二叉树的定义及其主要特性

二叉树的定义

二叉树是一种特殊的树形结构，其特点是每个结点至多只有两棵子树（即二叉树中不存在度大于2的结点），并且二叉树的子树有左右之分，其次序不能任意颠倒。

与树相似，二叉树也以递归的形式定义。二叉树是n(n>=0)个结点的有限集合：

1. 或者为空二叉树，即n=0。
2. 或者由一个根结点和两个互不相交的被称为根的左子树和右子树组成。左子树和右子树又分别是一棵二叉树。

二叉树是有序树，若将其左、右子树颠倒，则称为另一棵不同的二叉树。即使树中结点只有一棵子树，也要区分它是左子树还是右子树。二叉树的5种基本形态如图所示。

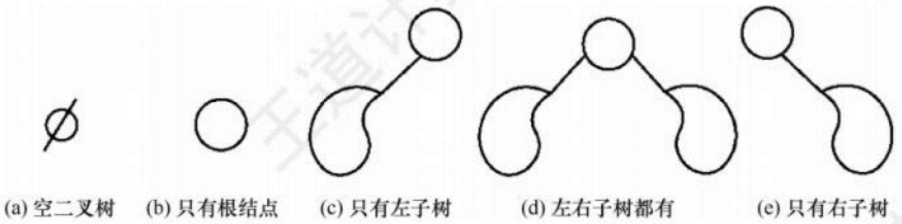


图 5.2　二叉树的 5 种基本形态

二叉树与度为2的有序树的区别：

1. 度为2的树至少有3个结点，而二叉树可以为空。
2. 度为2的有序树的孩子左右次序是相对于另一个孩子而言的，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，而二叉树无论其孩子数是否为2，均需确定其左右次序，即二叉树的结点次序不是相对于另一结点而言的，而是确定的。

几种特殊的二叉树

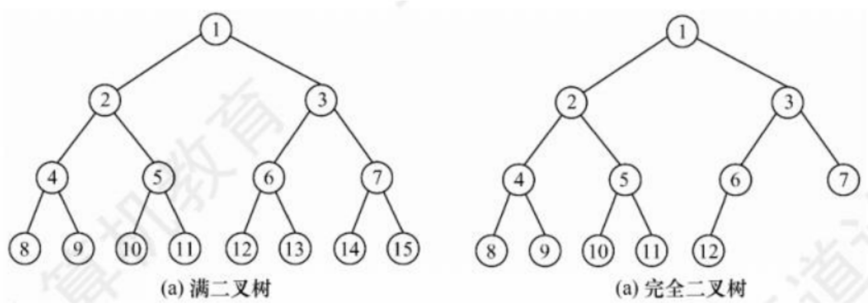


图 5.3　两种特殊形态的二叉树[①]

满二叉树

一棵高度为h,且有2^h-1个结点的二叉树称为满二叉树,即二叉树中的每层都含有最多的结点,如上图a所示.满二叉树的叶结点都集中在二叉树的最下一层,并且除叶结点之外的每个结点度数均为2.
可以对满二叉树按层序编号：约定编号从根结点(根结点编号为1)起,自上而下,自左向右.这样,每个结点对应一个编号,对于编号为i的结点,若有双亲,则其双亲为$\lfloor i/2 \rfloor$,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,如图b所示.其特点如下：

①若$i \le \lfloor n/2 \rfloor$,则结点$i$为分支结点,否则为叶结点.

②叶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结点,都依次排列在该层最左边的位置上.

③若有度为1的结点,则最多只可能有一个,且该结点只有左孩子而无右孩子.

④按层序编号后,一旦出现某结点(编号为i)为叶结点或只有左孩子,则编号大于i的结点均为叶结点.

⑤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.

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

树种任意一个结点的左子树和右子树的高度之差的绝对值不超过1.
关于二叉排序树和平衡二叉树的详细介绍, 见本书中7.3节.

正则k叉树树高和结点数的关系的应用

正则二叉树

树中每个分支结点都有2个孩子, 即树中只有度为0或2的结点.

二叉树的性质

1.非空二叉树上的叶结点数等于度为2的结点数加1, 即$n_0 = n_2 + 1$.
证明：设度为0, 1和2的结点个数分别为n_0, n_1和n_2, 结点总数$n = n_0 + n_1 + n_2$.
再看二叉树中的分支数, 除根结点外, 其余结点都有一个分支进入, 设B为分支总数,
则$n = B + 1$.由于这些分支是由度为1或2的结点射出的, 因此又有$B = n_1 + 2n_2$.
于是得$n_0 + n_1 + n_2 = n_1 + 2n_2 + 1$, 则$n_0 = n_2 + 1$.
该性质经常在选择题中涉及, 希望读者牢记并灵活应用

2.非空二叉树的第k层最多有2^{k-1}个结点$(k \geq 1)$.
第一层最多有$2^{1-1} = 1$个结点(根), 第2层最多有$2^{2-1} = 2$个结点, 以此类推, 可以证明
其为一个公比为2的等比数列2^{k-1}.

3.高度为h的二叉树至多有$2^h - 1$个结点$(h \geq 1)$.
该性质利用性质2求前h项的和, 即等比数列求和的结果.
性质2和性质3还可以拓展到m叉树的情况, 即m叉树的第k层最多有m^{k-1}个结点,
高度为h的m叉树至多有$(2^h - 1)/(m - 1)$个结点.

4.对完全二叉树按从上到下、从左到右的顺序依次编号$1, 2, \ldots, n$, 则有以下关系：
①若$i \leq \lfloor n/2 \rfloor$, 则结点$i$为分支结点, 否则为叶结点, 即最后一个分支结点的编号为$\lfloor n/2 \rfloor$.
②叶结点只可能在层次最大的两层上出现(若删除满二叉树中最底层, 最右边的连续2个或以上的叶结点, 则倒数第二层将会出现叶结点).
③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(度为1的分支结点只可能是最后一个分支结点, 其结点编号为$\lfloor n/2 \rfloor$).
④按层序编号后, 一旦出现某结点(如结点i)为叶结点或只有左孩子的情况, 则编号大于i的结点均为叶结点(与结论①和结论③是相通的).
⑤若n为奇数, 则每个分支结点都有左、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, 其余分支结点都有左、右孩子
⑥当$i > 1$时, 结点i的双亲结点的编号为$\lfloor i/2 \rfloor$.
⑦若结点i有左、右孩子, 则左孩子编号为$2i$, 右孩子编号为$2i + 1$.
⑧结点i所在层次(深度)为$\lfloor \log_2 i \rfloor + 1$.

5.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$\lfloor \log_2 n \rfloor + 1$.
设高度为h, 根据性质3和完全二叉树的定义有
$2^{h-1} - 1 < n \leq 2^h - 1$或者$2^{h-1} \leq n < 2^h$
得$2^{h-1} < n + 1 \leq 2^h$, 即$h - 1 < \log_2 (n + 1) \leq h$, 因为$h$为正整数, 所以$h = \lceil \log_2 (n + 1) \rceil$,
或者得$h - 1 \leq \log_2 n < h$, 所以$h = \lfloor \log_2 n \rfloor + 1$.

二叉树的存储结构

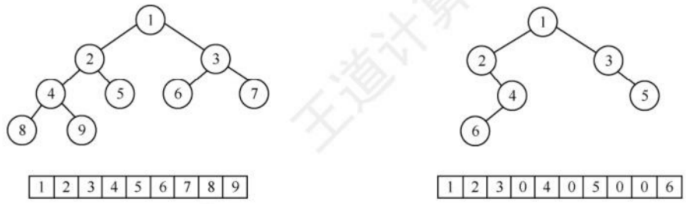
顺序存储结构

二叉树的顺序存储是指用一组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。

依据二叉树的性质，完全二叉树和满二叉树采用顺序存储比较合适，树中结点的序号可以唯一地反映结点之间的逻辑关系，这样既能最大可能地节省存储空间，又能利用数组元素的下标值确定结点在二叉树中的位置，以及结点之间的关系。

特定条件下二叉树树形及占用存储空间的分析

但对于一般的二叉树，为了让数组下标能反映二叉树中结点之间的逻辑关系，只能添加一些并不存在的空结点，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。然而，在最坏情况下，一个高度为h且只有h个结点的单支树却需要占据2^h-1个存储单元。二叉树的顺序存储结构如下图所示，其中0表示并不存在的空结点 。



(a) 完全二叉树的顺序存储结构　　(b) 一般二叉树的顺序存储结构

图 5.4　二叉树的顺序存储结构

建议从数组下标1开始存储树中的结点，保证数组下标和结点编号一致。

链式存储结构

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。在二叉树中，结点结构通常包括若干数据域和若干指针域，二叉链表至少包含3个域：数据域data、左指针域lchild和右指针域rchild，如下图所示。

lchild	data	rchild

图 5.5　二叉树链式存储的结点结构

下图所示为一棵二叉树及其对应的二叉链表。而实际上在不同的应用中，还可以增加某些指针域，如增加指向父结点的指针后，变为三叉链表的存储结构。

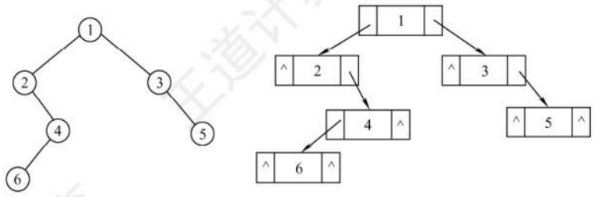


图 5.6　二叉链表的存储结构

二叉树的链式存储结构描述如下：

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

使用不同的存储结构时，实现二叉树操作的算法也会不同，因此要根据实际应用场合（二叉树的形态和需要进行的运算）来选择合适的存储结构。

容易验证，在含有n个结点的二叉链表中，含有n+1个空链域（重要结论，经常出现在选择题中）。在下一节中，我们将利用这些空链域来组成另一种链表结构——线索链表。