二叉树的顺序存储

```c
#define MaxSize 110

typedef int ElemType;

struct TreeNode{
    ElemType value;//结点中的顺序元素
    bool isEmpty;//结点是否为空
};

TreeNode t[MaxSize];
```

二叉树的链式存储

```c
#define MaxSize 110

typedef int ElemType;

typedef struct BiTNode{
    ElemType value;//结点中的顺序元素
    struct BiTNode *lchild,*rchild;
}BiTNode,*BiTree;
```

若要快速找到父结点,可使用**三叉链表**,即增加父结点指针域

```c
typedef struct BiTNode{
    ElemType value;//结点中的顺序元素
    struct BiTNode *lchild,*rchild;
    struct BiTNode *parent;
}BiTNode,*BiTree;
```

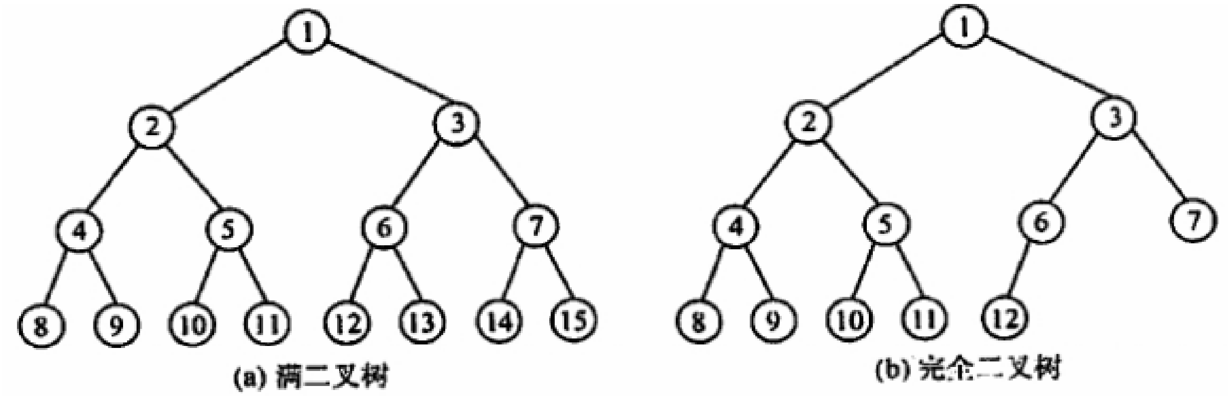
在 n个结点的二叉链表中,每个结点提供两个指针域,共 $2n$个,其中含有 $n+1$个空指针域(在 n个结点的树中有 $n-1$条边,每条边共享一个指针域),此处空指针域会在**线索链表**中使用.

二叉树的基本术语

- 二叉树与度为 2的有序树的区别:

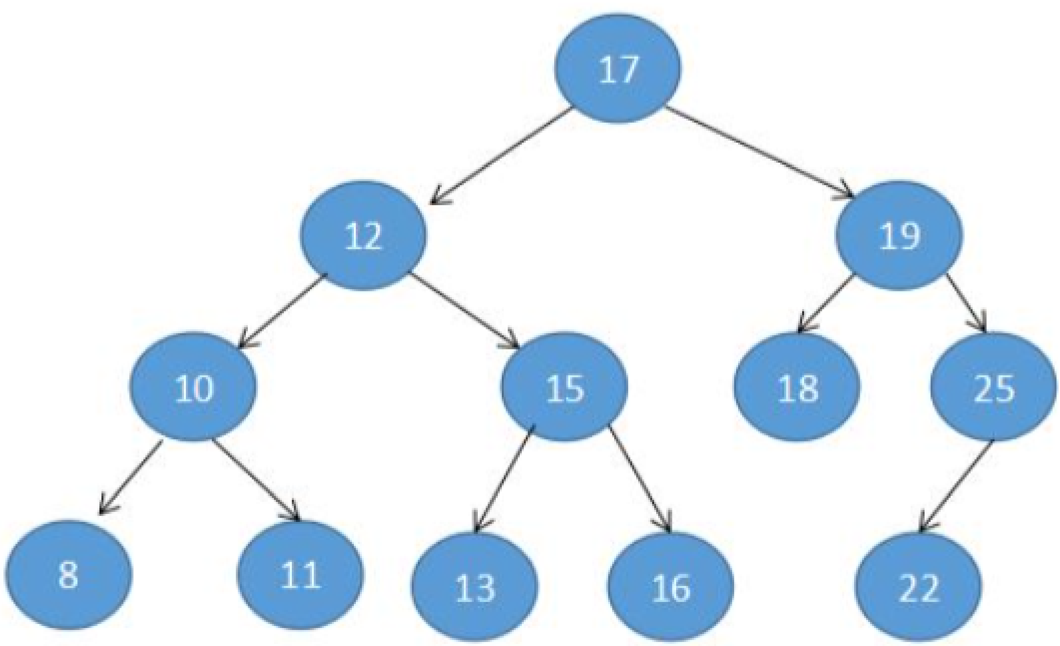
> 度为 2的树至少有 3个结点,而二叉树可以是空树.
>
> 度为 2的有序树的孩子的左右次序是根据相对另一个孩子而言,若某个结点只有一个孩子,则这个孩子就无须区分其左右次序,而二叉树无论其孩子树是否为 2,均需确定其左右次序,即二叉树的结点次序不是相对另一个结点而言,而是确定的.

- **满二叉树**: 一棵高度为 h,且含有 2^h-1个结点的二叉树.满二叉树的叶节点都集中在二叉树的最后一层(第 h层),并且除叶结点之外的每个结点度数均为 2.(它是特殊的**完全二叉树**)
- **完全二叉树**:当且仅当其每个结点都与高度为 h的满二叉树中编号为 $1\sim n$的结点一一对应时，称为完全二叉树.

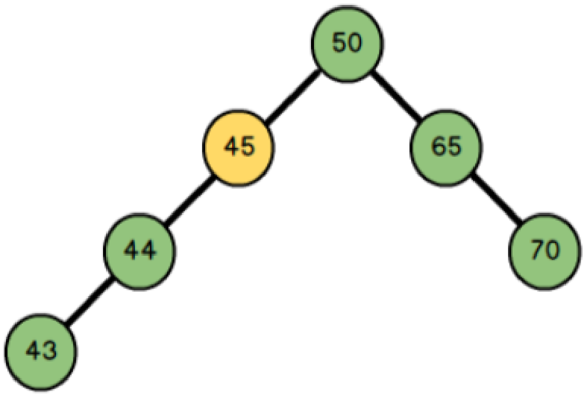


> **满二叉树**的特点:
>
> (1) 只有**最后一层**有叶子结点;
>
> (2) 不存在度为 1的结点;
>
> (3) 按层序从 1开始编号,结点 i的左孩子是 $2 \times i$或者为**空**(此处为空即叶子结点无左孩子),右孩子为 $2 \times i + 1$或者为**空**(此处为空即叶子结点无右孩子),结点 i的父结点为 $\left\lfloor \frac{i}{2} \right\rfloor$或者为**空**(此处为空即根结点无父结点);
>
> **完全二叉树**的特点:
>
> (1) 只有**最后两层可能**有叶子结点;
>
> (2) 最多只有一个度为 1的结点;
>
> (3) 同 满二叉树(3);
>
> (4) 当 $i \leq \left\lfloor \frac{n}{2} \right\rfloor$为分支结点,当 $i > \left\lfloor \frac{n}{2} \right\rfloor$为叶子结点;
>
> (5) 若 n为**奇数**,则每个分支结点都有左孩子和右孩子(即度为 1的结点无);若 n为**偶数**,则编号最大的分支结点(编号为 $\frac{n}{2}$)只有左孩子,没有右孩子,其余分支结点左、右孩子都有(即度为 1的结点有 1个).

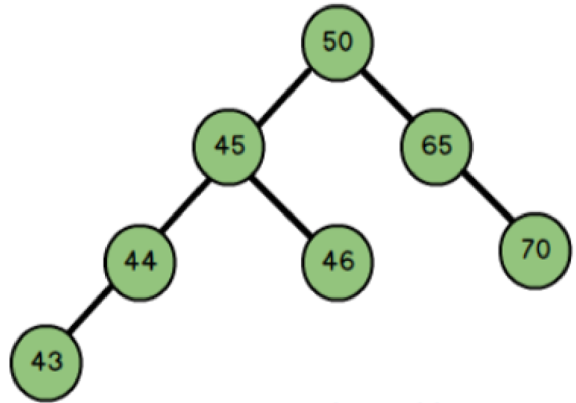
- **二叉排序树**:**左子树**上的所有结点的关键字均**小于**根结点的关键字;**右子树**上的所有结点的关键字均**大于**根结点的关键字;左子树和右子树又各是一棵二叉排序树.



- **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。



非平衡二叉树



平衡二叉树

二叉树的性质

- 非空二叉树的叶结点数等于度为 2的结点数加 1,即 $n_0 = n_2 + 1$.

> 设树中度为 i($i = 0, 1, 2, \cdots, m$)的结点数为 a_i,总结点个数 n为 $\sum_{i=0}^{m} a_i$ (表示结点个数之和),亦可表示为 $(\sum_{i=0}^{m} i \times a_i) + 1$(表示所有结点的度数之和 $+1$)
>
> **证明**:
>
> 套公式得 :
>
> $n = n_0 + n_1 + n_2$
>
> $n = n_0 \times 0 + n_1 \times 1 + n_2 \times 2 + 1$
>
> 联立两式得: $n_0 + n_1 + n_2 = n_1 + n_2 \times 2 + 1$
>
> 整理得: $n_0 = n_2 + 1$

- m叉树第 i层有 m^{i-1}个结点($i \geq 1$),故二叉树第 i层有 2^{i-1}个结点($i \geq 1$).
- 高度为 h的 m**叉树**至多有 $\frac{m^h-1}{m-1}$ 个结点,高度为 h的二叉树至多有 $2^h - 1$个结点(满二叉树).
- 具有 n个($n > 0$)结点的**完全二叉树**的高度 h为 $\lceil \log_2(n+1) \rceil$或 $\lfloor \log_2(n) \rfloor + 1$

> **证明**:
>
> 具有 n个结点的 m叉树的最小高度为 $\lceil \log_m(n \times (m-1) + 1) \rceil$,故具有 n个结点的完全二叉树的高度为 $\lceil \log_2(n \times (2-1) + 1) \rceil$即 $\lceil \log_2(n+1) \rceil$.
>
> 高为 $h-1$的满二叉树共有 $2^{h-1} - 1$个结点,高为 h的完全二叉树至少 2^{h-1}个结点,至多 $2^h - 1$;
>
> 故 $2^{h-1} \leq n < 2^h$
>
> 两边同时取 $\log_2$,得: $h - 1 \leq \log_2 n < h$
>
> 故: $h = \lfloor \log_2 n \rfloor + 1$
>
> **注**:具有 n个结点的**二叉树**(此处是`二叉树`,不是`完全二叉树`)的高度 h 范围为 $\lceil \log_2(n+1) \rceil \leq h \leq n$ (当它是**一条链**(每一个高度只有一个结点)时,高度是 n)

- 对于完全二叉树，可以由的结点数 n 推出度为 0、 1和 2的结点个数为 n_0、 n_1和 n_2.

> 完全二叉树最多只有一个度为 1的结点,即 $n_1=0或1, n_0=n_2+1$.
>
> 由 $n_0=n_2+1$知, n_0+n_2**一定是奇数**;
>
> - 若完全二叉树有 $2k$个(**偶数**)个结点,则必有 $n_1=1, n_0=k, n_2=k-1$.
> - 若完全二叉树有 $2k-1$个(**奇数**)个结点,则必有 $n_1=0, n_0=k, n_2=k-1$.
>
> 完全二叉树最多只会有一个度为 1的结点.

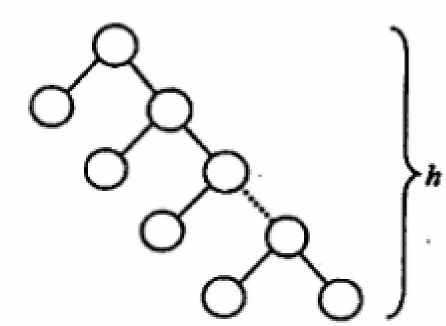
> 具有 10个叶子结点的二叉树中有()个度为 2的结点.
>
> A. 8
>
> B. 9
>
> C. 10
>
> D. 11

非空二叉树的叶结点数等于度为 2的结点数加 1,即 $n_0=n_2+1$.

故度为 2的结点个数为 $n_2=n_0-1=10-1=9$.

> 设高度为 h的二叉树上只有度为 0和度为 2的结点，则此类二叉树中所包含的结点数至少为().
>
> A. h
>
> B. $2h-1$
>
> C. $2h+1$
>
> D. $h+1$

除了根节点那一层以外，每一层至少有二个结点（满足双亲结点的度为 2），因此就是 $1+2(h-1)=2h-1$个结点



> 假设一棵二叉树的结点个数为 50,则它的最小高度是().
>
> A. 4
>
> B. 5
>
> C. 6
>
> D. 7

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

套结论: 结果为 $\lfloor\log_2 50\rfloor+1=5+1=6$

> 设二叉树有 $2n$个结点，且 $m<n$,则不可能存在(C)的结点.
>
> A. n个度为 0
>
> B. $2m$个度为 0
>
> C. $2m$个度为 1
>
> D. $2m$个度为 2

核心公式: $n_0+n_1+n_2=n$(n为总结点数), $n_0=n_2+1$

二叉树有 $2n$(偶数)个结点,故 $n_1=1$(奇数);

(A) 已知 $n+n_1+n_2=2n$且 $n=n_2+1$

由于 $n+n_1+n_2=2n$得 $n_1+n_2=n$, 解得 $n_1=1$,满足条件;

(B) 已知 $2m+n_1+n_2=2n$且 $2m=n_2+1$

故 $4m+n_1-1=2n$,即 $n_1=2n-4m+1$,可能满足条件;

(C) 已知 $n_0+2m+n_2=2n$且 $n_0=n_2+1$

故 $2n_0+2m+1=2n$,即 $n_0=\frac{2n-2m-1}{2}$

由于 $2n-2m-1$为奇数,但 n_0为小数,不满足条件;

(D)已知 $n_0+n_1+2m=2n$且 $n_0=2m+1$

即 $2m+1+n_1+2m=2n$, $n_1=2n-4m-1$,可能满足条件;

> 一个具有 1025个结点的二叉树的高 h为().
>
> A. 11
>
> B. 10
>
> C. $11\sim 1025$
>
> D. $10\sim 1024$

具有 n个结点的**二叉树**(此处是`二叉树`,不是`完全二叉树`)的高度 h 范围为 $\lceil\log_2(n+1)\rceil\le h\le n$(当它是**一条链**(每一个高度只有一个结点)时,高度是 n)

套结论 :一个具有 1025个结点的二叉树的高 h至少 $\lceil\log_2(1025+1)\rceil=11$层,至多 1025层

> 设二叉树只有度为 0和 2的结点，其结点个数为 15,则该二叉树的最大深度为().
>
> A. 4
>
> B. 5
>
> C. 8
>
> D. 9

设高度为 h的二叉树上只有度为 0和度为 2的结点，则此类二叉树中所包含的结点数至少为 $2h-1$

套结论:故 $2h-1=15$,解得 $h=8$.

> 高度为 h的完全二叉树最少有()个结点.
>
> A. 2^h
>
> B. 2^h+1
>
> C. 2^{h-1}
>
> D. 2^h-1

套结论:前 $h-1$层最多有 $2^{h-1}-1$,第 h层至少有 1个结点,故总共 $2^{h-1}-1+1=2^{h-1}$个结点;

> 已知一棵完全二叉树的第 6层(设根为第 1层)有 8个叶结点，则完全二叉树的结点个数最少是().
>
> A. 39
>
> B. 52
>
> C. 111
>
> D. 119

第 6层有叶子结点说明完全二叉树的高度可能是为 6或 7,又由于是**完全二叉树**且要最少,前 5层应为满二叉树(共 $2^5-1=31$个结点),第 6层有 8个叶子结点,故至少共有 $31+8=39$个结点.

> 若一棵深度为 6的完全二叉树的第 6层有 3个叶子结点，则该二叉树共有()个叶子结点.
>
> A. 17
>
> B. 18
>
> C. 19
>
> D. 20

完全二叉树的叶子结点主要集中在最后两层,故第五层可能存在叶子结点,计算出第 5层有 $2^{5-1}=16$个结点,在第 6层的 3个叶子结点的双亲结点为第 5层的最左边的两个结点,第五层剩余的结点有 $16-2=14$,故共有 $14+3=17$个叶子结点.

> 一棵完全二叉树上有 1001个结点，其中叶结点的个数是().
>
> A. 250
>
> B. 500
>
> C. 254
>
> D. 501

核心公式: $n_0+n_1+n_2=n$(n为总结点数), $n_0=n_2+1$

有题知,完全二叉树上有 1001(**奇数**)个结点,故 $n_1=0$;

故 $n_0+n_2=1001$,代入得, $n_0+n_0-1=1001$,解得, $n_0=501$

> 若一棵二叉树有 126个结点，在第 7层(根结点在第 1层)至多有()个结点.
>
> A. 32

> B. 64
>
> C. 63
>
> D. 不存在第 7层

假设前 6层是一个满二叉树,总共有 $2^6-1=63$个结点,故必然存在第 7层,而如果是满二叉树,第 7 层最多可以有 $2^6=64$个结点,但第七层剩余 $126-63=63$个结点,故第七层最多有 63个结点.

> 一棵有 124个叶子结点的完全二叉树，最多有()个结点.
>
> A. 247
>
> B. 248
>
> C. 249
>
> D. 250

核心公式: $n_0+n_1+n_2=n$(n为总结点数), $n_0=n_2+1$

已知 $n_0=124$,故 $n_2=n_0-1=123$

当 $n_1=1$时,总结点结果最多,总数为 $124+123+1=248$.

> 一棵有 n个结点的二叉树采用二叉链存储结点，其中空指针数为()
>
> A. n
>
> B. $n+1$
>
> C. $n-1$
>
> D. $2n$

$\text{非空指针数}=\text{总分支数}=n-1$,且 $\text{空指针数}=2\times\text{结点总数}-\text{非空指针数}$

故结果为 $2\times n-(n-1)=n+1$.

> 在一棵完全二叉树中，其根的序号为 1,()可判定序号为 p和 q的两个结点是否在同一层.
>
> A. $\lfloor\log_2 p\rfloor=\lfloor\log_2 q\rfloor$
>
> B. $\log_2 p=\log_2 q$
>
> C. $\lfloor\log_2 p\rfloor+1=\lfloor\log_2 q\rfloor$
>
> D. $\lfloor\log_2 p\rfloor=\lfloor\log_2 q\rfloor+1$

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

套结论,故 $\lfloor\log_2 p\rfloor+1=\lfloor\log_2 q\rfloor+1$,整理得: $\lfloor\log_2 p\rfloor=\lfloor\log_2 q\rfloor$

> 假定一棵三叉树的结点数为 50,则它的最小高度为().
>
> A. 3
>
> B. 4
>
> C. 5
>
> D. 6

具有 n个结点的 m叉树的最小高度为 $\lceil\log_m(n\times(m-1)+1)\rceil$;

由上知, $\lceil\log_3(50\times 2+1)\rceil=\lceil\log_3 101\rceil=5$

> 对于一棵满二叉树，共有 n个结点和 m个叶子结点，高度为 h,则()
>
> A. $n=h+m$
>
> B. $n+m=2h$
>
> C. $m=h-1$
>
> D. $n=2^h-1$

高度为 h的 m**叉树**至多有 $\frac{m^h-1}{m-1}$个结点,高度为 h的二叉树至多有 2^h-1个结点(满二叉树).

> **2009统考真题**：已知一棵完全二叉树的第 6层(设根为第 1层)有 8个叶结点，则该完全二叉树的结点个数最多是().
>
> A. 39
>
> B. 52
>
> C. 111
>
> D. 119

第 6层有叶子结点说明完全二叉树的高度可能是为 6或 7,又由于是**完全二叉树**且要最多,故该完全二叉树最高有 7层,说明第 6层的最后 8个叶子结点集中在最右边,而第 6层的剩余结点为分支结点(共 $2^5-8=24$个),第 7层有 $24\times 2=48$个叶子结点,前 6层为满二叉树共有 $2^6-1=63$个,故总结点最多为 $63+48=111$个.

> **2011统考真题**：若一棵完全二叉树有 768个结点，则该二叉树中叶结点的个数是().
>
> A. 257
>
> B. 258
>
> C. 384
>
> D.385

核心公式: $n_0+n_1+n_2=n$(n为总结点数), $n_0=n_2+1$

有题知,完全二叉树上有 768(**偶数**)个结点,故 $n_1=1$;

故 $n_0+1+n_0-1=768$,解得: $n_0=384$

> **2018统考真题**：设一棵非空完全二叉树 T的所有叶结点均位于同一层，且每个非叶结点都有 2个子结点。若 T有 k个叶结点，则 T的结点总数是().
>
> A. $2k-1$
>
> B. $2k$
>
> C. k^2
>
> D. 2^k-1

由每个非叶结点(**分支结点**)都有 2个子结点,故度为 1的结点数为 0,由因为度为 0的结点数为 k且 $n_0=k=n_2+1$,计算得度为 2的结点有 $k-1$个,故总结点数为 $n_0+n_1+n_2=k+0+k-1=2k-1$.

> **2020统考真题**：对于任意一棵高度为 5且有 10个结点的二叉树，若采用顺序存储结构保存，每个结点占 1个存储单元(仅存放结点的数据信息),则存放该二叉树需要的存储单元数量至少是().
>
> A. 31
>
> B. 16
>
> C. 15
>
> D. 10

虽然只有 10个结点,但仍需要存储一棵高度为 5的满二叉树,共 $2^5-1=31$个结点,每个结点占 1个存储单元,即存放该二叉树需要的存储单元数量至少是 31个.

5.2.1

> 在一棵完全二叉树中，含有 n_0个叶子结点，当度为 1的结点数为 1时，该树的高度是多少?当度为 1的结点数为 0时，该树的高度是多少?

- 具有 n个($n>0$)结点的**完全二叉树**的高度 h为 $\lceil\log_2(n+1)\rceil$或 $\lfloor\log_2(n)\rfloor+1$

已知 $n_0=n_2+1$,故 $n_2=n_0-1$,总结点数为 $n_0+n_1+n_2=2n_0+n_1-1$

当 $n_1=1$时,总结点数为 $n=2n_0$,故高度为 $\lceil\log_2(2n_0+1)\rceil$或 $\lfloor\log_2(2n_0)\rfloor+1$;

当 $n_1=0$时,总结点数为 $n=2n_0-1$,故高度为 $\lceil\log_2(2n_0-1+1)\rceil=\log_2 2+\lceil\log_2(n_0)\rceil=\lceil\log_2(n_0)\rceil+1$或 $\lfloor\log_2(2n_0-1)\rfloor+1$

5.2.2

> 一棵有 n个结点的满二叉树有多少个分支结点和多少个叶子结点?该满二叉树的高度是多少?

在满二叉树中无度为 1的结点,故 $n_1=0$,又因为 $n_0=n_2+1$且 $n=n_0+n_2=2n_0-1$,

故叶子结点树为 $n_0=\frac{n+1}{2}$,分支结点为 $n-n_0=n-\frac{n+1}{2}=\frac{n-1}{2}$,高度为 $\lceil\log_2(n+1)\rceil=\log_2(n+1)$(此处由于 $n+1$必定是 2的指数倍,可去掉上取整符)或 $\lfloor\log_2(n)\rfloor+1$

5.2.3

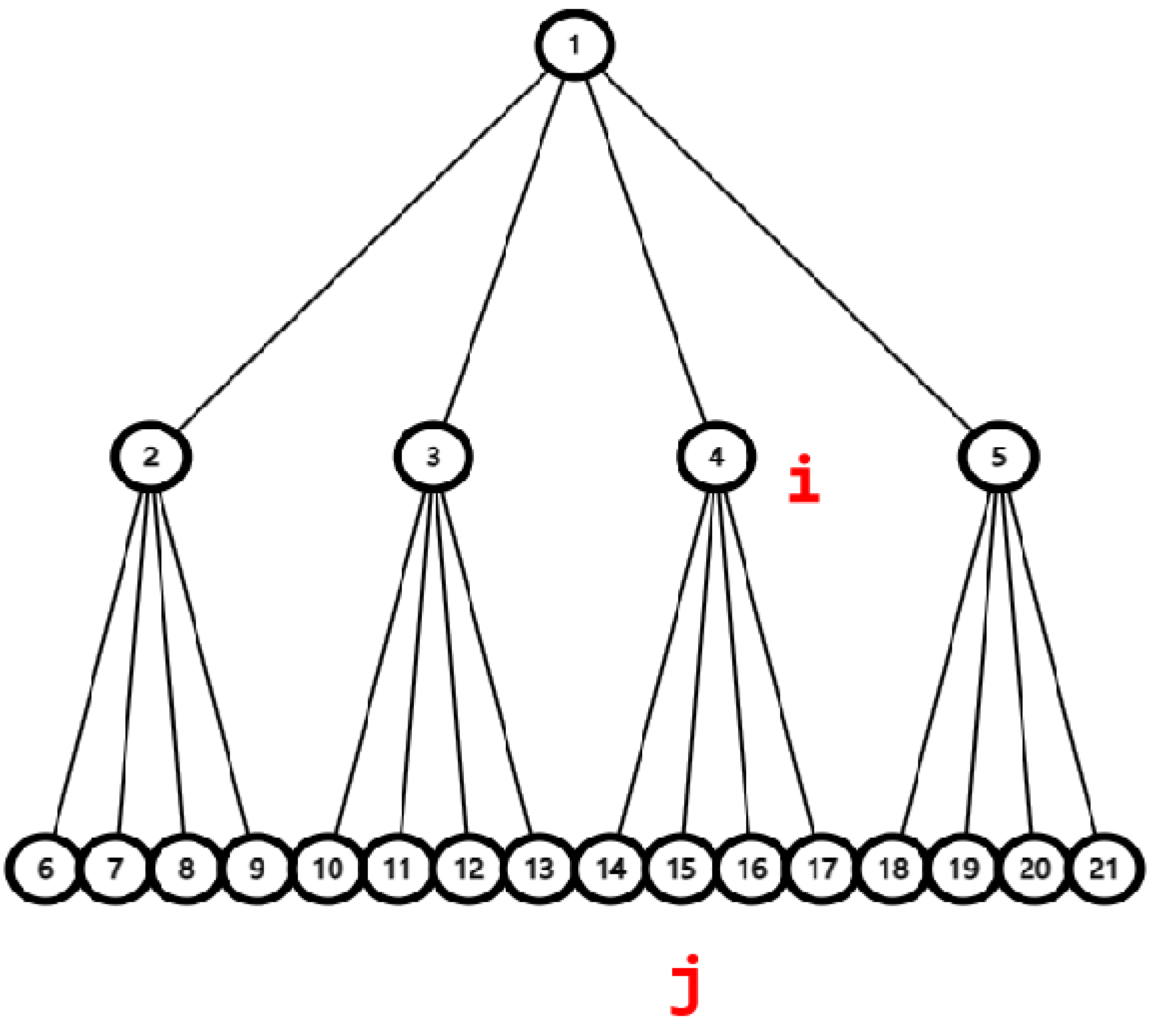
> 已知完全二叉树的第 9层有 240个结点，则整个完全二叉树有多少个结点?有多少个叶子结点?

在完全二叉树中，若第 9层是满的，则结点数 $=2^8=256$,而现在第 9层只有 240(<256)个结点,说明第 9层未满，是最后一层。$1\sim 8$层是满的(共有 $2^8-1=255$个结点)，所以总结点数 $=2^8-1+240=255+240=495$.

因为第 9层是最后一层，所以第 9层的结点都是叶子结点。且第 9层的 240个结点的双亲在第 8层中，其双亲个数为 120,即第 8层有 120个分支结点，其余为叶子结点，所以第 8层的叶子结点个数为 $2^7-120=128-120=8$。因此，总的叶子结点个数 $=8+240=248$.

5.2.4

> 一棵高度为 h的满 m叉树有如下性质：根结点所在层次为第 1层，第 h层上的结点都是叶结点，其余各层上每个结点都有 m棵非空子树，若按层次自顶向下，同一层自左向右，顺序从 1开始对全部结点进行编号，试问：
>
> 1)各层的结点个数是多少?
>
> 2)编号为 i的结点的双亲结点(若存在)的编号是多少?
>
> 3)编号为 i的结点的第 k个孩子结点(若存在)的编号是多少?
>
> 4)编号为 i的结点有右兄弟的条件是什么?其右兄弟结点的编号是多少?



(1)

- 第 1层有 $m^0 = 1$个结点;
- 第 2层有 $m^1 = m$个结点;
- 第 3层有 m^2个结点;
- $\cdots$
- 第 i层有 m^{i-1}个结点;

(2)

设 i的位置如上图所示, j是 i的第一个子结点;

i的前面有 $i-1$个结点,每一个结点有 m个子结点,故 j前面有 $m\times(i-1)+1$(**为什么要加** 1,由于 $m\times(i-1)$不包括根结点 1,需加上根结点);

故 j的编号为 前面的结点数 $+1$,即 $j=m\times(i-1)+2$

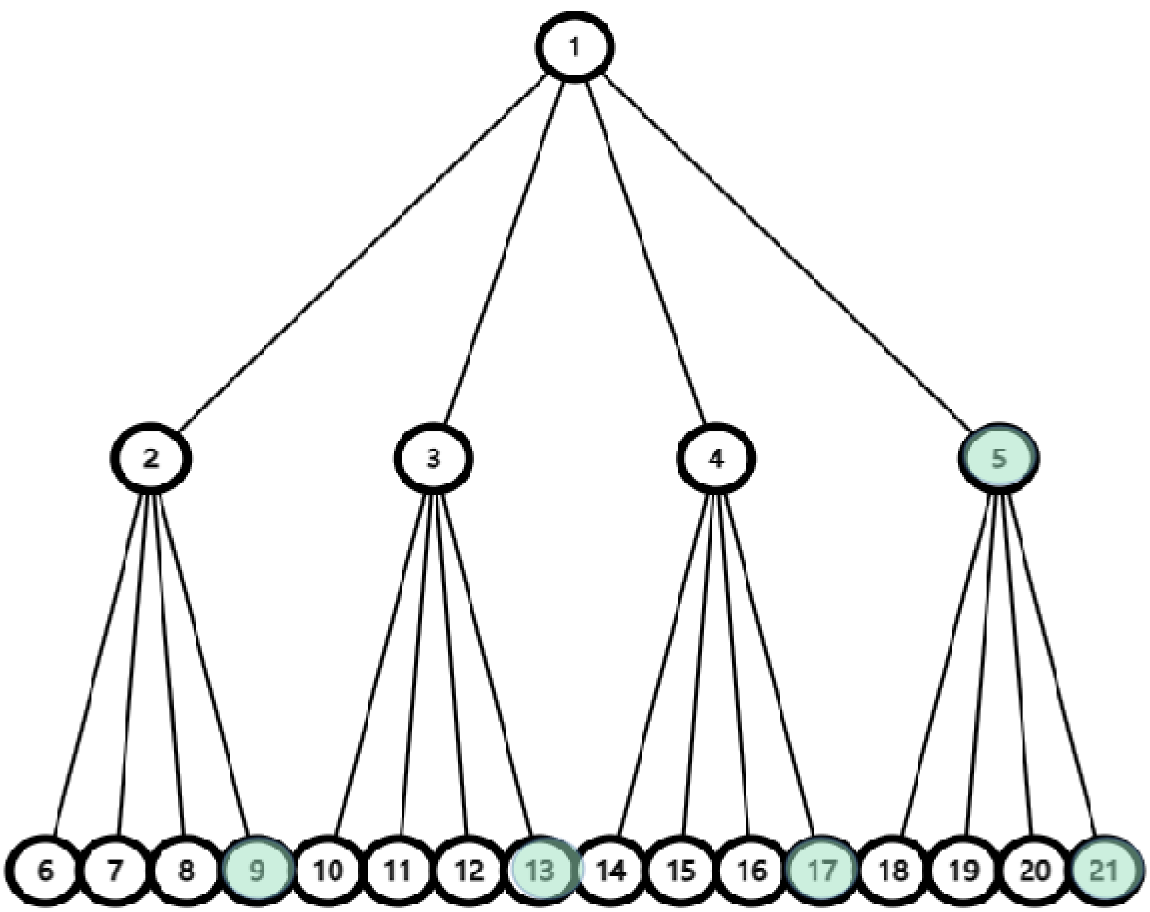
已知 $j=m\times(i-1)+2$,反过来求 j的双亲结点 i为 $\left\lfloor\frac{j-2}{m}\right\rfloor+1$

(3)

若求结点 i的的第 k个孩子

- 结点 i的第 1个孩子: $m\times(i-1)+2$
- 结点 i的第 2个孩子: $m\times(i-1)+2+1$
- $\cdots$
- 结点 i的第 k个孩子: $m\times(i-1)+2+k-1=m\times(i-1)+k+1$ ($1\le k\le m$)

(4)



其中每个绿色的结点为没有右兄弟的结点,当 i不是**双亲结点的最右结点**都是有右兄弟的结点.

在 4叉树中 $5,9,13,17,21,\cdots,4\times k+1$等没有右兄弟;

在 m叉树中 $m \times k+1$等没有右兄弟,故 $i \neq m \times k+1$,变形得, $i-1 \neq m \times k$

化简得: $(i-1) \mod k \neq 0$

当 $(i-1) \mod k \neq 0$时, i有右结点,右兄弟结点编号为 $i+1$.

4）结点 i 不是其双亲的第 m 个子女时才有右兄弟。设其双亲编号为 j，可得 $j=\lfloor(i+m-2)/m\rfloor$，结点 j 的第 m 个子女的编号为$(j-1)m+m+1=jm+1=\lfloor(i+m-2)/m\rfloor \times m+1$，所以当结点的编号 $i \leq \lfloor(i+m-2)/m\rfloor \times m$ 时才有右兄弟，右兄弟的编号为 $i+1$。或者，对于任一双亲结点 j，其第 m 个子女结点的编号是 $jm+1$，故若不为第 m 的子女结点，则$(i-1)\%m!=0$。

5.2.5

> 已知一棵二叉树按顺序存储结构进行存储，设计一个算法，求编号分别为 i和 j的两个结点的最近的公共祖先结点的值。

(1)若 $i>j$,则结点 i所在层次大于等于结点 j所在层次。结点 i的双亲结点为结点 $\frac{i}{2}$,若 $\frac{i}{2}=j$则结点 $\frac{i}{2}$是原结点 i和结点 j的最近公共祖先结点，若 $\frac{i}{2} \neq j$,则令 $i=\frac{i}{2}$,即以该结点 i的双亲结点为起点，采用递归的方法继续查找。

(2)若 $j>i$,则结点 j所在层次大于等于结点 i所在层次。结点 j的双亲结点为结点 $\frac{j}{2}$,若 $\frac{j}{2}=i$,则结点 $\frac{j}{2}$是原结点 i和结点 j的最近公共祖先结点，若 $\frac{j}{2} \neq i$,则令 $j=\frac{j}{2}$。

重复上述过程，直到找到它们最近的公共祖先结点为止。

```
bool Search_Common_Ancestor(TreeNode T[],int i,int j,ElemType &e)
{
    if(T[i].isEmpty!=true&&T[j].isEmpty!=true)
    {
        while(i!=j)
        {
            if(i>j)
                i/=2;
            else j/=2;
        }
        e=T[i].value;
        return true;
    }

    return false;
}
```