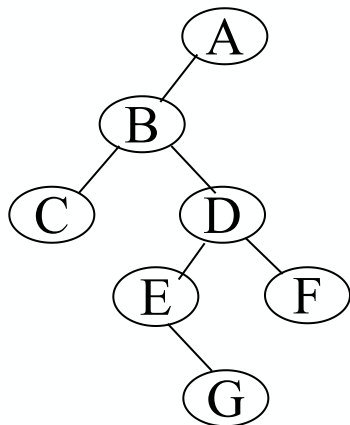
其中，**data**：数据域，存放该结点的数据信息；

lchild：左指针域，存放指向左孩子的指针；

rchild：右指针域，存放指向右孩子的指针。

四、二叉树的链式存储

1. 二叉链表



```
typedef struct BiNode{  
    TElemType  data;  
    struct BiNode  *lchild,*rchild; //左右孩子指针  
} BiNode,*BiTree;
```

▶▶▶ 四、二叉树的链式存储

2. 三叉链表

在二叉链表的基础上增加了一个指向双亲的指针域。

结点结构

lchild	data	parent	rchild
--------	------	--------	--------

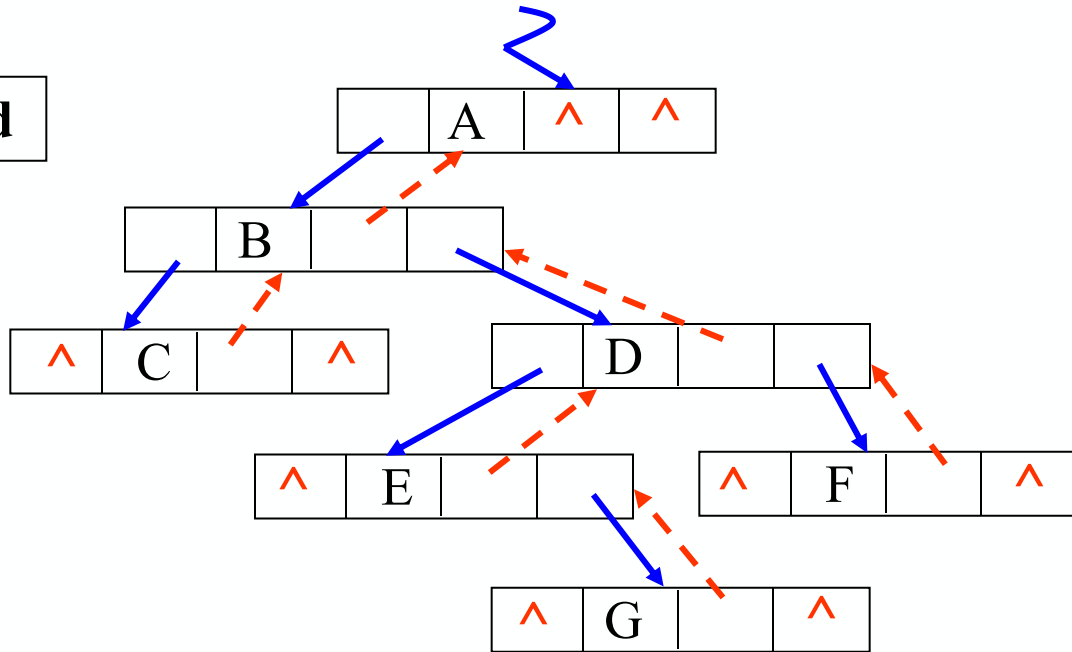
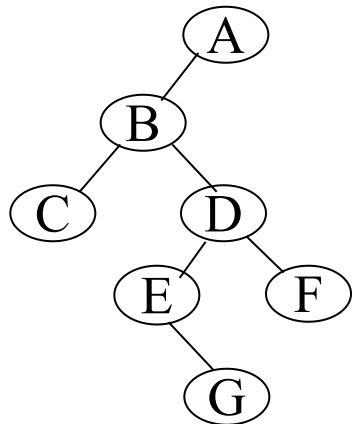
其中：**data**、**lchild**和**rchild**三个域的含义同二叉链表的结点结构；

parent域为指向该结点的双亲结点的指针。

四、二叉树的链式存储

2. 三叉链表

lchild	data	parent	rchild
--------	------	--------	--------



```
typedef struct TriTNode
```

```
{ TelemType data;
```

```
    struct TriTNode *lchild,*parent,*rchild;
```

```
}TriTNode,*TriTree;
```

1. 几种特殊形态的二叉树
2. 二叉树的性质
3. 二叉树常用的存储结构