

- 4.4 证明在  $N$  个节点的二叉树中, 存在  $N+1$  个 null 链, 代表  $N+1$  个儿子。
- 4.5 证明在高度为  $h$  的二叉树中, 节点的最大个数是  $2^{h+1}-1$ 。
- 4.6 满节点(full node)是具有两个儿子的节点。证明满节点的个数加 1 等于非空二叉树的树叶的个数。
- 4.7 设二叉树有树叶  $l_1, l_2, \dots, l_M$ , 各树叶的深度分别是  $d_1, d_2, \dots, d_M$ 。证明,  $\sum_{i=1}^M 2^{-d_i} \leq 1$  并确定何时等号成立。
- 4.8 给出对应图 4-70 中的树的前缀表达式、中缀表达式以及后缀表达式。

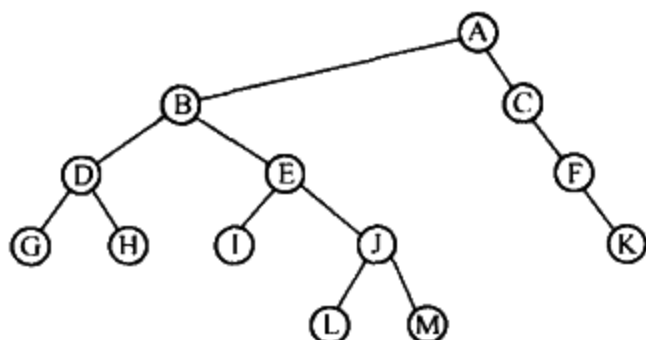


图 4-69 练习 4.1~4.3 所用的图



图 4-70 练习 4.8 中的树

- 4.9 a. 指出将 3, 1, 4, 6, 9, 2, 5, 7 插入到初始为空二叉查找树中的结果。  
b. 指出删除根后的结果。
- 4.10 编写一个程序, 该程序列出一个目录中所有的文件和它们的大小。模拟联机代码中的程序。
- 4.11 编写 TreeSet 类的实现程序, 其中相关的迭代器使用二叉查找树。在每个节点上添加一个指向其父节点的链。
- 4.12 通过存储类型 `TreeSet<Map.Entry<KeyType, ValueType>>` 的一个数据成员编写实现 TreeMap 类的程序。
- 4.13 编写 TreeSet 类的实现程序, 其中相关的迭代器使用二叉查找树。在每个节点上添加通向下一个最小节点和下一个最大节点的链。为使所编程序更简单, 添加头节点和尾节点, 它们不属于二叉树的一部分, 但有助于使得程序的链表部分更简单。
- 4.14 设欲做一个实验来验证由随机 insert/remove 操作对可能引起的问题。这里有一个策略, 它不是完全随机的, 但却是足够封闭的。通过插入从 1 到  $M = \alpha N$  之间随机选出的  $N$  个元素来建立一棵具有  $N$  个元素的树。然后执行  $N^2$  对先插入后删除的操作。假设存在例程 `randomInteger(a, b)`, 它返回一个在  $a$  和  $b$  之间(包括  $a, b$ )的均匀随机整数。
- a. 解释如何生成在 1 和  $M$  之间的一个随机整数, 该整数不在这棵树上(从而可以进行随机插入)。用  $N$  和  $\alpha$  来表示这个操作的运行时间。
- b. 解释如何生成在 1 和  $M$  之间的一个随机整数, 该整数已经存在于这棵树上(从而可以进行随机删除)。这个操作的运行时间是多少?
- c.  $\alpha$  的好的选择是什么? 为什么?
- 4.15 编写一个程序, 凭经验计算下列删除具有两个儿子的节点的各方法的值:
- a. 用  $T_L$  中最大节点  $X$  来代替, 递归地删除  $X$ 。
- b. 交替地用  $T_L$  中最大的节点以及  $T_R$  中最小的节点来代替, 并递归地删除适当的节点。
- c. 随机地选用  $T_L$  中最大的节点或  $T_R$  中最小的节点来代替(递归地删除适当的节点)。
- 哪种方法给出最好的平衡? 哪种在处理整个操作序列过程中花费最少的 CPU 时间?