第 5 章树和二叉树练习题答案

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贮结构，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。 （应当是二叉排序树的特点）

（ × ）6.满二叉树中所有结点个数是 $2^{k-1}-1$，其中 k 是树的深度。（应 2^k-1）

（ × ）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。（应 2^{i-1}）

（ √ ）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。

（正确。用二叉链表存储包含 n 个结点的二叉树，结点共有 2n 个链域。由于二叉树中，除根结点外，每一个结点有且仅有一个双亲，所以只有 n-1 个结点的链域存放指向非空子女结点的指针，还有 n+1 个空指针。）即有后继链接的指针仅 n-1 个。

（ √ ）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 5 种形态。

2. 一棵深度为 6 的满二叉树有 $n_1+n_2=0+n_2=n_0-1=31$ 个分支结点和 $2^{6-1}=32$ 个叶子。

注：满二叉树没有度为 1 的结点，所以分支结点数就是二度结点数。

3. 一棵具有 2 5 7 个结点的完全二叉树，它的深度为 9 。

（ 注：用$\lfloor \log_2(n) \rfloor+1=\lfloor 8.xx \rfloor+1=9$

4. 设一棵完全二叉树有 700 个结点，则共有 350 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 500 个叶子结点，有 499 个度为 2 的结点，有 1 个结点只有非空左子树，有 0 个结点只有非空右子树。

答：最快方法：用叶子数＝[n/2]＝500 ，$n_2=n_0-1=499$。 另外，最后一结点为 2i 属于左叶子，右叶子是空的，所以有 1 个非空左子树。完全二叉树的特点决定不可能有左空右不空的情况，所以非空右子树数＝0.

6. 一棵含有 n 个结点的 k 叉树，可能达到的最大深度为 n ，最小深度为 2 。

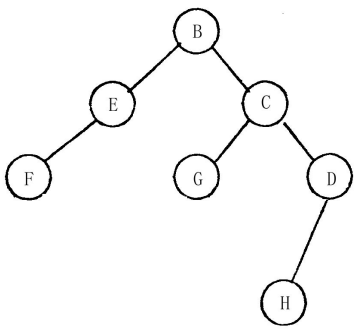
答：当 k=1(单叉树)时应该最深，深度＝n（层）；当 k=n-1（n-1 叉树）时应该最浅，深度＝2（层），但不包括 n=0 或 1 时的特例情况。

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 L R N 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 F E G H D C B 。

解：法 1：先由已知条件画图，再后序遍历得到结果；

法 2：不画图也能快速得出后序序列，只要找到根的位置特征。由前序先确定 root，由中序先确定左子树。例如，前序遍历 BEFCGDH 中，根结点在最前面，是 B；则后序遍历中 B 一定在最后面。

法 3：递归计算。如 B 在前序序列中第一，中序中在中间（可知左右子树上有哪些元素），则在后序中必为最后。如法对 B 的左右子树同样处理，则问题得解。

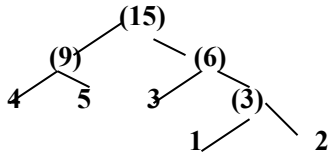


8. 中序遍历的递归算法平均空间复杂度为___O(n)___。

答：即递归最大嵌套层数，即栈的占用单元数。

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___33___。

解：先构造哈夫曼树，得到各叶子的路径长度之后便可求出 WPL＝（4＋5＋3）×2＋（1＋2）×3=33



（注：两个合并值先后不同会导致编码不同，即哈夫曼编码不唯一）

（注：合并值应排在叶子值之后）

三、单项选择题

（ C ）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

（ C ）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

（ C ）3. 具有 n(n>0)个结点的完全二叉树的深度为_____。

(A)⌈$\log_2(n)$⌉　(B)⌊$\log_2(n)$⌋　(C)⌊$\log_2(n)$⌋+1　(D)⌈$\log_2(n)$+1⌉

（ A ）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

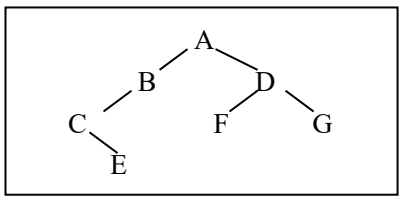
四、简答题

1.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。（共 8 分）

二叉树 B



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);
   }
}
```

解：这是“先根再左再根再右”，比前序遍历多打印各结点一次，输出结果为：A B C C E E B A D F F D G G

特点：①每个结点肯定都会被打印两次；②但出现的顺序不同，其规律是：凡是有左子树的结点，必间隔左子树的全部结点后再重复出现；如 A，B，D 等结点。反之马上就会重复出现。如 C，E，F，G 等结点。

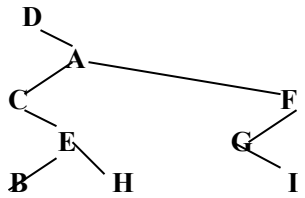
时间复杂度以访问结点的次数为主，精确值为 2*n，时间渐近度为 O(n).

2.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。

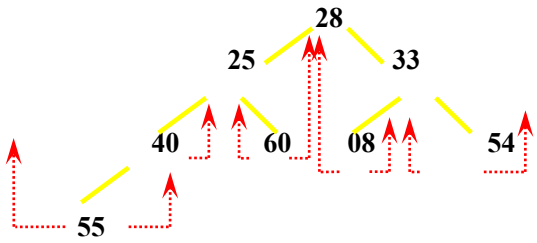
解：方法是：由前序先确定 root，由中序可确定 root 的左、右子树。然后由其左子树的元素集合和右子树的集合对应前序遍历序列中的元素集合，可继续确定 root 的左右孩子。将他们分别作为新的 root，不断递归，则所有元素都将被唯一确定，问题得解。

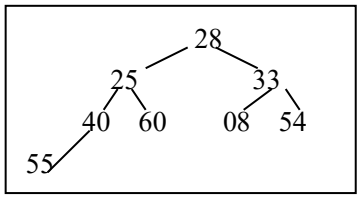


3.给定如图所示二叉树 T，请画出与其对应的中序线索二叉树。

解：要遵循中序遍历的轨迹来画出每个前驱和后继。

中序遍历序列：55　40　25　60　28　08　33　54



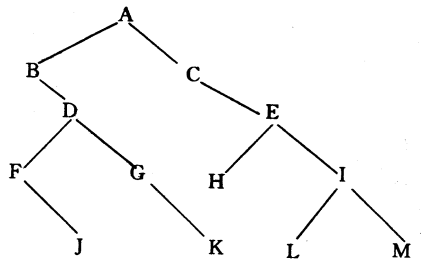


4.试写出如图所示的二叉树分别按先序、中序、后序遍历时得到的结点序列。

答：DLR：A B D F J G K C E H I L M

LDR：B F J D G K A C H E L I M

LRD：J F K G D B H L M I E C A

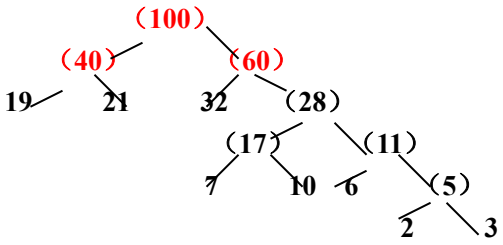


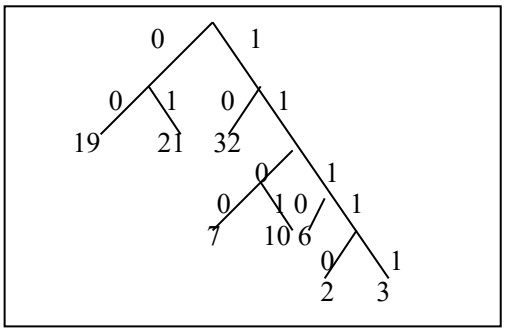
5.假设用于通信的电文仅由 8 个字母组成，字母在电文中出现的频率分别为 0.07，0.19，0.02，0.06，0.32，0.03，0.21，0.10。试为这 8 个字母设计哈夫曼编码。使用 0～7 的等长二进制表示形式是另一种编码方案。对于上述实例，比较两种方案的优缺点。

解：方案 1；哈夫曼编码

先将概率放大 100 倍，以方便构造哈夫曼树。

w={7,19,2,6,32,3,21,10}，按哈夫曼规则：【[（2,3），6], (7,10)】，……19, 21, 32





方案比较：

字母编号	对应编码	出现频率
1	1100	0.07
2	00	0.19
3	11110	0.02
4	1110	0.06
5	10	0.32
6	11111	0.03
7	01	0.21
8	1101	0.10

字母编号	对应编码	出现频率
1	000	0.07
2	001	0.19
3	010	0.02
4	011	0.06
5	100	0.32
6	101	0.03
7	110	0.21
8	111	0.10

方案 1 的 WPL＝2(0.19+0.32+0.21)+4(0.07+0.06+0.10)+5(0.02+0.03)=1.44+0.92+0.25=2.61
方案 2 的 WPL＝3(0.19+0.32+0.21+0.07+0.06+0.10+0.02+0.03)=3
结论：哈夫曼编码优于等长二进制编码

五、算法设计题

编写递归算法，计算二叉树中叶子结点的数目。

解：思路：输出叶子结点比较简单，用任何一种遍历递归算法，凡是左右指针均空者，则为叶子，将其打印出来。

法一：核心部分为：

```
DLR(BiTree root)        /*中序遍历     递归函数*/
{if(root!=NULL)
  {if((root->lchild==NULL)&&(root->rchild==NULL)){sum++; printf("%d\n",root->data);}
    DLR(root->lchild);
    DLR(root->rchild); }
  return(0);
}
```

法二：

```
int LeafCount_BiTree(Bitree T)//求二叉树中叶子结点的数目
{
   if(!T) return 0; //空树没有叶子
   else if(!T->lchild&&!T->rchild) return 1; //叶子结点
   else return Leaf_Count(T->lchild)+Leaf_Count(T->rchild);//左子树的叶子数加
上右子树的叶子数
}//LeafCount_BiTree
```