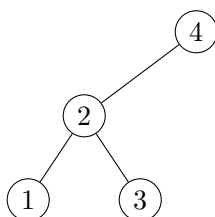


算法基础 作业 9

12.2-4. Bunyan 教授认为他发现了一个二叉搜索树的重要性质. 假设在一棵二叉搜索树中查找一个关键字 k , 查找结束于一个树叶. 考虑三个集合: A 为查找路径左边的关键字集合; B 为查找路径上的关键字集合; C 为查找路径右边的关键字集合. Bunyan 教授声称: 任何 $a \in A$, $b \in B$ 和 $c \in C$, 一定满足 $a \leq b \leq c$. 请给出该教授这个论断的一个最小可能的反例.

解: 反例如下, 查找 1, 则 $B = \{1, 2, 4\}$, $C = \{3\}$, 取 $b = 4$, $c = 3$, 有 $b > c$ 违反了教授的论断.

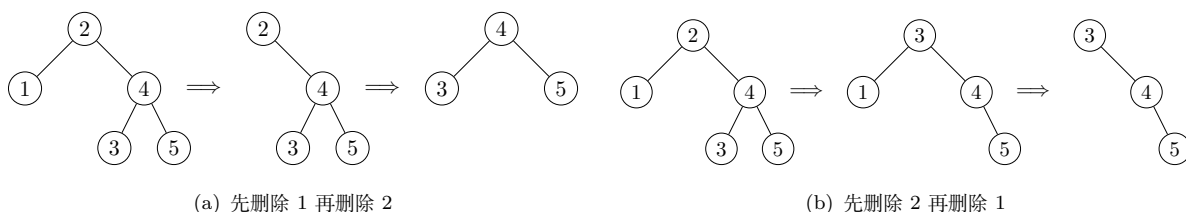


12.2-9. 设 T 是一棵二叉搜索树, 其关键字互不相同; 设 x 是一个叶结点, y 为其父结点. 证明: $y.key$ 或者是 T 中大于 $x.key$ 的最小关键字, 或者是 T 中小于 $x.key$ 的最大关键字.

解: 若 $x = y.left$, 调用 TREE-SUCCESSOR(x), 因为 x 是叶结点, 所以前两行的 **if** 语句不会执行, 而 x 是 $y = x.p$ 的左子结点, 所以不会执行 **while** 循环, 直接返回 y , 说明 $y.key$ 是 T 中大于 $x.key$ 的最小关键字. 同理, 若 $x = y.right$, 调用 TREE-PREDECESSOR(x) 将返回 $y = x.p$, 说明 $y.key$ 是 T 中小于 $x.key$ 的最大关键字. 综上所述, 题中命题成立.

12.3-4. 删除操作可交换吗? 可交换的含义是, 先删除 x 再删除 y 留下的结果树与先删除 y 再删除 x 留下的结果树完全一样. 如果是, 说明为什么? 否则, 给出一个反例.

解: 不可交换. 如下所示, 先删除 1 再删除 2 与先删除 2 再删除 1 留下的结果树不同.



12.Extra-1. 假设某棵二叉搜索树的所有键均为 1 到 10 之间的整数, 我们要查找 5, 那么一下哪个不可能是键的检查序列? 说明原因.

- A. 10, 9, 8, 7, 6, 5
- B. 3, 10, 8, 7, 4, 5
- C. 1, 10, 2, 9, 3, 8, 4, 7, 6, 5
- D. 2, 7, 3, 8, 4, 5
- E. 1, 2, 10, 8, 4, 5

解: 因为在二叉搜索树中查找等价于在一个有序数组中二分查找, 设检查到的键为 c , 之前检查过的键中, 小于 5 的最大值为 m , 大于 5 的最小值为 n (初始值分别为 1 和 10), 那么 c 一定满足 $m \leq c \leq n$. 五个序列中 C. 和 D. 违反了这一条件, 因此不可能是检查序列.

13.1-4. 假设将一棵红黑树的每一个红结点“吸收”到它的黑色父结点中, 使得红结点的子结点变成黑色父结点的子结点 (忽略关键字的变化). 当一个黑结点的所有红色子结点都被吸收后, 它可能的度为多少? 所得的树的叶结点深度如何?

解: 该黑结点可能的度为 0, 2, 3, 4. 当该黑结点为叶结点 (NIL) 时, 吸收后其仍为叶结点, 度为 0; 若该黑结点分别有 0/1/2 个子红结点时, 吸收后黑结点的度为 2/3/4.

吸收后所得树的叶结点的深度即原红黑树的黑高.

13.1-5. 证明: 在一棵红黑树中, 从某结点 x 到其后代叶结点的所有简单路径中, 最长的一条至多是最短一条的 2 倍.

解: 不妨假设最长路径和最短路径上结点数分别为 m 和 n , 二者均有 b 个黑结点. 因为红结点的子结点必为黑结点, 所以路径上红结点数最多为 b 个, 因此有 $2b \geq m \geq n \geq b$, 显然 $2n \geq m$.

13.2-4. 证明: 任何一棵含 n 个结点的二叉搜索树可以通过 $O(n)$ 次旋转, 转变为其他任何一棵含 n 个结点的二叉搜索树. (提示: 先证明至多 $n-1$ 次右旋足以将树转变为一条右侧伸展的链)

解: 采用如下方法将二叉搜索树转变为右侧伸展的链: 考虑从根结点一直向右伸展的路径, 从初始情况开始, 每次迭代都对链上有左子结点的所有结点右旋, 直到链上没有有左子结点的结点为止. 因为每次旋转都会让链上增加一个结点, 而初始情况下链上至少有一个根结点, 因此上述操作至多进行 $n-1$ 次右旋.

因为左旋和右旋互为逆操作, 因此可以通过至多 $n-1$ 次左旋即可将一条右侧伸展的链转变为一棵二叉搜索树, 只需要将上述方法的执行顺序反过来并用左旋代替右旋即可. 因此可以通过至多 $2n-2 = O(n)$ 次旋转, 先将二叉搜索树转变为右侧伸展的链, 再将其转变为目标二叉搜索树.

13.Extra-1. 如何构造一棵最差的红黑树, 其中从根结点到很多叶结点 (空链接) 的路径长度均为 $2 \lg N$.

解: (真的能构造吗? 考虑 $N = 2^k$, 则需要有一条路径长度为 $2k$, 只有这一条路径长度为 $2k$, 其他路径长度也至少为 k , 显然不可能)