## 第五章知识小结

**1、树和二叉树的定义**

结点：即树的数据元素

结点的度：结点挂接的子树数

结点的层次：从根到该结点的层数（根结点算第一层）

终端结点：即度为0的结点，即叶子

分支结点：即度不为0的结点（也称为内部结点）

树的度：所有结点度中的最大值

树的深度：指所有结点中最大的层数，也称树的高度

1. **二叉树的性质**

性质1: 在二叉树的第i层上至多有**2i-1**个结点

性质2: 深度为k的二叉树至多有2**k**-1个结点

性质3: 对于任何一棵二叉树，若度为2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）

性质4: 具有n个结点的完全二叉树的深度必为06bef199a654356c90d3401396534b6

性质5: 对完全二叉树，若从上至下、从左至右编号，则编号为i 的结点，其左孩子编号必为2i，其右孩子编号必为2i＋1；其双亲的编号必为dd6b532b8e8b88af025b1bd0802df71。

**3、特殊形态的二叉树**

满二叉树：一棵深度为k 且有2**k** -1个结点的二叉树。（特点：每层都“充满”了结点）

完全二叉树：深度为k 的，有n（n< 2k -1 ）个结点的二叉树，当且仅当其每一个结点都与深度为k 的满二叉树中编号从1至n的结点一一对应。

只有最后一层叶子不满，且最后一层叶子全部集中在左边

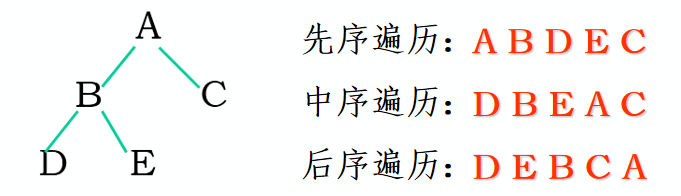
1. **二叉树的顺序存储**

存储方法：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。

优点：结点间关系蕴含在其存储位置中

缺点：浪费空间，仅适于存满二叉树和完全二叉树

1. **遍历二叉树**

****D=根节点 L=左孩子 R=右孩子

先序遍历：DLR，即先根再左再右

中序遍历：LDR，即先左再根再右

后序遍历：LRD，即先左再右再根

结论：

1.若二叉树中各结点的值均不相同，则：

(1)由二叉树的前序序列和中序序列，或由其后序序列和中序序列均能唯一地确定一棵二叉树，

(2)但由前序序列和后序序列却不一定能唯一地确定一棵二叉树。

2.时间效率:O(n) //每个结点只访问一次

3.空间效率:O(n) //栈占用的最大辅助空间

**5、线索二叉树**

在n个结点的二叉链表中，有n+1个空指针域

1）若结点有左子树，则lchild指向其左孩子；

否则， lchild指向其直接前驱（即线索）；

2）若结点有右子树，则rchild指向其右孩子；

否则， rchild指向其直接后继（即线索）。

为了避免混淆，增加两个标志域

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| lchild | LTag | data | RTag | rchild |

若LTag=0, lchild指向左孩子；若 LTag=1, lchild指向其前驱。

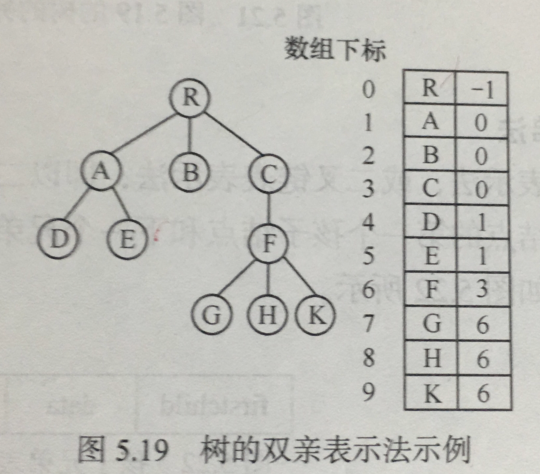
若RTag=0, rchild指向右孩子；若 RTag=1, rchild指向其后继。

**6、树的存储结构**

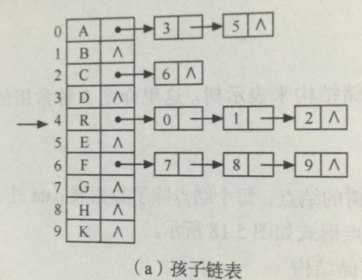
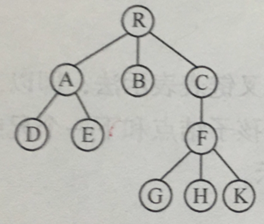
1.双亲表示法

用一组连续的存储单元存储树的结点。

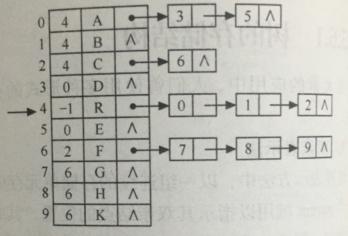
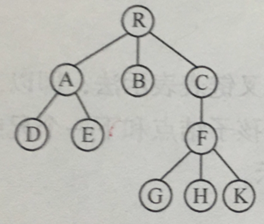
每个结点包含一个数据域和一个指向双亲结点位置的指针域。



2.孩子表示法

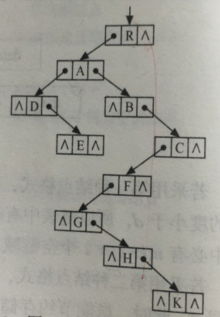
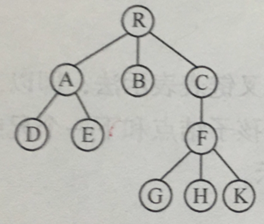


3.孩子双亲表示法



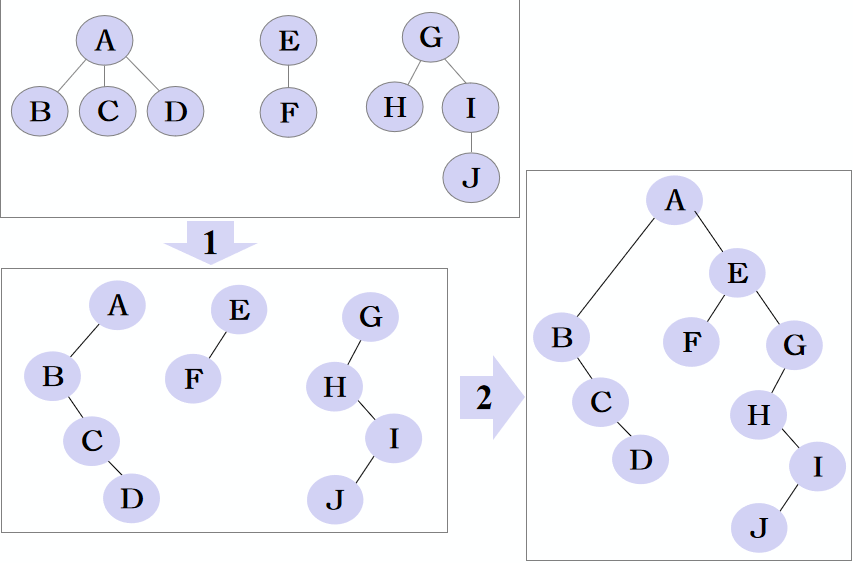
4.孩子兄弟表示法

|  |  |  |
| --- | --- | --- |
| firstchild | Data | nextsibling |



**树转换成二叉树规则：每个结点的左指针指向其第一个孩子结点，右指针指向它在树中的相邻兄弟结点，即左孩子右兄弟。**

**此处举个栗子**



**7、哈夫曼树及其应用**

路径：由一结点到另一结点间的分支所构成。

路径长度：路径上的分支数目，**a→e的路径长度＝2**

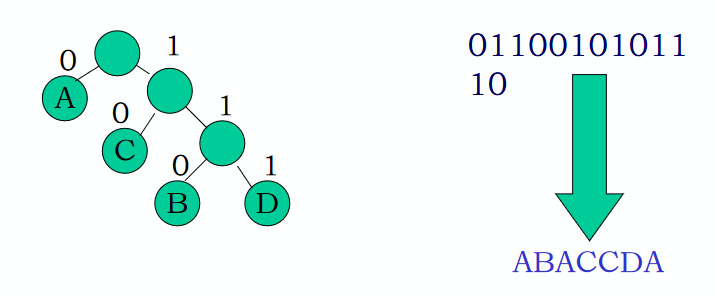
带权路径长度：结点到根的路径长度与结点上权的乘积。

树的带权路径长度：树中所有叶子结点的带权路径长度之和。

哈夫曼树：带权路径长度最小的树

**哈夫曼编码的译码过程：**

分解接收字符串：遇“0”向左，遇“1”向右；一旦到达叶子结点，则译出一个字符，反复由根出发，直到译码完成。



特点：每一码都不是另一码的前缀，绝不会错译! 称为前缀码

**哈夫曼编码的总结：**

1.哈夫曼编码是不等长编码。

2.哈夫曼编码是前缀编码，即任一字符的编码都不是另一字符编码的前缀。

3.哈夫曼编码树中没有度为1的结点。若叶子结点的个数为n，则哈夫曼编码树的结点总数为 2n-1。

4.发送过程：根据由哈夫曼树得到的编码表送出字符数据。

5.接收过程：按左0、右1的规定，从根结点走到一个叶结点，完成一个字符的译码。反复此过程，直到接收数据结束。