

Homework 6

PB17000297 罗晏宸

October 26 2019

1 Problem 14-1 Point of maximum overlap

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：最大重叠点一定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。

解

a 假设所有的最大重叠点均不是任意区间的端点，设一个最大重叠点为 m ，与其相距最近的区间低端点为 $i.low$ ，与其相距最近的区间高端点为 $j.high$ ，则在区间 $[i.low, j.high]$ 上的所有点均被和 m 相同数目的区间所覆盖，故 $i.low$ 与 $j.high$ 也是最大重叠点，这与假设矛盾，因此假设不成立，即最大重叠点中一定有区间的端点。

b 维护一个升序链表，链表的基类型是一个拥有两个成员的结构体，成员 i 表示区间的序号（加入数组的次序），成员 end 表示区间的端点位置，即对于每一个区间 x ，数组中有两个元素分别对应其左端点和右端点。这个链表的有序基于对于各成员端点值的比较，通过二分查找定位，INTERVAL-INSERT、INTERVAL-DELETE 操作均可以在 $O(\lg n)$ 的时间下实现。对于 FIND-POM 操作，从表头开始遍历链表，同时对每个区间，记录其左端点是否已出现且仅出现左端点（即对于每一个序号 i ，记录其出现次数是否为

1), 对于每一个左端点, 覆盖厚度加一, 每出现一个右端点 (区间的端点此前出现过), 覆盖厚度减一, 比较是否更新最大覆盖厚度, 且更新时记录端点, 最终返回最大覆盖厚度对应的端点, 以上操作消耗的时间是 $O(n)$ 。

2 Problem 19-1 Alternative implementation of deletion

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。

```
PISANO-DELETE( $H, x$ )
1  if  $x == H.min$ 
2      FIB-HEAP-DELETE( $H$ )
3  else
4       $y = x.p$ 
5      if  $y \neq \text{NIL}$ 
6          CUT( $H, x, y$ )
7          CASCADING-CUT( $H, y$ )
8      add  $x$ 's child list to the root list of  $H$ 
9      remove  $x$  from the root list of  $H$ 
```

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?

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。

解

a 该假设的问题在于, 将 x 的所有孩子添加到 H 中需要的时间应当依赖于 x 孩子的个数, 并不是常数时间。

b 由上题可知, 第 8 行实际时间应为 $O(x.degree)$, 而 CUT 和 CASCADING-CUT 是常数时间内可以完成的, 因此当 x 不是由 $H.min$ 指向时, PISANO-DELETE 实际时间是 $O(x.degree + c)$ 。

3 Exercise 21.2-1

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。

解

MAKE-SET(x)

- 1 Let o be an object with three fields, next, value, and set
- 2 Let L be a linked list object with head = tail = o
- 3 $o.next = \text{NIL}$
- 4 $o.set = L$
- 5 $o.value = x$
- 6 $L.size = 1$
- 7 **return** L

FIND-SET(x)

- 1 **return** $o.set.head.value$

```

UNION( $x, y$ )
1  if  $x.set.size \geq y.set.size$ 
2       $L1 = x.set$ 
3       $L2 = y.set$ 
4  else
5       $L1 = y.set$ 
6       $L2 = x.set$ 
7   $L1.tail.next = L2.head$ 
8   $z = L2.head$ 
9  while  $z.next \neq \text{NIL}$ 
10      $z.set = L1$ 
11   $L1.tail = L2.tail$ 
12   $L1.size = L1.size + L2.size$ 
13  return  $L1$ 

```

4 OnlineJudge Problem H6-1 在线比赛

解 [Accepted](#)

5 OnlineJudge Problem H6-2 朋友圈

解 [Accepted](#)