**树**

1. **树**

**定义：**

树是一个或多个节点的有限集合，且其中：

1. 存在一个称为根的特定节点；
2. 其余的节点被分成n0个互不相交的集合,…,,其中的每个集合都是一棵树，,…,称为根节点的子树。

**术语：**

节点：一个节点代表信息项和指向其他节点的分支。

度： 一个节点的度是指该节点的子树个数；

树的度是树中所有节点的度的最大值。

叶子(叶节点)：度为0的节点。

父/子节点：具有子树的节点称为这些子树根节点的父节点，而这些子树

的根节点称为该节点的子节点。

层：规定根节点为第一层。其他所有节点的层都是其父节点的层号加1.

树的高度(或深度)：树中所有节点的最大层号。

**树的存储表示：**



图 1 树

**列表存储表示：**在这种表示方法中，任何一棵子树都是一个列表。如图1中的树，按列表形式表示为：(A(B(E(K,L),F),C(G),D(H(M),I,J))。

如果以链表形式表示，则根据子树的不同，1个节点需要有可变数量的链接域。

**左儿子-右兄弟存储表示：**

图1所示的树用左儿子-右兄弟存储表示为：



图 2 树的左儿子-右兄弟存储表示

由于每个节点都最多只会有一个最左儿子和一个最近的右兄弟，所以此表示方法是用两个链接域的节点结构来表示树。即：”数据+左儿子+右兄弟”的结构。

**2叉树存储表示：**



图 3 树的二叉树存储表示

1. **二叉树**

**定义：**

二叉树是有限多个节点的集合，这个集合或者是空寂，或者由一个根节点和两棵互不相交的、分别称为左子树和右子树的二叉树组成。

二叉树和树的区别：一棵树至少要含有一个节点，而二叉树可以为空；

其次，树的子树不区分顺序，而二叉树要区分左子树和右子树。

左倾斜树： 树中的每一个节点都是该结点父亲的左儿子；

右倾斜树： 树中的每一个节点都是该结点父亲的右儿子；

满二叉树： 深度为k的满二叉树是深度为k且具有个结点的二叉树；

对满二叉树进行顺序编号，即从第一层的根节点开始编号，然后对第二层的所有结点进行编号，以此类推。同一层结点按照从左到右的顺序编号。

完全二叉树：当且仅当二叉树的结点与深度为k的满二叉树结点编号相对应时，该树是完全的。

**性质：**

1. 在二叉树中，第i层节点数最多为；
2. 在深度为k的二叉树中，节点总数最多为；
3. 对于任何非空的二叉树T，如果叶结点的个数为，而度为2的结点树为，则；

证明： 设为二叉树中度为1的结点数，n是结点总数，则易知：

对二叉树中的分支数进行计数，可以看到，除根节点外，其余结点都有一个进入分支，设B为分支总数，则n = B + 1。又因为所有分支都是从度为1和度为2的结点发出的，从而B = 。整理3个等式即得：；

**二叉树的存储表示：**

1. **数组存储表示：**

可以使用一维数组来存储二叉树的结点(不使用数组的第0个位置)。

对于一棵n个结点的完全二叉树，采用顺序存储表示，那么，对于任意一个下表为i()的结点，有：

若i，则其父结点parent(i)的编号为；若i=1，则i是根节点；

若2i，则其左儿子left\_child(i)的编号为2i；否则，i没有左儿子；

若2i+1，则其右儿子right\_child(i)的编号为2i+1;否则，i没有右儿子。