

树和二叉树


深度为 h 的二叉树至多有

$2^h - 1$ 个结点。

(仅供课后复习使用, 不得外传)

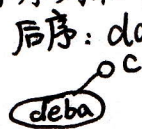
一、填空题

1. 设根的层数为 1, 若深度为 h 的完全二叉树具有最少的结点, 则编号最小和最大的叶子编号分别是 $2^{h-2} + 1$, 2^{h-1} ; n 个结点的完全二叉树的高度为

多少 $\lfloor \log_2 n \rfloor + 1$. hint:  前 $h-1$ 层为满二叉树, h 层只有一个结点。

2. 一棵二叉树的第 i ($i \geq 1$) 层上最多有 2^{i-1} 个结点。一棵高为 h 的满二叉树具有 $2^h - 1$ 个结点。一棵有 n ($n > 0$) 个结点的满二叉树共有 $2^{\lfloor \log_2 n \rfloor}$ 个叶子和 $2^{\lfloor \log_2 n \rfloor} - 1$ 个非终端结点。注意: 后两空书错, 满二叉树高度为 $\lfloor \log_2 n \rfloor + 1$

3. 已知一棵二叉树的后序序列和中序序列分别为 $dabec$ 和 $debac$, 则它的前序序列为 $cedba$ 。



4. 二叉树的前序序列中, 任意一结点均处在其子女结点之前, 而后序序列中任意结点均处在其子女节点之后, 此说法 正确。(正确或错误)。

5. 若哈夫曼树中有 m 个叶子, 则该树结点总数为 $2m - 1$ 。

6. 设二叉树 T 的所有结点的左、右子树交换后得到的二叉树为 T' , 则 T 和 T' 的中序序列的关系是 互为逆序列。(中序遍历遵从投影法, 可以用投影法去理解)。

7. 二叉树由 根结点, 左子树, 右子树 三个基本单元组成。

8. 树在计算机内的表示方式有 双亲链表表示法, 孩子链表表示法, 孩子兄弟表示法。

9. 在二叉树中, 指针 p 所指结点为叶子结点的条件是 $p \rightarrow lchild == \text{NULL} \ \&\& \ p \rightarrow rchild == \text{NULL}$ 。

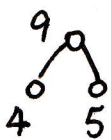
10. 具有 256 个结点的完全二叉树的深度为 9 ($\lfloor \log_2 n \rfloor + 1 = \lfloor \log_2 256 \rfloor + 1 = 8 + 1 = 9$)

11. 二叉树的前序序列和中序序列相同的条件是 空树或右单链树。

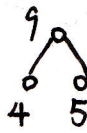
12. 哈夫曼树是: 带权路径长度最小的二叉树, 亦称最优二叉树。

13. 若以 $\{4, 5, 6, 7, 8\}$ 作为叶子结点的权值构造哈夫曼树, 则其带权路径长度为:

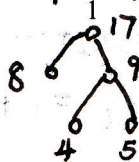
(1)



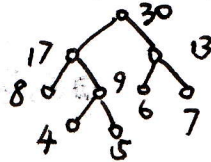
(2) $\{6, 7, 8, 9\}$



(3) $\{8, 9, 13\}$

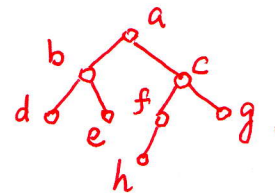
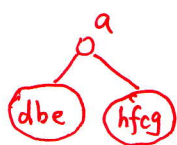


(4) $\{13, 17\}$



$$\begin{aligned} WPL &= 8 \times 2 + 4 \times 3 \\ &\quad + 5 \times 3 + 6 \times 2 + 7 \times 2 \\ &= 16 + 12 + 15 + 12 + 14 \\ &= 69 \end{aligned}$$

69.



14. 已知一棵二叉树的前序序列为 abdecfg^{hg}, 中序序列为 dbeahfcg, 则该二叉树的根为 a, 左子树中有 dbe, 右子树中有 hfcg.

二、判断 (判断下列概念的正确性, 并作出简要的说明。)

(X) 二叉树是度为 2 的有序树。 区别清见课件 二叉树 P10.

(X) 对于有 n 个结点的二叉树, 其高度为 $\log_2 n$ 。 完全二叉树 $\leq \lfloor \log_2 n \rfloor + 1$

(X) 完全二叉树的前序序列中, 若结点 u 在结点 v 之前, 则 u 一定是 v 的祖先。
反例:

(V) 完全二叉树中, 若一个结点没有左孩子, 则它必是树叶。 从定义去理解。

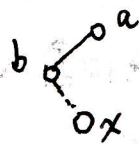
(X) 一棵有 n 个结点的二叉树, 从上到下, 从左到右用自然数依次给予编号, 则编号为 i 的结点的左儿子的编号为 $2i (2i \leq n)$, 右儿子是 $2i+1 (2i+1 \leq n)$ 。 只对完全二叉树适用。

(V) 给定一棵树, 可以找到唯一的一棵二叉树与之相对应。

(V) 树形结构中元素之间存在一对多的关系。

(V) 树与二叉树是两种不同的树型结构。

(V) 非空二叉树一定满足: 某结点若有左孩子, 则其中序前驱一定没有右孩子。
在中序序列中, 若 $\dots ba \dots$ b 是 a 的前驱, 那么 b 必没有右孩子。否则序列为 bxa



三、选择题

1. 具有 3 个结点的树有 B 种不同的形态, 具有 3 个结点的二叉树有 E 种不同形态。
树: 二叉树:

A. 1 B. 2 C. 3 D. 4 E. 5 F. 6

2. 在一棵度为 4 的树中, 已知度为 4, 3, 2, 1 的结点个数分别为 2, 3, 4, 5, 则叶子结点个数数为 A。
设树中结点总数为 n , 叶子 (度为 0) 数目为 n_0 , 则存在 $n = n_0 + n_1 + \dots + n_m$ (度为 m 的树) ①
此外, 除了根外, 每一个结点必是其双亲的孩子, 度为 i 的结点必有 i 个孩子, 故孩子结点总数为 $n_1 + 2n_2 + \dots + mn_m$ ②

A. 17 B. 18 C. 19 D. 20

3. 存在这样的二叉树, 其前序, 中序, 后序序列均相同, 此说法 A。

$$n_1 + 2n_2 + \dots + mn_m + 1 = n \quad ②$$

$$\text{由 } ①, ② \Rightarrow n_0 = 1 + \sum_{i=2}^m (i-1)n_i$$

A. 正确 B. 错误 只有一个结点的情形

4. 在完全二叉树中, 若某结点 p 无左孩子, 则 C。

A. p 一定有右孩子 B. p 可能有右孩子 C. p 必为叶子 D. p 是分支结点

5. 下述陈述正确的是 A。

A. 满二叉树一定是完全二叉树;
B. 完全二叉树一定是满二叉树;
C. 严格的二叉树一定是完全二叉树;

6. 一棵 124 个叶子结点的完全二叉树最多有 B 个结点。 二叉树满足 $n_0 = n_2 + 1$

A. 247 B. 248 C. 249 D. 250

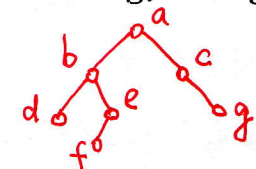
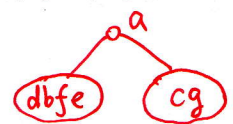
完全二叉树最多只有 1 个度为 1 的结点, 故 $n_0 + n_1 + n_2 = 124 + 1 + 123 = 248$

7. 在以下四种树的存储结构中, 容易实现查找指定结点的祖先和后代的是 C。

A. 双亲链表表示法 B. 孩子链表表示法
C. 双亲孩子链表表示法 D. 孩子兄弟链表表示法

8. 已知某二叉树的前序, 中序和后序序列分别是 abdefc, dbfeacg, dfebgca, 该

二叉树的三个叶子是 A。



A. dfg B. abc C. bde D. bcg

9. 一棵完全二叉树上有 1001 个结点, 其中叶子结点的个数为 E。

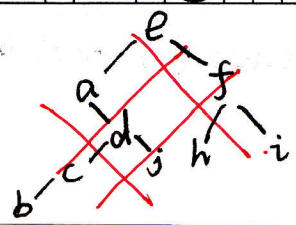
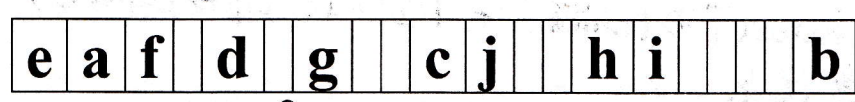
A. 250; B. 500; C. 254; D. 505; E. 以上答案都不对

$$\begin{aligned} n &= n_0 + n_1 + n_2 \\ &= n_0 + n_1 + (n_0 - 1) \\ &= 2n_0 + n_1 - 1 = 1001 \end{aligned}$$

四、简答题

10. 二叉树结点数采用顺序存储结构, 如图所示,

- 画出二叉树表示
- 写出前序遍历, 中序遍历和后序遍历的结果
- 写出结点值 c 的父结点, 其左、右孩子。

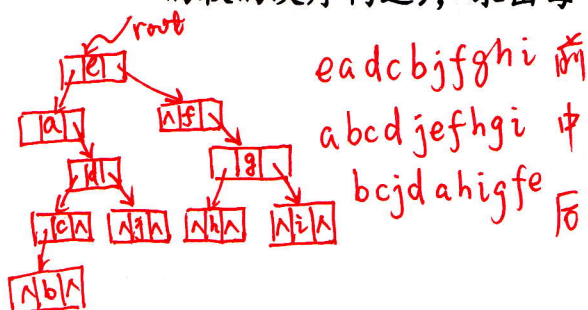


$1002 = 2n_0 + n_1$
由于在完全二叉树中, n_1 只能取 0 或 1
故本题中 n_1 只能为 0.

$$n_0 = 1002 / 2 = 501$$

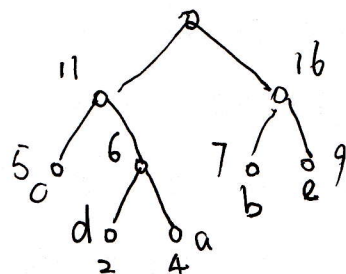
见下页

■ 有一份电文中共使用五个字符：a、b、c、d、e，它们的出现频率依次为 4、7、5、2、9，试画出对应的 Huffman 树（请按左子树根结点的权小于等于右子树根结点的权的次序构造），求出每个字符的 Huffman 编码。



eadcbjfgghi 前
abcdjefhgi 中
bcjda h i g f e 后

a: 011;
b: 10;
c: 00
d: 010;
e: 11



■ 设给定权集 $w=\{2, 3, 4, 7, 8, 9\}$ ，试构造关于 w 的一棵哈夫曼树，并求其加权路径长度 WPL。

WPL=80

五、算法设计题

1. 写一递归算法求二叉树中度为 2 的结点总数。

分析：1) 当 T 为空或 T 是叶子时，以 T 为根的二叉树的 2 度结点数为 0；
2) 当 T 是 2 度结点时，以 T 为根的二叉树的 2 度结点数为 T 的左右子树中 2 度结点数之和再加上 T 结点本身。
3) 当 T 是 1 度结点时，以 T 为根的二叉树的 2 度结点数为 T 的左右子树中 2 度结点数之和。

```

递归： int D2Nodes(BinTree T){
    if (!T || !T->lchild && !T->rchild) // T 为空或 T 是叶子
        return 0;
    if (T->lchild && T->rchild) // T 是 2 度结点
        return 1 + D2Nodes(T->lchild) + D2Nodes(T->rchild);
    else // T 是 1 度结点
        return D2Nodes(T->lchild) + D2Nodes(T->rchild);
}
    
```