

## 算法基础

第四次作业 (DDL: 2024 年 10 月 19 日 23:59)

解答过程中请写出必要的计算和证明过程

**Q1.** (10 + 10 = 20 分) 红黑树:

(a) 将关键字 41, 38, 31, 12, 19, 8 连续地插入一棵初始为空的红黑树之后, 请画出该结果树。

(b) 对于 (a) 中得到的红黑树, 依次删除 8, 12, 19, 31, 38, 41, 请画出每次删除操作后的红黑树。

**Solution:**

**Q2.** (20 分) 在一棵黑高为  $k$  的红黑树中, 内部结点 (带关键字的结点) 最多可能有多少个? 最少可能有多少个? 说明理由。

**Solution:**

- 最多: 从根到叶子的路径中, 任意两个黑结点之间最多只能有一个红结点。因此, 这棵树的最大高度是  $2k$  (路径上的结点交替为红色和黑色)。为了最大化内部结点的数量, 可以将树构造成一棵满二叉树, 这样总共会有  $2^{2k} - 1$  个内部结点。
- 最少: 构造一棵完全的二叉树, 所有结点均为黑色。这样的树有  $2^k - 1$  个内部节点。

**Q3.** (15 + 15 = 30 分) 区间树:

假设我们希望记录一个区间集合的**最大重叠点**, 即被最多数目区间所覆盖的那个点。

(a) 证明: 在最大重叠点中, 一定存在一个点是其中一个区间的端点。

(b) 设计一个数据结构, 使得它能够有效地支持 INTERVAL-INSERT、INTERVAL-DELETE, 以及返回最大重叠点的 FIND-POM 操作。

**Solution:**

- (a) 假设我们有一个最大重叠点  $p$ 。只要我们想象移动点  $p$ , 但不越过任何区间的任何端点, 那么我们就不会改变包含  $p$  的区间的数量。所以, 我们将它向右移动, 直到我们到达某个区间的端点, 因此我们有一个最大重叠点, 这就是一个区间的端点。

- (b) 数据结构: 考虑一个列表, 列表中元素  $x$  是区间的端点, 每个  $x$  有值  $x.pos$  和一个表示  $x$  是区间左端点还是右端点的值 ( $x.dir = L$  或  $x.dir = R$ )。

FIND-POM: 对该列表关于  $x.pos$  排序 (若一个值既是区间的左端点, 又是区间的右端点时, 把左端点排在前面)。然后遍历该列表, 维护一个计数器  $c$  表示当前位置的区间嵌套数, 当经过一个左端点时,  $c \leftarrow c+1$ , 当经过一个右端点时,  $c \leftarrow c-1$ , 在遍历过程中记录  $c$  的最大值, 时间复杂度为  $O(n \log n)$ 。这种方法返回的最大重叠点可能不全。

若要返回所有最大重叠点, 需要遍历最小值到最大值之间的所有数, 记录它们出现在多少个区间里。

**Q4.** (15 + 15 = 30 分) 斐波那契堆删除操作的另一种实现:

Pisano 教授提出了下面的 FIB-HEAP-DELETE 过程的一个变种, 声称如果删除的结点不是由  $H.min$  指向的结点, 那么该程序运行地更快。

---

**Algorithm 1:** PISANO-DELETE( $H, x$ )

---

**Input** : A Fibonacci heap  $H$ , a node  $x$

**Output:** Heap  $H$  after deleting node  $x$

```

1 if  $x == H.min$  then
2   | FIB-HEAP-EXTRACT-MIN( $H$ );
3 end
4 else
5   |  $y \leftarrow x.p$ ;
6   | if  $y \neq NIL$  then
7     |   Cut( $H, x, y$ );
8     |   Cascading-Cut( $H, y$ );
9   | end
10  | Add  $x$ 's child list to the root list of  $H$ ;
11  | Remove  $x$  from the root list of  $H$ ;
12 end

```

---

- (a) 该教授的声称是基于第 10 行可以在  $O(1)$  实际时间完成的这一假设, 它的程序可以运行的更快。该假设有什么问题吗?

- (b) 当  $x$  不是由  $H.min$  指向时, 给出 PISANO-DELETE 实际时间的一个

好（紧凑）上界。你给出的上界应该以  $x.degree$  和调用 CASCADING-CUT 的次数  $c$  这两个参数来表示。

**Solution:**

- (a) 错误。实际运行时间与节点  $x$  的子节点数量成正比，因为对于每个子节点，在将其放入根列表时，其父指针需要从  $x$  更新为 NIL。
- (b) 第 10 行的实际运行时间以  $x.degree$  为界限，因为更新每个  $x$  的子节点只需要常数时间。因此，如果执行的级联切割数量为  $c$ ，实际的开销为  $O(c + degree)$ 。

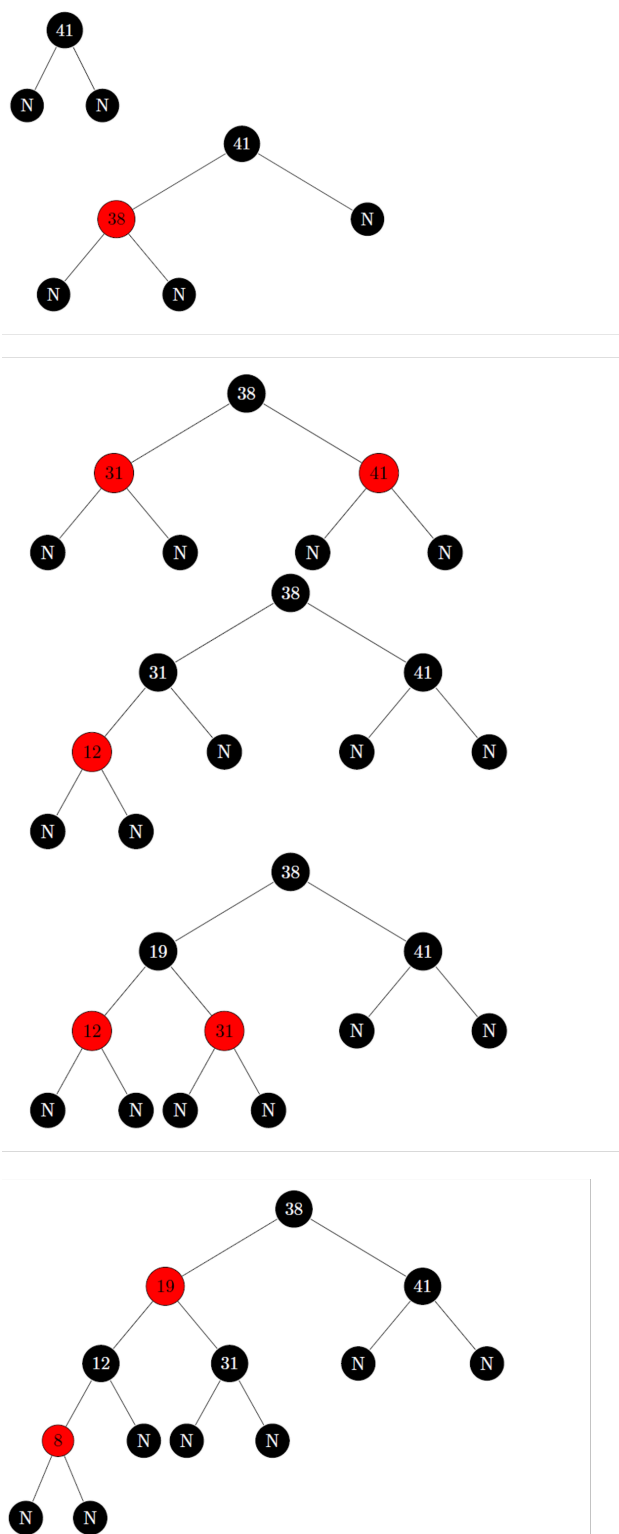


图 1: (a) insert

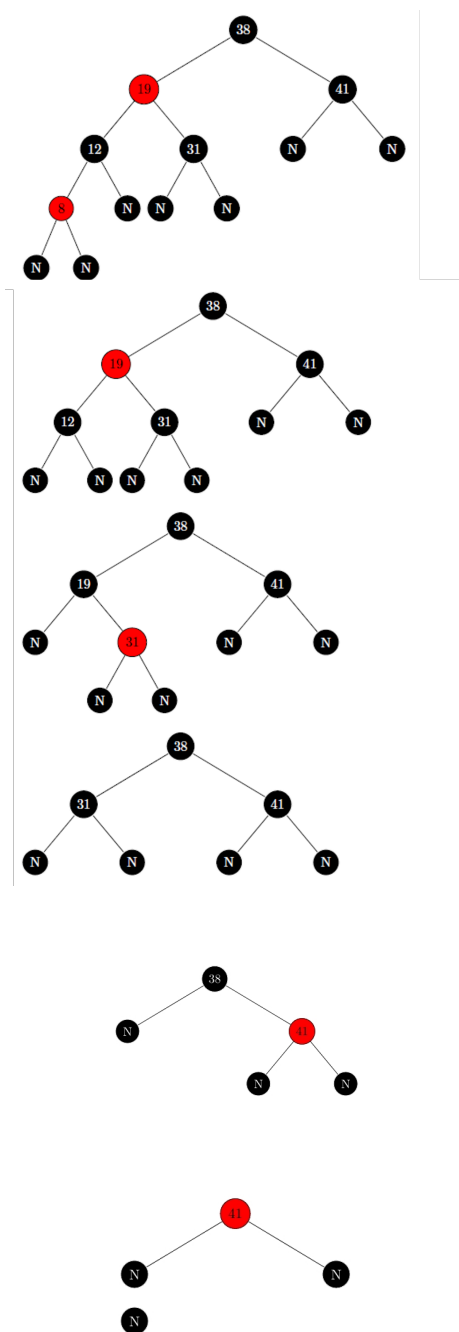


图 2: (b) delete