**1：树的定义**

**由一个或者多个（n>=0）结点组成的有限集合，把它叫做“树”是因为它看起来像一棵倒挂的树，**

**也就是说它是根朝上，而叶朝下的。它具有以下的特点：**

**每个节点有零个或多个子节点；没有父节点的节点称为根节点；**

**每一个非根节点有且只有一个父节点；除了根节点外，每个子节点可以分为多个不相交的子树**

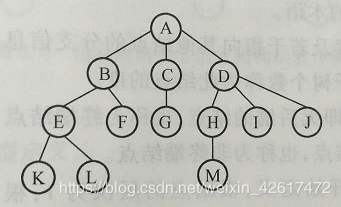
**①其中必有一个称为根的特定结点，它没有直接前驱，但有零个或多个直接后继。**

**②其余n-1个结点可以划分成m(m>=0)个互不相交的有限集T1,T2,……,Tm,其中Ti又是一棵树，称为根的子树。**

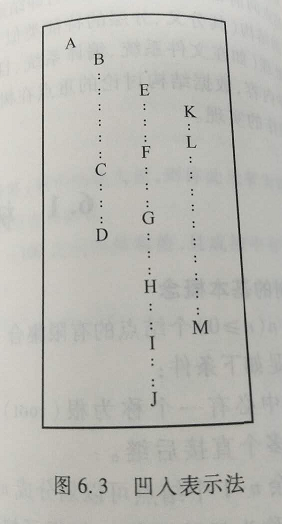
**每棵子树的根结点有且仅有一个直接前驱，但有零个或多个直接后继。**

**树的图解表示法：**

**1>树形表示法：例如家族关系的表示，A有3个孩子B、C、D，B有两个孩子E、F，D有3个孩子H、I、J等。**

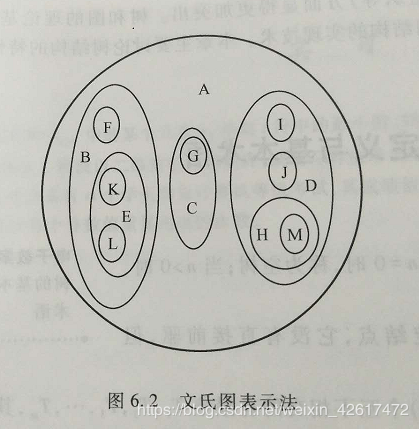


**2>广义表形式（嵌套括号表示法）：即(A(B(E(K,L),F),C(G),D(H(M),I,J)))。例如一本书A分为B、C、D三章，B章又分为E、F两节，E节又分为K、L两段，等等。**



**3>凹入表示法：用位置的缩进表示其层次，实际上程序的锯齿形结构就是这种结构。**

**3>嵌套表示法**



**树的相关术语：**

**根-----根结点（没有前驱）**

**有序树：在树T中，如果各子树Ti之间是有先后次序的，则称为有序树。**

**无序树：结点各子树可交换位置**

**孩子结点：一个结点的直接后继称为该结点的孩子结点。**

**双亲结点:一个结点的直接前驱称为该结点的双亲结点。**

**兄弟结点：同一双亲结点的孩子结点之间互称兄弟结点。**

**堂兄弟结点：父亲是兄弟关系或堂兄关系的结点称为堂兄弟结点。**

**祖先结点：一个结点的祖先结点是指从根节点到该结点的路径上的所有结点。**

**子孙结点：一个结点的直接后继和间接后继称为该结点的子孙结点。**

**结点：包括一个数据元素及若干指向其他结点的分支信息。**

**结点的度：一个结点的子树个数称为此结点的度。**

**结点的层次：从根结点开始定义，根结点的层次为1，根的直接后继的层次为2，以此类推。**

**叶结点：度为0的结点，即无后继的结点，也称为终端结点。**

**分支结点：度不为0的结点，也称为非终端结点。**

**树的度：树中所有结点的度的最大值。**

**树的高度（深度）：树中所有结点的层次的最大值。**

**森林：m(m>=0)棵互不相交的树的集合。将一棵非空树的根节点删去，树就变成一个森林；反之，给森林增加一个统一的根结点，森林就变成一棵树。**

**二叉树**

**二叉树的结构最简单**

**二叉树的定义：**

**基本特征：**

**每个结点最多有两个子树的树结构，**

**左子树”和“右子树”次序不能颠倒**

**二叉树性质**

**(1) 在非空二叉树中，第i层的结点总数不超过个结点（i>0)**

**(2) 深度为h的二叉树最多有个**IMG_256**结点(h>=1)，最少有h个结点；**

**(3) 对于任意一棵二叉树，如果其叶结点数为N0，而度数为2的结点总数为N2，则N0=N2+1**

**(4) 具有n个结点的[完全二叉树](https://baike.baidu.com/item/%E5%AE%8C%E5%85%A8%E4%BA%8C%E5%8F%89%E6%A0%91" \t "https://baike.baidu.com/item/%E4%BA%8C%E5%8F%89%E6%A0%91/_blank)的深度为IMG_256（注：[ ]表示向下取整）**

**[满二叉树](https://baike.baidu.com/item/%E6%BB%A1%E4%BA%8C%E5%8F%89%E6%A0%91" \t "https://baike.baidu.com/item/%E4%BA%8C%E5%8F%89%E6%A0%91/_blank)——除了叶结点外每一个结点都有左右子叶且叶子结点都处在最底层的二叉树。**

**[完全二叉树](https://baike.baidu.com/item/%E5%AE%8C%E5%85%A8%E4%BA%8C%E5%8F%89%E6%A0%91" \t "https://baike.baidu.com/item/%E4%BA%8C%E5%8F%89%E6%A0%91/_blank)——若设二叉树的高度为h，除第 h 层外，其它各层 (1～h-1) 的结点数都达到最大个数，第h层有[叶子结点](https://baike.baidu.com/item/%E5%8F%B6%E5%AD%90%E7%BB%93%E7%82%B9" \t "https://baike.baidu.com/item/%E4%BA%8C%E5%8F%89%E6%A0%91/_blank)，并且叶子结点都是从左到右依次排布，这就是[完全二叉树](https://baike.baidu.com/item/%E5%AE%8C%E5%85%A8%E4%BA%8C%E5%8F%89%E6%A0%91" \t "https://baike.baidu.com/item/%E4%BA%8C%E5%8F%89%E6%A0%91/_blank)**。

**特点：（1）叶子结点只可能在层次最大的两层上出现**

**(2) 具有n个结点的[完全二叉树](https://baike.baidu.com/item/%E5%AE%8C%E5%85%A8%E4%BA%8C%E5%8F%89%E6%A0%91" \t "https://baike.baidu.com/item/%E4%BA%8C%E5%8F%89%E6%A0%91/_blank)的深度为IMG_256**

**（注：[ ]表示向下取整）**

## 二叉树遍历

**遍历是对树的一种最基本的运算，所谓遍历二叉树，就是按一定的规则和顺序走遍二叉树的所有结点，使每一个结点都被访问一次，而且只被访问一次**。

### **1>先序遍历**

**首先访问根，再先序遍历左（右）子树，最后先序遍历右（左）子树**

### **2>中序遍历**

**首先中序遍历左（右）子树，再访问根，最后中序遍历右（左）子树**

### **3>后序遍历**

**首先后序遍历左（右）子树，再后序遍历右（左）子树，最后访问根**

### **4>层次遍历**

**即按照层次访问，通常用[队列](https://baike.baidu.com/item/%E9%98%9F%E5%88%97" \t "https://baike.baidu.com/item/%E4%BA%8C%E5%8F%89%E6%A0%91/_blank)来做。访问根，访问子女，再访问子女的子女（越往后的层次越低）（两个子女的级别相同）**