

# 算法基础第四次作业

艾语晨

2020 年 11 月 5 日

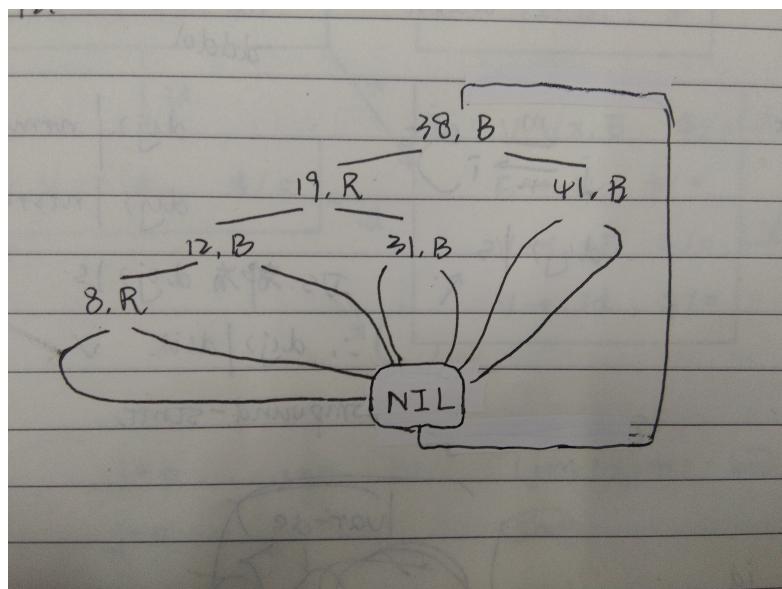
# 目录

<b>1 (Week 7)</b>	<b>2</b>
1.1 红黑树 . . . . .	2
(a) 构建红黑树 . . . . .	2
(b) 删除结点 . . . . .	2
1.2 最大重叠点 . . . . .	3
(a) 一定有一个是端点 . . . . .	3
(b) 红黑树的扩增 . . . . .	3
1.3 Pisano . . . . .	3
(a) 这个假设有问题 . . . . .	3
(b) 时间复杂度上界 . . . . .	3

# 第四次作业 (Week 7)

## 第 1.1 题 红黑树

### (a) 构建红黑树



### (b) 删除结点

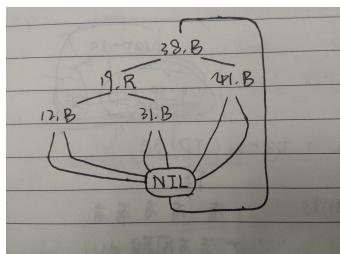


图 1.1: 删除 8

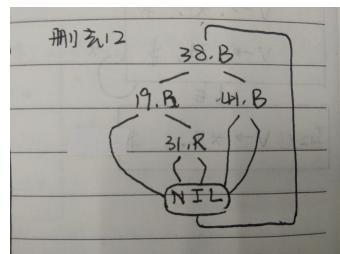


图 1.2: 删除 12

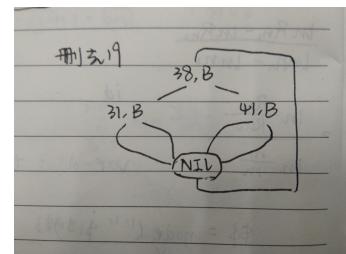


图 1.3: 删除 19

## 第 1.2 题 最大重叠点

### (a) 一定有一个是端点

不妨设点  $p$  是一个最大重叠点，由最大重叠点定义可知，这样的点一定存在。只要没有将  $p$  移动到超过某一个区间到边界（低端点或者高端点），就将  $p$  向正方向移动。故在此情形下，最终一定会到达某一个区间的端点。这时候就会得到一个是某个区间端点的最大重叠点

### (b) 红黑树的扩增

使红黑树记录所有的端点，并额外维护一个域，用于区分左端点和右端点。左端点记为 +1，右端点记为 -1，并且给树中的每个结点扩张一个额外信息，维护比这个节点的值小的所有节点，为左端点的个数减去右端点的个数，即为这个结点的涵盖区间数

#### 操作 1 INTERVAL-INSERT

对两个端点分别调用红黑树的插入算法，并更新各自路径上的结点（最大重叠点信息）

#### 操作 2 INTERVAL-DELETE

按照红黑树的删除算法分别删除两个左右端点，并更新各自路径上的结点

#### 操作 3 FIND-POM

遍历树上的所有节点，涵盖区间数最大的点即为所求最大重叠点

## 第 1.3 题 Pisano

### (a) 这个假设有问题

这一步的时间复杂度实际上是  $O(D[x])$  的而非  $O(1)$  的，这是因为对于  $x$  的每一个子节点，都需要至少做一个将其  $.p$  指针从  $x$  更改置为 NIL

### (b) 时间复杂度上界

第 5 ~ 7 行（主要为 CASCADING-CUT 函数）的时间复杂度为  $O(c)$ ，而第 8 行第调用次数由  $x$  的度数  $x.degree$  决定，故总的时间复杂度为  $O(c + x.degree)$