

- 树最适合用来表示()。
A、有序数据元素 B、无序数据元素
C、元素之间具有分支层次关系的数据 D、元素之间无联系的数据
- 二叉树的第 k 层的结点数最多为()。
A、 2^k-1 B、 $2K+1$ C、 $2K-1$ D、 2^{k-1}
- 设哈夫曼树中的叶子结点总数为 m ，若用二叉链表作为存储结构，则该哈夫曼树中总共有()个空指针域。
A、 $2m-1$ B、 $2m$ C、 $2m+1$ D、 $4m$
- 设某棵二叉树的中序遍历序列为 ABCD，前序遍历序列为 CABD，则后序遍历该二叉树得到序列为()。
A、BADC B、BCDA C、CDAB D、CBDA
- 设某棵二叉树中有 2000 个结点，则该二叉树的最小高度为()。
A、9 B、10 C、11 D、12
- 将含 100 个结点的完全二叉树从根这一层开始，每层上从左到右依次对结点编号，根结点的编号为 1，编号为 89 的结点 X 的双亲的编号为()。
A、44 B、45 C、43 D、无法确定
- 若 X 是二叉中序线索树中一个有右孩子的结点，且 X 不是根结点，则 X 的直接后继为()。
A、X 的双亲 B、X 的右子树中最左的结点
C、X 的左子树中最右结点 D、X 的左子树中最右叶子结点
- 把一棵树转换为二叉树后，这棵二叉树的形态是()。
A、有多种，但根结点没有右孩子 B、唯一的，且根结点没有右孩子
C、唯一的，且根结点可能有右孩子 D、有多种，但根结点都没有右孩子
解释：转换规则是唯一的，所以转换成的二叉树是唯一的
- 一棵二叉树 Tree，度为 2 的结点数为 100 个，则叶子结点数为()。
A、99 个 B、100 个 C、101 个 D、102 个
- 在一棵度为 3 的树中，度为 3 的结点数为 2 个，度为 2 的结点数为 1 个，度为 1 的结点数为 2 个，则度为 0 的结点数为()个。
A、4 B、5 C、6 D、8
- 算法阅读题：

```
void ABC(BTNode * BT)
{
    if (BT) {
        ABC (BT->left);
        ABC (BT->right);
        printf(BT->data);
    }
}
```

该算法的功能是：
- 应用题：某子系统在通信联络中 2 可能出现 8 种字符，其出现的概率分别为 0.05, 0.29, 0.07, 0.08, 0.14, 0.23, 0.03, 0.11，试设计哈夫曼编码。
- 已知用一维数组存放的一棵完全二叉树：ABCDEFGHIJKL，写出该二叉树的先序、中序和后序遍历序列。
- 设某棵二叉树的中序遍历序列为 DBEAC，前序遍历序列为 ABDEC，要求画出该二叉树；求出二叉树的后序遍历序列；
- 将 14 题的二叉树转换成树