

一、基础题

- 已知一棵树边的集合为 $\{<I,M>, <I,N>, <E,I>, <B,E>, <B,D>, <A,B>, <G,J>, <G,K>, <C,G>, <C,F>, <H,L>, <C,H>, <A,C>\}$, 请画出这棵树, 并回答下列问题:
 - (1) 哪个是根结点?
 - (2) 哪些是叶子结点?
 - (3) 哪个是结点 G 的双亲?
 - (4) 哪些是结点 G 的祖先?
 - (5) 哪些是结点 G 的孩子?
 - (6) 哪些是结点 E 的子孙?
 - (7) 哪些是结点 E 的兄弟? 哪些是结点 F 的兄弟?
 - (8) 结点 B 和 N 的层次号分别是什么?
 - (9) 树的深度是多少?
 - (10) 以结点 C 为根的子树的深度是多少?
- 一棵深度为 H 的满 k 叉树有如下性质: 第 H 层上的结点都是叶子结点, 其余各层上每个结点都有 k 棵非空子树。如果按层次顺序从 1 开始对全部结点编号, 问:
 - (1) 各层的结点数目是多少?
 - (2) 编号为 p 的结点的父亲结点 (若存在) 的编号是多少?
 - (3) 编号为 p 的结点的第 i 个儿子结点 (若存在) 的编号是多少?
 - (4) 编号 p 的结点有兄弟的条件是什么? 其右兄弟的编号是多少?
- 已知一棵度为 k 的树中有 n_1 个度为 1 的结点, n_2 个度为 2 的结点, ..., n_k 个度为 k 的结点, 问该树中有多少个叶子结点?
- 一棵含有 n 个结点的 k 叉树, 可能达到的最大和最小深度各为多少?
- 画出和下列已知序列对应的树 T:
树的先根次序访问序列为: GFKDAIEBCHJ;
树的后根次序访问序列为: DIAEKFCJHBG。

二、算法题

- 假定用两个一维数组 $L[n+1]$ 和 $R[n+1]$ 作为有 n 个结点的二叉树的存储结构, $L[i]$ 和 $R[i]$ 分别指示结点 $i(i=1,2,\dots,n)$ 的左孩子和右孩子, 0 表示空。试写一个算法判别结点 u 是否结点 v 的子孙。
- 假设在二叉链表的节点中增设两个域: 双亲域 (parent) 以指示其双亲结点; 标志域 (mark 取值 0..2) 以区分在遍历过程中到达该结点时应继续向左或向右或访问该结点。试以此存储结构编写不用栈进行后续遍历的递推形式的算法。
- 编写递归算法, 将二叉树中所有结点的左右子树相互交换。
- 编写按层次顺序 (同一层自左向右) 遍历二叉树的算法。