

第 5 章

树和二叉树

读史可以明鉴,知古可以鉴今。中国古代大量鸿篇巨制为中华文明和人类文明做出了重大贡献。坚定文化自信首先要对民族文化有更准确的理解。以二十四史为代表的纪传体正史与中华民族数千年文明相依相伴,是世界上唯一载录绵延数千年的信史,成为一代又一代中国人可以源源不断汲取的智慧源泉。

作为二十四史之首的《史记》,是司马迁以“究天人之际,通古今之变,成一家之言”的史识,专注十四年创作的中国第一部纪传体通史,被鲁迅誉为“史家之绝唱,无韵之离骚”,对后世具有非常大的影响。《史记》记载了上至皇帝时代,下至武帝太初四年间共 3000 年的历史。我们可以通过图 5-1 来了解这一部亘古通今的千年巨著。

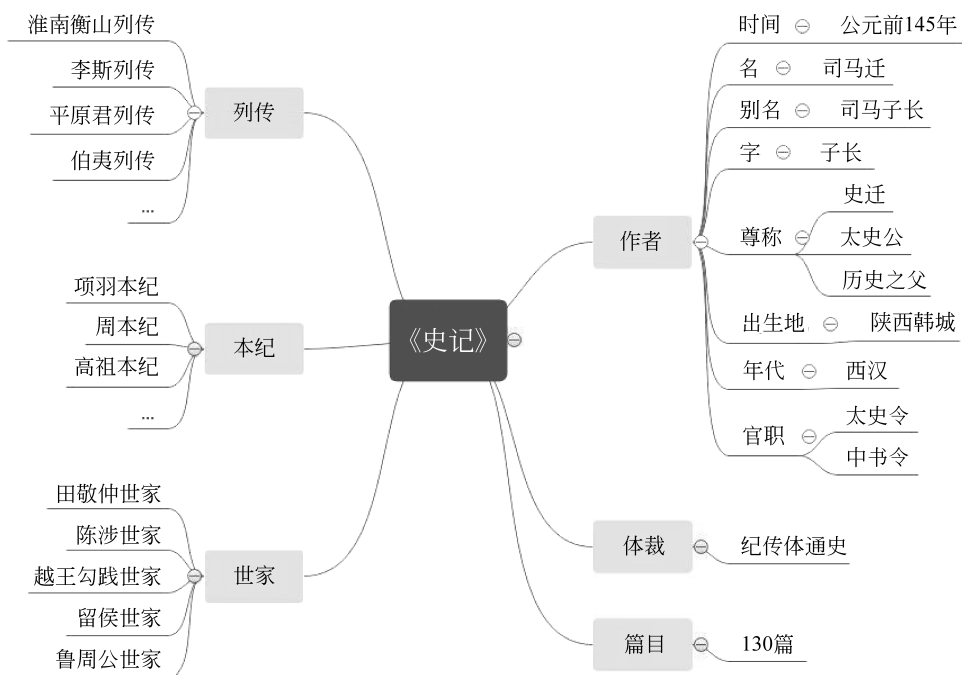


图 5-1 《史记》结构图

这种最简单的思维导图把所有的信息都组织在一个树状的结构图上,可以加强对信息的记忆与理解。树结构不仅可以被用于构建思维导图,还可以用于组织机构管理。

本章将介绍这种树状结构,包括其逻辑结构、存储结构以及基本操作,并将给出实际

应用问题的案例实现。

5.1 问题的提出

问题 1: 某大学设有若干个职能处室和院系,每个职能处室又根据业务需要设置若干个科室和研究所,而每个院系按专业和学科方向设置不同的教研室、研究所和实验室。要求对机构的设置进行动态管理,并提供创建、删除、增加、查询和显示等功能。

这些部门之间的关系如图 5-2 所示。

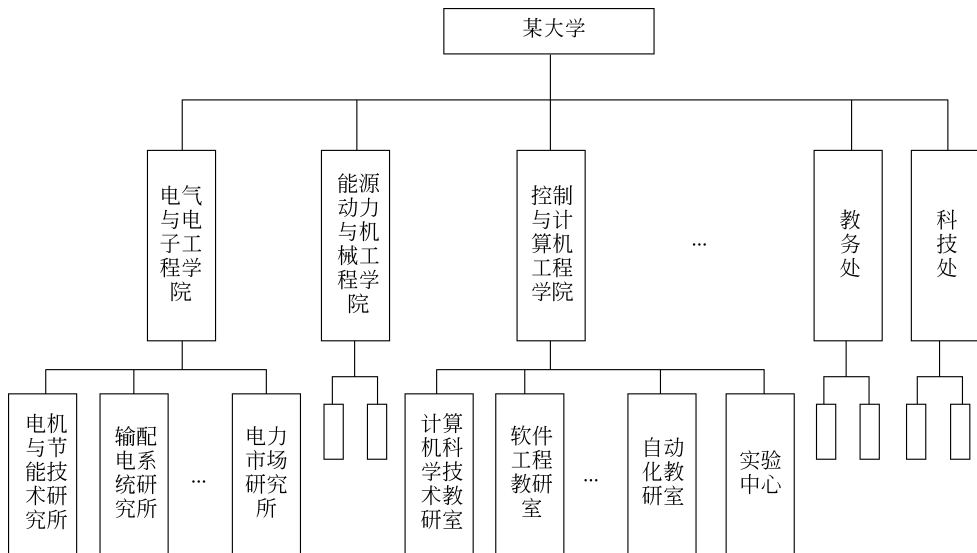


图 5-2 某大学机构设置示意图

问题 2: 在表达式求值中,如何存储表达式才能更便于计算表达式的值?

在第 3 章介绍了两种表达式求值的方法。无论是原表达式还是后缀表达式都是以字符串形式存储,这需要事先确定存储空间的大小。如果可以根据算术表达式动态分配存储空间,并且可方便地进行表达式求值,则可以有效解决一些应用系统中的表达式计算问题。

图 5-3 是表达式 $a+(b * c-d/e)$ 对应的二叉树存储结构,其后序遍历序列“ $abc * de/-+$ ”是原表达式的后缀表达式。

问题 1 中各个部门的名称是数据元素,问题 2 中的操作对象和运算符是数据元素,数据元素之间的关系不再是线性关系,而是层次关系。相邻两层中,上一层的一个数据元素可以与下一层的多个数据元素有关系,下一层的一个数据元素只能与上一层的一个数据元素有关系。这种关系称为一对多的层次关系,即

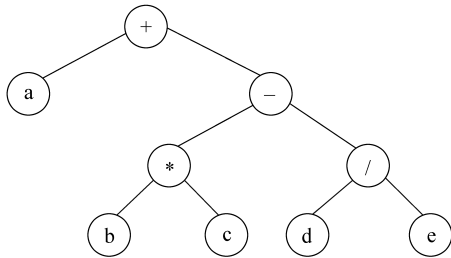


图 5-3 表达式的二叉树存储示意图

1:n。我们把具有这种层次关系的结构称为树状结构。

5.2 树的定义和基本术语

5.2.1 树的递归定义

树是由根结点和子树组成,具有以下特征。

(1) 有一个特定的称为该树之根的结点,即根结点。

(2) 结点数 $n=0$ 时,表示树是空树。

(3) 结点数 $n>0$ 时,有 $m(m\geq 0)$ 个互不相交的有限结点集,每个结点集对应一棵子树。

树的定义说明了树是一种递归的数据结构,即树中包含树。如图 5-4 所示的是一棵空树和一棵一般的树。在图 5-4 中,A 是根结点,有三棵子树,B 是子树的根结点,有两棵子树。树的结点之间有明确的层次关系。

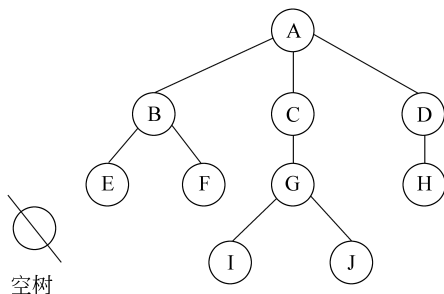


图 5-4 空树和一般的树

通常,企事业单位的行政机构、书的目录结构、人类的家族血缘关系、操作系统的资源管理器以及结构化程序模块之间的关系等都是树状结构。

5.2.2 树的基本术语

下面根据树的特征,分类给出树的基本术语描述。

1. 树的结点

包含一个数据元素及若干指向其子树的分支。

2. 树的规模

树的规模包括结点的度、树的度、结点的层次、树的高度(深度)和树的宽度。

(1) 结点的度: 结点拥有的子树数。如图 5-4 中结点 A 的度为 3, 结点 B 的度为 2, 结点 C 的度为 1, 结点 E 的度为 0。

(2) 树的度: 树中各结点度的最大值。如图 5-4 中的树, 度为 3。

(3) 结点的层次: 根为第一层, 根的孩子为第二层, 以此类推。

(4) 树的深度(高度): 树中结点的最大层次。如图 5-4 中非空树的深度是 4。

(5) 树的宽度: 树中各层的结点数的最大值。如图 5-4 中非空树的宽度是 4。

3. 结点

结点分为叶子(终端结点)、根和内部结点(非终端结点、分支结点)。

(1) 根结点: 无前驱。如图 5-4 中的结点 A。

(2) 叶子(终端结点): 度为零的结点, 无后继。如图 5-4 中的结点 E、F、I、J 和 H。

(3) 分支结点(非终端结点): 度不为零的结点。除根结点外, 分支结点也称内部结点(1个前驱, 多个后继)。如图 5-4 中的结点 B、C、D 和 G。

4. 结点间的关系

根据结点所在层次与其上层、下层以及同层的关系, 结点之间存在以下关系:

(1) 孩子: 结点的子树的根称为该结点的孩子。相应地, 该结点称为孩子的双亲。如图 5-4 中结点 B、C、D 是结点 A 的孩子; 结点 A 是结点 B、C、D 的双亲。

(2) 兄弟: 同一双亲的孩子之间互为兄弟。如图 5-4 中的结点 B、C、D 互为兄弟, 它们共同的双亲是 A。

(3) 祖先: 结点的祖先是根到该结点所经分支上的所有结点。如图 5-4 中的结点 I 的祖先是 A、C、G。

(4) 子孙: 以某结点为根的子树中的任一结点都称为该结点的子孙。如图 5-4 中的结点 C 的子孙是 G、I、J。

5. 兄弟间的关系

根据兄弟间的关系, 一棵树的子树分为无序树和有序树。

如图 5-5 所示的两棵树根结点相同, 构成子树的结点相同, 但子树的顺序不同, 它们是两棵不同的有序树。

森林指 $m(m \geq 0)$ 棵互不相交的树的集合, 任何一棵非空树是一个二元组, 即 $Tree = (root, F1)$, 其中 root 称为根结点, F1 称为子树森林, 如图 5-6 所示。

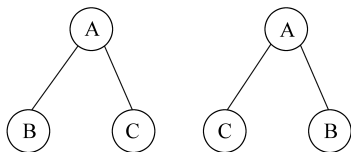


图 5-5 两棵不同的树

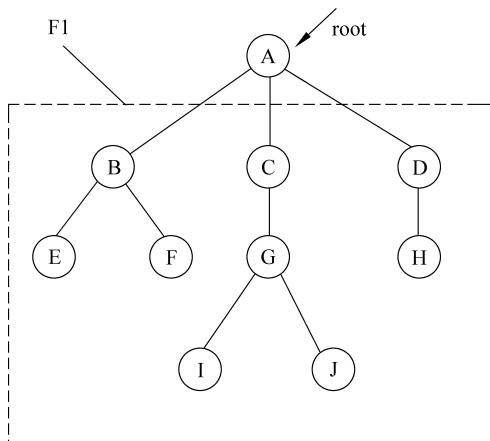


图 5-6 子树森林、森林与树

5.2.3 树的表示

1. 结点连线表示

连线意指上方结点是下方结点的前驱(双亲), 下方结点是上方结点的后继(孩子)。如图 5-7 所示, 结点 A 与结点 B 的连线表示结点 A 是结点 B 的双亲, 结点 B 是结点 A 的孩子。

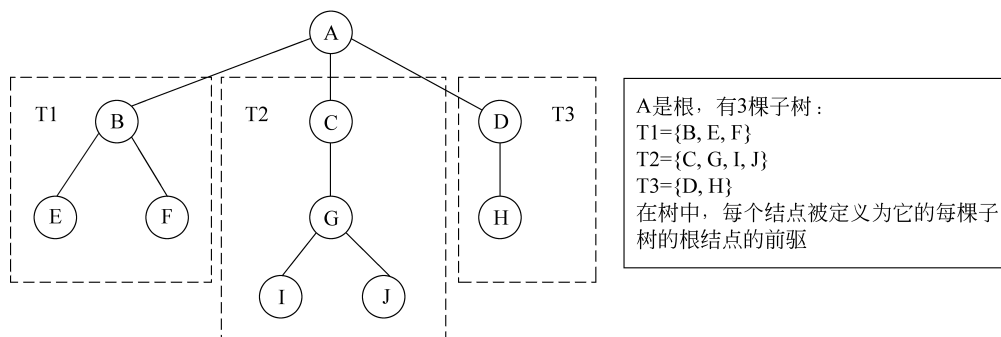


图 5-7 一棵树的结点连线示例

2. 二元组表示

如图 5-7 所示的树可表示为： $T=(D, R)$ ，其中， $D=\{A, B, C, D, E, F, G, H, I, J\}$ ， $R=\{\langle A, B \rangle, \langle A, C \rangle, \langle A, D \rangle, \langle B, E \rangle, \langle B, F \rangle, \langle C, G \rangle, \langle D, H \rangle, \langle G, I \rangle, \langle G, J \rangle\}$ 。

3. 集合图表示

每棵树对应一个圆形，圆内包含根结点和子树。对图 5-6 所示的树，其对应集合图的表示如图 5-8 所示。

4. 凹入表表示

每棵树的根对应一个条形，子树对应的根是一个较短的条形，同一层次的结点对应等长度的条形，这种表示方法常用于打印和屏幕输出。图 5-7 所示的树对应的凹入表如图 5-9 所示。

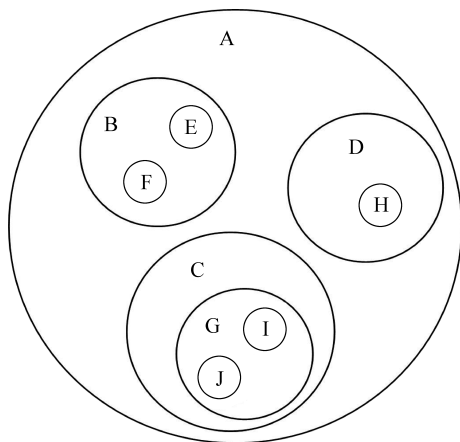


图 5-8 集合表示示例

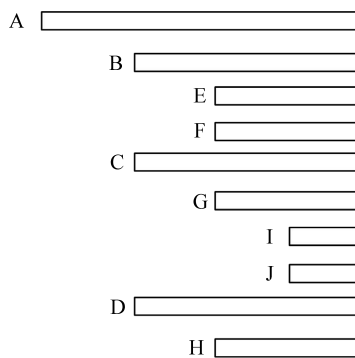


图 5-9 凹入表示例

5. 广义表表示

假设树的根为 $root$ ，子树为 $T1, T2, \dots, Tn$ ，与该树对应的广义表为 L ，则 $L=(原子(子表 1, 子表 2, \dots, 子表 n))$ ，其中原子对应 $root$ ，子表 $i(1 \leq i \leq n)$ 对应 Ti 。图 5-6 所示的树对应的广义表为 $A(B(E, F), C(G(I, J)), D(H))$ 。广义表表示法既可用于输入一

棵树,又可用于显示一棵树。

5.2.4 树的抽象数据类型描述

树的抽象数据类型描述如下。

ADT Tree

{ **数据对象**:具有相同数据类型的数据元素集合。

数据关系:具有一对多的层次关系。

基本操作:

Initiate(&T):初始化空树。

Root(T):求根结点。

Parent(T, x):求当前结点的双亲结点。

Child(T, x, i):求树 T 中结点 x 的第 i 个孩子。

Right_Sibling(T, x):求结点 x 的右兄弟。

Crt_Tree(&T, x, F):以结点 x 为根并以 F 为子树森林构造树 T。

Ins_Tree(&T, y, i, x):将以 x 为根的子树作为结点 y 的第 i 棵子树插入树 T 中。

Del_Child(&T, x, i):将树 T 中结点 x 的第 i 棵子树删除。

Traverse_Tree(T):按某种次序依次访问树中各个结点,并使每个结点只被访问一次。

Clear(&T):清除树结构。

}ADT Tree

由于树状结构的子树个数可多可少,其操作相对复杂。但是树与二叉树可以相互转换,只要掌握了有关二叉树的基本操作,就很容易实现树的基本操作。下面先讨论二叉树。

5.3 二叉树及其应用

5.3.1 二叉树的定义

二叉树可以是空树,或者是由一个根结点加上两棵分别称为左子树和右子树的互不相交的二叉树组成,如图 5-10 所示。

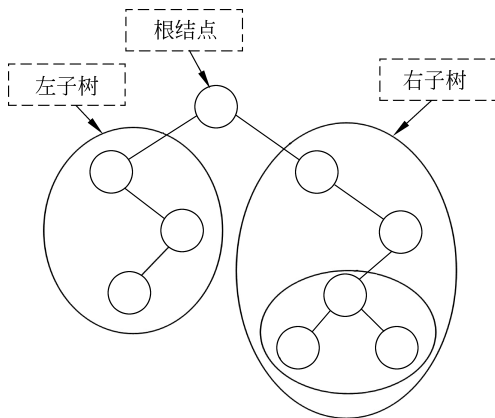


图 5-10 二叉树示例

二叉树的特点是：每个结点至多只有两棵子树，子树有左右之分，其次序不能任意颠倒。

1. 二叉树的五种基本形态

二叉树的五种基本形态如图 5-11 所示。

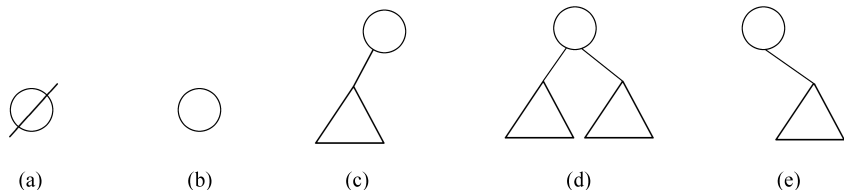


图 5-11 二叉树的五种形态

(a)空二叉树；(b)仅有根结点的二叉树；(c)右子树为空的二叉树；

(d)左右子树均为非空的二叉树；(e)左子树为空的二叉树

2. 三个结点的树

由于树的子树是不区分顺序的，所以只有两种情况，如图 5-12 所示。

3. 三个结点的二叉树

由于二叉树的子树是有序的，三个结点组成的二叉树共有 5 种情况，如图 5-13 所示。

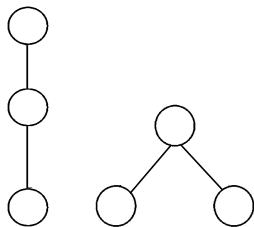


图 5-12 三个结点的树

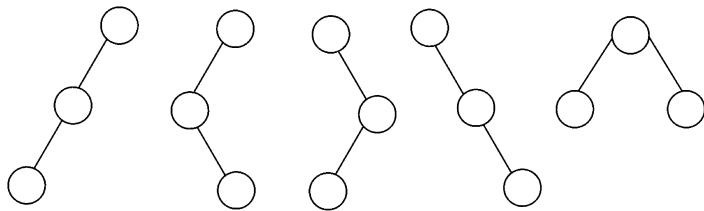


图 5-13 三个结点的二叉树

4. 特殊的二叉树

(1) 满二叉树。二叉树中每一层的结点数都达到最大，所有的叶子结点均在最后一层。图 5-14 所示的是深度为 4、结点数为 15 的满二叉树。

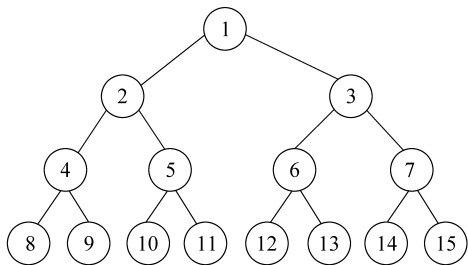


图 5-14 一棵满二叉树

(2) 完全二叉树。

① 对于高度为 k 的完全二叉树，除最后一层外，其余各层都是满的，即前 $k-1$ 层是

一棵满二叉树。

② 完全二叉树的右子树高度与左子树高度相等或少 1。最后一层可以是满的,或者是从最右边开始缺少连续的若干结点。叶子结点只能出现在最后一层或次上一层;度为 1 的结点只能出现在次上一层,一定是只有左子树的结点,如图 5-15 所示。

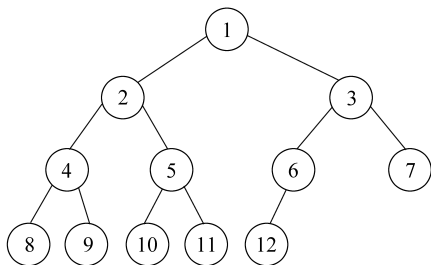


图 5-15 一棵完全二叉树

(3) 理想平衡树。在一棵二叉树中,若除最后一层外其余层都是满的,则称此树是理想平衡树,如图 5-16 所示。

满二叉树和完全二叉树是理想平衡树,但理想平衡树不一定是完全二叉树。

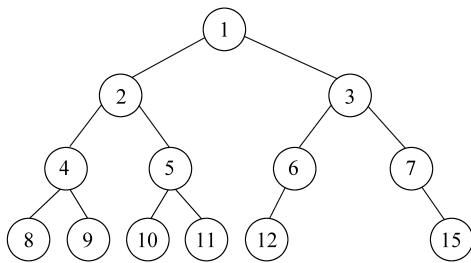


图 5-16 一棵理想平衡树

结论: 二叉树是一种重要的树状结构。

5.3.2 二叉树的性质

性质 1 在二叉树的第 i 层上至多有 2^{i-1} 个结点($i \geq 1$)。

【证明】

(1) $i=1$ 时, $2^{i-1} = 2^0 = 1$, 即只有一个根结点。

(2) 设 $j=i-1$ 时命题成立, 证明 $j=i$ 时命题也成立。

设第 $i-1$ 层上至多有 2^{i-2} 个结点, 每个结点最多引出两个分支, 在第 i 层上的结点数最多为:

$$2^{i-2} \times 2 = 2^{i-1}, \text{命题得证。}$$

进一步推广: m 叉树的第 i 层至多有 m^{i-1} 个结点。

性质 2 一棵深度为 k 的二叉树中, 最多具有 $2^k - 1$ 个结点。

【证明】 二叉树达到最大结点数, 即每一层结点数都需达到最大。设第 i 层的结点

数为 $x_i (1 \leq i \leq k)$, 深度为 k 的二叉树的结点数为 M , 根据性质 1, x_i 最多为 2^{i-1} , 则有:

$$M = \sum_{i=1}^k x_i = \sum_{i=1}^k 2^{i-1} = 2^k - 1$$

进一步推广: 深度为 k 的 m 叉树最多有 $\frac{m^k - 1}{m - 1}$ 个结点。

性质 3 对于一棵非空的二叉树, 如果叶子结点数为 n_0 , 度为 2 的结点数为 n_2 , 则有: $n_0 = n_2 + 1$ 。

【证明】 设 n 为二叉树的结点总数, n_1 为二叉树中度为 1 的结点数, 则有:

$$n = n_0 + n_1 + n_2 \quad (5-1)$$

在二叉树中, 除根结点外, 其余结点都有唯一的一个进入分支。设 B 为二叉树中的分支数, 那么有:

$$B = n - 1 \quad (5-2)$$

这些分支是由度为 1 和度为 2 的结点发出的, 一个度为 1 的结点发出一个分支, 一个度为 2 的结点发出两个分支, 所以有:

$$B = n_1 + 2n_2 \quad (5-3)$$

综合式(5-1)、式(5-2)和式(5-3)式可以得到: $n_0 = n_2 + 1$ 。

性质 4 具有 n 个结点的完全二叉树的深度 k 为 $\lfloor \log_2 n \rfloor + 1$ 或 $\lceil \log_2 (n+1) \rceil$ 。

【证明】 根据完全二叉树的定义和性质 2 可知, 当一棵完全二叉树的深度为 k 、结点个数为 n 时, 有 $2^{k-1} - 1 < n \leq 2^k - 1$ 或 $2^{k-1} \leq n < 2^k$, 如图 5-17 所示。

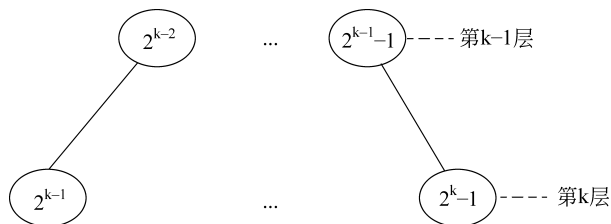


图 5-17 k 层完全二叉树的结点最后两层结点数的示意图

对不等式 $2^{k-1} - 1 < n \leq 2^k - 1$ 的各项 +1, 得到 $2^{k-1} < n+1 \leq 2^k$, 再取以 2 为底的对数, 得到 $k-1 < \log_2 (n+1) \leq k$, 即 $\log_2 (n+1) \leq k < \log_2 (n+1) + 1$ 。由于 k 是整数, 所以有 $k = \lceil \log_2 (n+1) \rceil$, 其中 $\lceil \log_2 (n+1) \rceil$ 表示不小于 $\log_2 (n+1)$ 的一个最小整数, 如图 5-18 所示。

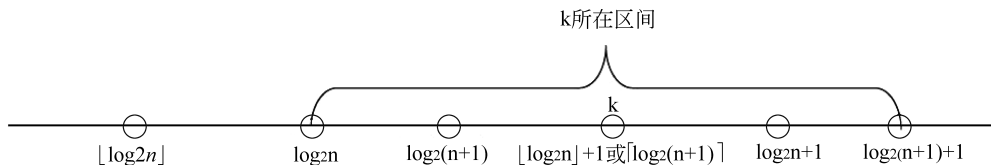


图 5-18 k 的取值区间示意图

对不等式 $2^{k-1} \leq n < 2^k$, 取以 2 为底的对数, 得到 $k-1 \leq \log_2 n < k$, 即 $\log_2 n < k \leq \log_2 n + 1$ 。由于 k 是整数, 所以有 $k = \lfloor \log_2 n \rfloor + 1$, 其中 $\lfloor \log_2 n \rfloor$ 表示不大于 $\log_2 n$ 的一个最大整数, 如图 5-18 所示。

进一步推广: 具有 n 个结点的 m 叉树的最小深度是 $\lfloor \log_m (n(m-1)) \rfloor + 1$ 。

性质 5 对于一棵有 n 个结点的完全二叉树, 其深度为 $\lfloor \log_2 n \rfloor + 1$, 对结点按层序编号(从上到下、从左到右), 则对任一结点 $i (1 \leq i \leq n)$, 有:

- (1) 如果 $i = 1$, 则结点 i 是根。如果 $i > 1$, 则其双亲结点是 $\lfloor i/2 \rfloor$ 。
- (2) 如果 $2i > n$, 则结点 i 为叶子, 否则其左孩子是结点 $2i$ 。
- (3) 如果 $2i+1 > n$, 则结点 i 无右孩子, 否则其右孩子是结点 $2i+1$ 。

【证明】 首先证明(2)和(3), 再导出(1)。

对于任意一棵完全二叉树, 对编号为 i 的结点, 假设它所在的层为 k 。

(1) 如果 k 是第 1 层, $k=1$, i 是根结点的编号, $i=1$ 。如果结点 i 有左孩子, 则左孩子编号是 $2i=2$; 如果有右孩子, 则右孩子的编号是 $2i+1=3$ 。

(2) 如果 k 是最后一层, 结点 i 为叶子结点, 结点 i 既无左孩子, 又无右孩子。

(3) 如果 k 不是最后一层, 假设 i 是第 k 层上所有非叶子结点的最后一个结点的编号, 根据性质 2, 第 k 层上的结点 i 满足: $2^{k-1} \leq i \leq 2^k - 1$, 那么在第 k 层上结点 i 之前的 $(i - 2^{k-1})$ 个结点都是度为 2 的结点, 编号为 i 的结点度为 2 或 1。所以对于第 k 层上的非叶子结点 i , 一定存在左孩子, 左孩子编号为第 $k+1$ 层的第一个结点的编号 2^k 与第 k 层上结点 i 之前度为 2 的结点个数 $(i - 2^{k-1})$ 的 2 倍之和, 即 $2^k + 2(i - 2^{k-1}) = 2i$ 。如果存在右孩子, 右孩子编号为 $2i+1$, 如表 5-1 所示。

表 5-1 完全二叉树第 k 层上的非叶子结点的孩子编号一览表

第 k 层结点编号	2^{k-1} (度为 2)	$2^{k-1} + 1$ (度为 2)	...	i (度为 2 或 1)	...	$2^k - 1$
	左孩子: $2^k = 2 \times 2^{k-1}$ 右孩子: $2^k + 1 = 2 \times 2^{k-1} + 1$	左孩子: $2^k + 2 = 2(2^{k-1} + 1)$ 右孩子: $2^k + 3 = 2(2^{k-1} + 1) + 1$		如果 i 的度为 2 左孩子: $2i$ 右孩子: $2i+1$ 如果 i 的度为 1 左孩子: $2i$		

当 $2i > n$ 时, 编号为 $2i$ 的结点不存在, 说明双亲结点 i 既没有左孩子也没有右孩子, 所以编号为 i 的结点一定是叶子结点。当 $2i \leq n$ 时, 说明编号为 $2i$ 的结点存在, 是编号为 i 的结点的左孩子。

当 $2i+1 > n$ 时, 编号为 $2i+1$ 的结点不存在, 说明双亲结点 i 没有右孩子; 当 $2i+1 \leq n$ 时, 说明编号为 $2i+1$ 的结点存在, 是编号为 i 的结点的右孩子。

由上述结果不难推出, 对任何一个结点 i , 如果 i 是根结点, 那么 i 无双亲, 否则 $\lfloor i/2 \rfloor$ 是 i 的双亲。

5.3.3 二叉树的抽象数据类型

二叉树的抽象数据类型描述如下。