

Homework 6

PB17111623 范睿

2019 年 11 月 17 日

1

1.1 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。

1.1.1 a. 证明：最大重叠点一定是其中一个区间的端点

证明：

若最大重叠点A不是任何一个区间的端点，那么一定可以找到距离此点最近的任何一个区间的端点B。B的区间重叠数 \geq A的区间重叠数。所以B为最大重叠点。所以最大重叠点一定是其中一个区间的端点。

1.1.2 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。

在区间树的基础上做一些操作：

区间树结点包含：*key*（区间左端点）、*max*（此节点带领的树的最右端）、*i*（区间值）、*left*（左子树指针）、*right*（右子树指针）、*parent*（父节点指针）。

增加一项 $\langle degree, intervals \rangle$ 。*intervals*记录若干个区间，这些区间内所有的点都是此结点带领的树的最大重叠区间，*intervals*记录他们的最大区间重叠数。

每个叶结点的*degree*为1，*intervals*中只有一个区间，与此叶节点的区间值相同。

确定除了叶结点的所有其他结点的*degree*和*intervals*(下一页)：

时间复杂度分析：

INTERVAL-INSERT、INTERVAL-DELETE的时间复杂度与区间树相同，为 $\mathcal{O}(\log n)$

FIND-POM时间复杂度为 $\mathcal{O}(1)$

Algorithm 1 计算degree和intervals

Require: 非叶结点 p

```

1: if  $p \rightarrow \text{left} \rightarrow \text{degree} > p \rightarrow \text{right} \rightarrow \text{degree}$  then
2:   for  $i$  in  $p \rightarrow \text{left} \rightarrow \text{intervals}$  do
3:     if  $\text{Overlap}(i, p \rightarrow i)$  then
4:       //add  $\text{Overlap}(i, p \rightarrow i)$  into  $p \rightarrow \text{intervals}$ 
5:       //  $p \rightarrow \text{degree} \leftarrow p \rightarrow \text{left} \rightarrow \text{degree} + 1$ 
6:     end if
7:   end for
8:   if  $p \rightarrow \text{intervals}$  has nothing then
9:     if  $p \rightarrow \text{left} \rightarrow \text{degree} == p \rightarrow \text{right} \rightarrow \text{degree} + 1$  then
10:      for  $i$  in  $p \rightarrow \text{right} \rightarrow \text{intervals}$  do
11:        if  $\text{Overlap}(i, p \rightarrow i)$  then
12:          //add  $\text{Overlap}(i, p \rightarrow i)$  into  $p \rightarrow \text{intervals}$ 
13:          //  $p \rightarrow \text{degree} \leftarrow p \rightarrow \text{right} \rightarrow \text{degree} + 1$ 
14:        end if
15:      end for
16:       $p \rightarrow \text{intervals} \leftarrow p \rightarrow \text{intervals} \cup p \rightarrow \text{left} \rightarrow \text{intervals}$ 
17:    else
18:       $p \rightarrow \text{intervals} \leftarrow p \rightarrow \text{left} \rightarrow \text{intervals}$ 
19:       $p \rightarrow \text{degree} \leftarrow p \rightarrow \text{left} \rightarrow \text{degree}$ 
20:    end if
21:  end if
22: else if  $p \rightarrow \text{left} \rightarrow \text{degree} < p \rightarrow \text{right} \rightarrow \text{degree}$  then
23:   //Like Above
24: else
25:   for  $i$  in  $p \rightarrow \text{left} \rightarrow \text{intervals} \cup p \rightarrow \text{right} \rightarrow \text{intervals}$  do
26:     if  $\text{Overlap}(i, p \rightarrow i)$  then
27:       //add  $\text{Overlap}(i, p \rightarrow i)$  into  $p \rightarrow \text{intervals}$ 
28:       //  $p \rightarrow \text{degree} \leftarrow p \rightarrow \text{left} \rightarrow \text{degree} + 1$ 
29:     end if
30:   end for
31:   if  $p \rightarrow \text{intervals}$  has nothing then
32:      $p \rightarrow \text{intervals} \leftarrow p \rightarrow \text{left} \rightarrow \text{intervals} \cup p \rightarrow \text{right} \rightarrow \text{intervals}$ 
33:      $p \rightarrow \text{degree} \leftarrow p \rightarrow \text{left} \rightarrow \text{degree} + 1$ 
34:   end if
35: end if

```

1.2

- 1.2.1 a. 该教授的声称是基于第 8 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

把x的所有孩子加入链表需要将x的所有孩子的parent指针改为NIL，实际上需要 $O(x.degree)$ 的时间。

- 1.2.2 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。

$$O(c + x.degree)$$

- 1.3 使用链表表示和加权合并启发式策略，写出 MAKE-SET, FIND-SET 和 UNION 操作的伪代码。

Algorithm 2 MAKE-SET

Require: 数字x

Ensure: 只包含x的集合S

```

1:  $N.key \leftarrow x$ 
2:  $N.next \leftarrow NULL$ 
3:  $N.set \leftarrow S$ 
4:  $S.head \leftarrow N$ 
5:  $S.tail \leftarrow N$ 
6:  $S.num \leftarrow 1$ 
7: return  $S$ 

```

Algorithm 3 FIND-SET

Require: 结点x

Ensure: 包含x结点的唯一集合S

```

1: return  $x.set$ 

```

Algorithm 4 UNION

Require: 结点 x, y

```

1:  $Sx \leftarrow x.set$ 
2:  $Sy \leftarrow y.set$ 
3: if  $Sx.num \leq Sy.num$  then
4:    $Sy.tail.next \leftarrow Sx.head$ 
5:    $Sy.tail \leftarrow Sx.tail$ 
6:   for  $i$  in linklist  $Sx.head$  do
7:      $i.set \leftarrow Sy$ 
8:   end for
9:    $Sy.num += Sx.num$ 
10: else
11:    $Sx.tail.next \leftarrow Sy.head$ 
12:    $Sx.tail \leftarrow Sy.tail$ 
13:   for  $i$  in linklist  $Sy.head$  do
14:      $i.set \leftarrow Sx$ 
15:   end for
16:    $Sx.num += Sy.num$ 
17: end if

```
