 1．6树与二叉树一、树的基本概念在树结构中，每一个结点只有一个前件，称为父结点，没有前件的结点只有一个，称为树的根结点，简称为树的根。在树结构中，每一个结点可以有多个后件，它们都称为该结点的子结点。没有后件的结点称为叶子结点。在树结构中，一个结点所拥有的后件个数称为该结点的度。叶子结点的度为0。树的最大层次称为树的深度。在一个算术表达式中，有运算符和运算对象。一个运算符可以有若干个运算对象。例职，取正（）等只有一个运算对象，称为单目运算符；二个运算对象称为双目运算符，三目运算符。用树来表示算术表达式的原则如下：表达式中的每一个运算符在树中对应一个结点，称为运算符结点。运算符的每一个运算对象在树中为该运算符结点的子树（在树中的顺序为从左到右）。运算对象中的单变量均为叶子结点。二、二叉树及其基本性质1、什么是二叉树二叉树是一种很有用的非线性结构。二就树具有以下两个特点：非空二叉树只有一个根结点；每一个结点最多有两棵子树，且分别称为该结点的左子树与右子树。由以上特点可以看出，在二叉树中，每一个结点的度最大为2，即所有子树（左子树或右子树）也均为二叉树，而树结构中的每一个结点的度可以是任意的。另外，二叉树中的每一个结点的子树被明显地分为左子树与右子树。可以没有其中的一个，也可以全没有。二叉树的基本性质性质1：在二叉树的第K层上，最多有（K≥1）个结点。性质2：浓度

为M的二叉树最多有2m-1 个结点。深度为m 的二叉树是指二叉树共有m层。性质3：在任意一棵二叉树中度为0的结点（即叶子结点）总是比度为2的结点多一个。性质4：具有n个结点的二叉树，其深度至少为[log2n] 1,其中[log2n]表示取的整数部分。满二叉树与完全二叉树满二叉树与完全二叉树是两种特殊形态的二叉树。满二叉树所谓满二叉树是指这样的一种二叉树；除最后一层外，每一层上的所有结点都有两个子结点。这就是说，在满二叉树中，每一层上的结点数都达到最大值，即在满二叉树的第K层上有2K-1个结点，且深度为m的满二叉树有2m-1个结点。完全二叉树所谓完全二叉树是指这样的二叉树，除最后一层外，每一层上的结点数均达的最大值；在最后一层上只缺少右边的若干结点。列确切地说，如果从根结点起，对二叉树的结点自上而下、自左至右用自然数进行边疆编号，则深度为m、且有n 个结点的二叉树，当且仅当其每一个结点都与深度为m的满二叉树中编号从1到n的结点一一对应时，称之为完全二叉树。对于完全二叉树来说，叶子结点只可能在层次最大的两层上出现；对于任何一个结点，若其右分支下的子孙结点的最大层次为p，则其左分支下的子孙结点的最大层次或为p，或为p 1。由满二叉树与完全二叉树的特点可以看出，满二叉树也是完全二叉树，而完全二叉树一般不是满二叉树。完全二叉树还具有以下两个性质：性质5：具有n个结点的完全二叉树的深度为[log2n] 1。性质6：设完全二叉树共有n个结点。如果从根结点开始，按层序（每一层从左到右）用自然数1，2，…，n给结点进行编号，则对于编号为k （k=1,2,…n）的结点有以下结论：若k=1，则该结点为根结点，它没有父结点；若k>1，则该结

点的父结点编号为INT(k/2)。若2k≤n，则编号为k 的结点的左子结点编号为2k ；否则该结点无左子结点（显然也没有右子结点）。若2k 1≤n，则编号为k 的结点的右子结点编号为2k 1；否则该结点无右子结点。三、二叉树的存储结构二叉树的遍历二叉树的遍历是指不重复地访问二叉树的所有结点。在遍历二叉树的过程中，一般先遍历左子树，然后再遍历右子树。1、前序遍历（DLR）所谓前序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中，首先访问根结点，然后遍历左子树，最后遍历右子树；并且，在遍历左、右子树时，仍然先访问根结点，然后遍历左子树，最后遍历右子树。F，C，A，D，B，E，G，H，P2、中序遍历（LDR）所谓中序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中，首先遍历左子树，然后访问根结点，最后遍历右子树；并且，在遍历左、右子树时，仍然先遍历左子树，然后访问根结点，最后遍历右子树。A，C，B，D，F，E，H，G，P3、后序遍历（LRD）所谓中序遍历是指在访问根结点、遍历左子树与遍历右子树这三者中，首先遍历左子树，然后遍历右子树，最后访问根结点；并且，在遍历左、右子树时，仍然先遍历左子树，然后遍历右子树，最后访问根结点。A，B，D，C，H，P，G，E，F1.7查找技术一、顺序查找顺序查找又称顺序搜索。顺序查找一般是指在线性表中查找指定的元素，其基本方法如下：从线性表的第一个元素开始，依次将线性表中的元素与被查元素进行比较，若相等则表示找到（即查找成功）；若线性表中所有的元素都与被查元素进行了比较但都不相等，则表示线性表中没有要找的元素（即查找失败）。顺序查找的效率是很低的。以下两种情况

只能采用顺序查找：如果线性表无序表（即表中元素的排列是无序的），则不管是顺序存储结构还是链式存储结构，都只能用顺序查找。即使是有序线性表，如果采用链式存储结构，也只能用顺序查找。二、二分法查找二分法查找只适用于存储的有序表。在此所说的有序表是指线性表的中元素按值非递减排列（即从小到大，但允许相邻元素值相等）。设有序线性表的长度为n，被查元素为x，则对分查找的方法如下：将x与线性表的中间项进行比较：若中间项的值等于x，则说明查到，查找结束；若x小于中间项的值，则在线性表的前半部分（即中间项以前的部分）以相同的方法进行查找；若x大于中间项的值，则在线性表的后半部分（即中间项以后的部分）以相同的方法进行查找。这个过程一直进行到查找成功或子表长度为0（说明线性表中没有这个元素）为止。显然，当有序线性表为顺序存储时才能采用二分查找，并且，二分查找的效率要比顺序查找高得多。可以证明，对于长度为n的有序线性表，在最坏情况下，二分查找只需要比较log2n次，而顺序查找需要比较n次。