

一、下面是有关二叉树的叙述，请判断正误（）

- ( ) 1. 若二叉树用二叉链表作存储结构，则在  $n$  个结点的二叉树链表中只有  $n-1$  个非空指针域。
- ( ) 2. 二叉树中每个结点的两棵子树的高度差等于 1。
- ( ) 3. 二叉树中每个结点的两棵子树是有序的。
- ( ) 4. 二叉树中每个结点有两棵非空子树或有两棵空子树。
- ( ) 5. 二叉树中每个结点的关键字值大于其左非空子树（若存在的话）所有结点的关键字值，且小于其右非空子树（若存在的话）所有结点的关键字值。
- ( ) 6. 二叉树中所有结点个数是  $2^{k-1}-1$ ，其中  $k$  是树的深度。
- ( ) 7. 二叉树中所有结点，如果不存在非空左子树，则不存在非空右子树。
- ( ) 8. 对于一棵非空二叉树，它的根结点作为第一层，则它的第  $i$  层上最多能有  $2^i-1$  个结点。
- ( ) 9. 用二叉链表法 (link-rlink) 存储包含  $n$  个结点的二叉树，结点的  $2n$  个指针区域中有  $n+1$  个为空指针。
- ( ) 10. 具有 12 个结点的完全二叉树有 5 个度为 2 的结点。

二、填空（）

- 1. 由 3 个结点所构成的二叉树有\_\_\_\_\_种形态。
- 2. 一棵深度为 6 的满二叉树有\_\_\_\_\_个分支结点和\_\_\_\_\_个叶子。
- 3. 一棵具有 2 5 7 个结点的完全二叉树，它的深度为\_\_\_\_\_。
- 4. 设一棵完全二叉树有 700 个结点，则共有\_\_\_\_\_个叶子结点。
- 5. 设一棵完全二叉树具有 1000 个结点，则此完全二叉树有\_\_\_\_\_个叶子结点，有\_\_\_\_\_个度为 2 的结点，有\_\_\_\_\_个结点只有非空左子树，有\_\_\_\_\_个结点只有非空右子树。
- 6. 一棵含有  $n$  个结点的  $k$  叉树，可能达到的最大深度为\_\_\_\_\_，最小深度为\_\_\_\_\_。
- 7. 二叉树的基本组成部分是：根 (N)、左子树 (L) 和右子树 (R)。因而二叉树的遍历次序有六种。最常用的是三种：前序法（即按 N L R 次序），后序法（即按\_\_\_\_\_次序）和中序法（也称对称序法，即按 L N R 次序）。这三种方法相互之间有关联。若已知一棵二叉树的前序序列是 BEFCGDH，中序序列是 FEBGCHD，则它的后序序列必是\_\_\_\_\_。
- 8. 中序遍历的递归算法平均空间复杂度为\_\_\_\_\_。
- 9. 用 5 个权值 {3, 2, 4, 5, 1} 构造的哈夫曼 (Huffman) 树的带权路径长度是\_\_\_\_\_。

### 三、选择题 ( )

- ( ) 1. 不含任何结点的空树\_\_\_\_\_。  
(A) 是一棵树; (B) 是一棵二叉树;  
(C) 是一棵树也是一棵二叉树; (D) 既不是树也不是二叉树
- ( ) 2. 二叉树是非线性数据结构, 所以\_\_\_\_\_。  
(A) 它不能用顺序存储结构存储; (B) 它不能用链式存储结构存储;  
(C) 顺序存储结构和链式存储结构都能存储; (D) 顺序存储结构和链式存储结构都不能使用
- ( ) 3. 具有  $n(n>0)$  个结点的完全二叉树的深度为\_\_\_\_\_。  
(A)  $\lceil \log_2(n) \rceil$  (B)  $\lfloor \log_2(n) \rfloor$  (C)  $\lfloor \log_2(n) \rfloor + 1$  (D)  $\lceil \log_2(n) + 1 \rceil$
- ( ) 4. 把一棵树转换为二叉树后, 这棵二叉树的形态是\_\_\_\_\_。  
(A) 唯一的 (B) 有多种  
(C) 有多种, 但根结点都没有左孩子 (D) 有多种, 但根结点都没有右孩子

5. 树是结点的有限集合, 它 A 根结点, 记为 T。其余的结点分成为  $m (m \geq 0)$  个 B 的集合  $T_1, T_2, \dots, T_m$ , 每个集合又都是树, 此时结点 T 称为  $T_i$  的父结点,  $T_i$  称为 T 的子结点 ( $1 \leq i \leq m$ )。一个结点的子结点个数为该结点的 C。

供选择的答案

- A: ①有 0 个或 1 个 ②有 0 个或多个 ③有且只有 1 个 ④有 1 个或 1 个以上  
B: ①互不相交 ②允许相交 ③允许叶结点相交 ④允许树枝结点相交  
C: ①权 ②维数 ③次数 ④序

答案: A=\_\_\_\_\_ B=\_\_\_\_\_ C=\_\_\_\_\_

6. 二叉树 A。在完全的二叉树中, 若一个结点没有 B, 则它必定是叶结点。每棵树都能惟一地转换成与它对应的二叉树。由树转换成的二叉树里, 一个结点 N 的左子女是 N 在原树里对应结点的 C, 而 N 的右子女是它在原树里对应结点的 D。

供选择的答案

- A: ①是特殊的树 ②不是树的特殊形式 ③是两棵树的总称 ④有是只有二个根结点的树形结构  
B: ①左子结点 ②右子结点 ③左子结点或者没有右子结点 ④兄弟  
C~D: ①最左子结点 ②最右子结点 ③最邻近的右兄弟 ④最邻近的左兄弟  
⑤最左的兄弟 ⑥最右的兄弟

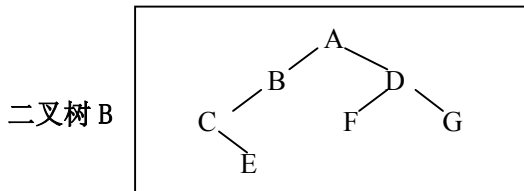
答案: A=\_\_\_\_\_ B=\_\_\_\_\_ C=\_\_\_\_\_ D=\_\_\_\_\_

### 四、简答题 ( )

1. 一棵度为 2 的树与一棵二叉树有何区别?

2. 设如下图所示的二叉树 B 的存储结构为二叉链表, root 为根指针, 结点结构为: (lchild, data, rchild)。其中 lchild, rchild 分别为指向左右孩子的指针, data 为字符型, root 为根指针, 试回答下列问题:

1. 对下列二叉树 B, 执行下列算法 traversal(root), 试指出其输出结果;
2. 假定二叉树 B 共有 n 个结点, 试分析算法 traversal(root) 的时间复杂度。(每问 4 分, 两问共 8 分)



C 的结点类型定义如下:

```

struct node
{char data;
struct node *lchild, rchild;
};
  
```

C 算法如下:

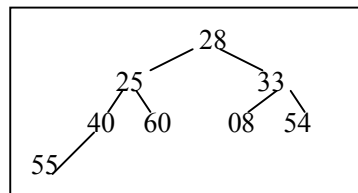
```

void traversal(struct node *root)
{if (root)
{printf("%c", root->data);
traversal(root->lchild);
printf("%c", root->data);
traversal(root->rchild);
}
}
  
```

3. 给定二叉树的两种遍历序列, 分别是:

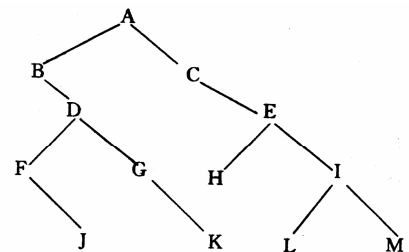
前序遍历序列: D, A, C, E, B, H, F, G, I; 中序遍历序列: D, C, B, E, H, A, G, I, F,  
试画出二叉树 B, 并简述由任意二叉树 B 的前序遍历序列和中序遍历序列求二叉树 B 的思想方法。

4. 给定如图所示二叉树 T, 请画出与其对应的中序线索二叉树。

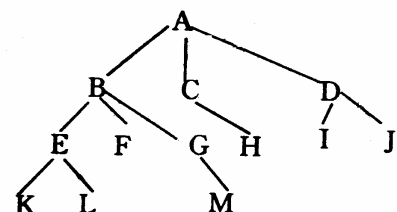


## 五、阅读分析题 ( )

1. 试写出如图所示的二叉树分别按先序、中序、后序遍历时得到的结点序列。



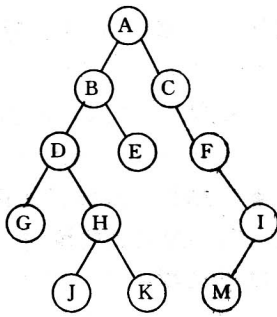
2. 把如图所示的树转化成二叉树。



3. 阅读下列算法，若有错，改正之。

```
BiTree InSucc(BiTree q){  
//已知 q 是指向中序线索二叉树上某个结点的指针，  
//本函数返回指向*q 的后继的指针。  
r=q->rchild;  
if(!r->rtag)  
    while(!r->rtag)r=r->rchild;  
    return r;  
} //ISucc
```

4. 画出和下列二叉树相应的森林。



## 六、算法设计题 ( )

1. 编写递归算法，计算二叉树中叶子结点的数目。
2. 写出求二叉树深度的算法，先定义二叉树的抽象数据类型。
3. 编写递归算法，求二叉树中以元素值为  $x$  的结点为根的子树的深度。
4. 编写按层次顺序（同一层自左至右）遍历二叉树的算法。
5. 编写算法判别给定二叉树是否为完全二叉树。
6. 假设用于通信的电文仅由 8 个字母组成，字母在电文中出现的频率分别为 0.07, 0.19, 0.02, 0.06, 0.32, 0.03, 0.21, 0.10。试为这 8 个字母设计哈夫曼编码。使用 0~7 的二进制表示形式是另一种编码方案。对于上述实例，比较两种方案的优缺点。