

5.2 二叉树的概念

二叉树的定义：二叉树是另一种树形结构，其特点是每个结点至多只有两棵子树，并且二叉树的子树有左右之分，其次序不能任意颠倒

二叉树的定义及其主要特性

二叉树与度为2的有序树的区别

度为2的树至少有3个结点，而二叉树可以为空

孩子结点

度为2的有序树的孩子的左右次序是相对于另一孩子而言的，若某个结点只有一个孩子，则这个孩子就无须区分其左右次序

二叉树无论其孩子数是否为2,均需确定其左右次序

满二叉树

一棵高度为h,且含有 $2^h - 1$ 个结点的二叉树称为满二叉树

双亲为 $\lfloor i/2 \rfloor$

对于编号为i的结点

若有左孩子,则左孩子为2i

若有右孩子,则右孩子为2i + 1

完全二叉树

高度为h、有n个结点的二叉树,当且仅当其每个结点都与高度为h的满二叉树中编号为1 ~ n的结点一一对应时,称为完全二叉树

$i \leq \lfloor n/2 \rfloor$ 则结点为分支结点，否则为叶子结点

叶子结点只可能在层次最大的两层上出现。对于最大层次中的叶子结点，都依次排列 在该层最左边的位置上

若有度为1的结点，则只可能有一个，且该结点只有左孩子而无右孩子（重要特征）

按层序编号后,一旦出现某结点（编号为i）为叶子结点或只有左孩子,则编号大于i 的结点均为叶子结点

若n为奇数,则每个分支结点都有左孩子和右孩子

若n为偶数,则编号最大的分支结点（编号为n/2）只有左孩子,没有右孩子,其余分支结点左、右孩子都有

特点

二叉排序树

左子树上所有结点的关键字均小于根结点的关键字;右子树上的所有结点的关键字均大于根结点的关键字;左子树和右子树又各是一棵二叉排序树

平衡二叉树

树上任一结点的左子树和右子树的深度之差不超过1

非空二叉树上的叶子结点数等于度为2的结点数加1

非空二叉树上第k层上至多 2^{k-1} 个结点

高度为h的二叉树至多有 $2^h - 1$ 个结点

对完全二叉树按从上到下、从左到右的顺序依次编号1,2,...,n,则有以下关系

当i>1时,结点i的双亲的编号为 $\lfloor i/2 \rfloor$ 即当i为偶数时,其双亲的编号为i/2它是双亲的左孩子;当i为奇数时,其双亲的编号为(i-1)/2,它是双亲的右孩子

$2i \leq n$ 结点i的左孩子编号为2i,否则无左孩子

$2i + 1 \leq n$ 结点i的右孩子编号为2i + 1,否则无右孩子

结点i所在层次（深度）为 $\lfloor \log_2 i \rfloor + 1$

具有n个（n>0）结点的完全二叉树的高度为 $\lceil \log_2(n + 1) \rceil$ 或 $\lfloor \log_2 n \rfloor + 1$

二叉树的性质

顺序存储结构

用一组地址连续的存储单元依次自上而下、自左至右存储完全二叉树 上的结点元素，即将完全二叉树上编号为i的结点元素存储在一维数组下标为i - 1的分量中

完全二叉树和满二叉树采用顺序存储比较合适

一般的二叉树，为了能反映二叉树中结点之间的逻辑关系，只能添加并不存在的空结点，让其每个结点与完全二叉树上的结点相对照，再存储到一维数组的相应分量 中

由于顺序存储的空间利用率较低，因此二叉树一般都采用链式存储结构，用链表结点来存储二叉树中的每个结点

链式存储结构

结构描述

```
typedef struct BiTNode{
    ElemType data;           //数据域
    struct BiTNode *lchild,*rchild; //左、右孩子指针
}BiTNode,*BiTree;
```

在含有n个结点的二叉链表中,含有n + 1个空链域