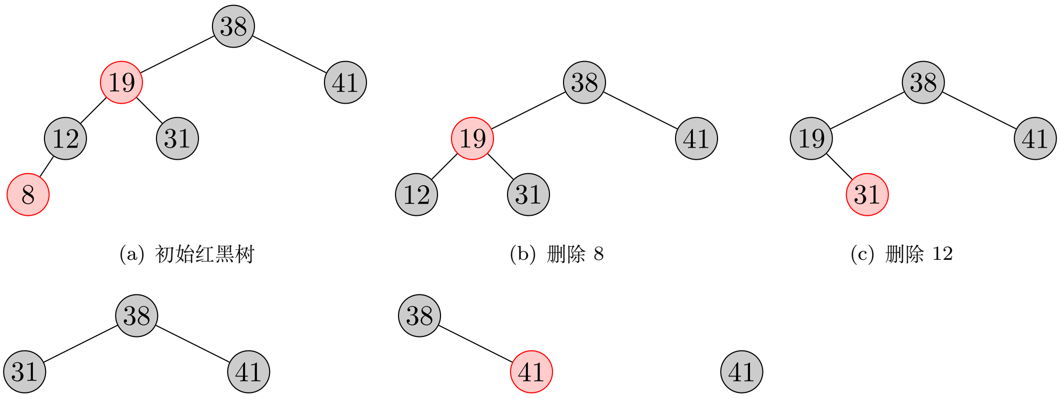
傅申 PB20000051
2022 秋

算法基础 作业 10

13.4–3. 在练习 13.3–2 中, 将关键字 41, 38, 31, 12, 19, 8 连续插入一棵初始的空树中, 从而得到一棵红黑树. 请给出从该树中连续删除关键字 8, 12, 19, 31, 38, 41 后的红黑树.

解: 忽略 Nil 叶结点, 删除关键字后的红黑树依次如下

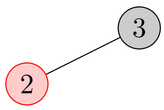


(a) 初始红黑树 (b) 删除 8 (c) 删除 12
(d) 删除 19 (e) 删除 31 (f) 删除 38 (g) 删除 41

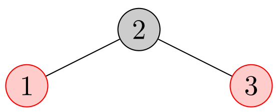
图 3.1: 依次删除关键字后的红黑树

13.4–7. 假设用 RB-Insert 将一个结点 x 插入一棵红黑树, 紧接着又用 RB-Delete 将它从树中删除. 结果的红黑树与初始的红黑树是否一样? 证明你的答案.

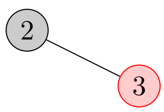
解: 不一样, 考虑如下的红黑树.



插入关键字为 1 的结点后为



而再删除 1 后得到



14.1–2. 对于图 14–1 中的红黑树 T 和关键字 $x.key$ 为 35 的结点 x, 说明执行 OS-RANK (T, x) 的过程.

解: 首先 r 被初始化为 0, y 初始化为 x. 然后进入循环, 第一轮 $y = y.p.left$, r 不改变, y 更新为关键字 38 的结点; 第二轮 $y = y.p.right$, r 增加 2, y 更新为关键字 30 的结点; 第三轮 $y = y.p.left$, r 不改变, y 更新为关键字 41 的结点; 第四轮 $y = y.p.right$, r 增加 13, y 更新为根结点, 循环结束. 最后 $r = 15$.

14.1–4. 写出一个递归过程 OS-KEY-RANK (T, k), 以一棵顺序统计树 T 和一个关键字 k 作为输入, 要求返回 k 在由 T 表示的动态集合中的秩. 假设 T 的所有关键字都不相同.

解: 如下

Algorithm 1: OS-KEY-RANK (T, k)

$x = T.root$
if $k = x.key$
    return $x.left.size + 1$
else if $k > x.key$
    return $x.left.size + 1 +$ OS-KEY-RANK$(x.right, k)$
else
    return OS-KEY-RANK$(x.left, k)$

14.2–2. 能否在不影响红黑树任何操作的渐进性能的前提下, 将结点的黑高作为树中结点的一个属性来维护? 说明如何做, 如果不能, 请说明理由. 如何维护结点的深度?

解: 因为结点的黑高只与其子结点的黑高和子结点的颜色有关, 所以可以完成题中操作. 在插入时, 插入的结点的子结点都是哨兵, 因此其黑高为 1, 并由此向上修改其他结点的黑高, 最后在 FIXUP 过程中对部分结点进行调整. 同理, 在删除时先将结点删除, 局部修改黑高并向上传播, 然后在 FIXUP 过程中对部分结点进行调整.

而由于结点的深度取决于其父结点的深度, 因此无法维护结点深度.

14.3–4. 给定一棵区间树 T 和一个区间 i, 请描述如何在 $O(\min(n, k \lg n))$ 时间内列出 T 中所有与 i 重叠的区间, 其中 k 是输出的区间数. (提示: 一种简单的方法是做若干次查询, 并且在这些查询操作中修改树, 另一种略微复杂点的方法是不对树进行修改)

解: 如下, 其中每个结点最多遍历一次, 所以时间是 $O(n)$ 的; 而每次输出一个区间对应的递归深度最多为 $\lg n$, 因此时间是 $O(k \lg n)$ 的.

Algorithm 2: INTERVALS-SEARCH (T, i)

$x = T.root$
if i overlaps $x.int$
    Print $x.int$
if $x.left \neq T.nil$ **and** $x.left.max \geqslant i.low$
    INTERVALS-SEARCH($x.left, i$)
if $x.right \neq T.nil$ **and** $x.int.low \leqslant i.high$ **and** $x.right.max \geqslant i.low$
    INTERVALS-SEARCH($x.right, i$)