HW 6

PB17111603 刘高聪

T1.

(a).证： 命题1：无论区间是否重叠，最大重叠点一定存在。

证明：这个命题是显而易见的。

命题2：存在一个区间[a,b]其中a<=b，使得这个区间内的所有点都是最大重叠点。

证明：由命题1可知，对于区间集合 S，必然存在最大重叠点p。假设与点p重叠的区间有n个，分别为 Interval1，Interval2，，，，，intervaln。n >=1并且n<= S的大小

由于p与这n个区间都有重叠，那么p必然属于这n个区间的公共部分.(这也是一个命题，不过结果显而易见，用反证法即可证明)。那么只需要证明任意两个区间的重叠部分也是一个区间，然后再利用n个区间的重合部分，也是由两个区间重合不断复合得到的，并且这个重叠区间的所有的点的重叠区间数量和最大重叠点p是一样的。

命题3：任意两个区间的重叠部分也是一个区间.

证明：考虑区间A = [s,e],B[s1,e1]。如果e1<s或者s1>e的话那么重叠部分为空区间。如果不是那重叠区间就是[max(s,s1),min(e,e1)]

所以命题得证。

命题4：最大重叠点一定是其中一个区间的端点。

证明：由命题2和命题3可以发现由于区间H非空，并且H是由多个区间复合重叠而成，再结合命题3.可以看见任意两个区间重叠如果非空，那么重叠区间是由原区间的端点组成了新的重叠区间端点。非空区间H一定包含了某个原区间的端点。

(b).

设有n个区间，将所有2n个点从小到大排序，对于排序后的第i个点，若它是某个区间的左端点，则p[i]=1，若它是某个区间的右端点，则p[i]=-1。由第一问可知，所求的点一定是某条线段的端点，所以从端点集合中找出被最多区间覆盖的那个。若一个端点是排序后的第i个点，则有个SUM(s[1],s[i])个区间覆盖这个点。

步骤1：基础数据结构

红黑树，p[x]=1表示它是区间的左端点，p[x]=-1表示它是区间的右端点

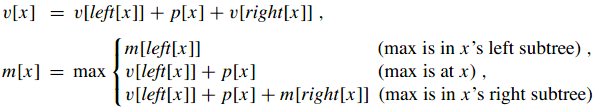
步骤2：附加信息

v[x]：以x为根的所有结点的p值之和

m[x]：以x为根的树中，最大重叠数

o[x]： 以x为根的所有结点中的最大覆盖点

步骤3：对信息的维护



.o[x] = o[left[x]] (max is in x’s left subtree)

. p[x] (max is at x)

. o[right[x]] (max is in x’s right subtree)

. v,m,o都只依赖于左右子树和本身，故于定理14.1，插入和删除操作渐进时间依旧为O(lgn).而FIND-POM操作只需return T.root->o ,时间复杂度为O(1)。

T2.

(a). 实际上第七行所需要的时间与x的孩子数成正比，因为需要将其孩子的父指针指向NIL.

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

T3.

Algorithm 1 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 = NIL
4. o.set = L
5. o.value = x
6. L.length = 1
7. **return** L
8. Algorithm 2 FIND-SET(x)
9. **return**  o.set.head.value
10. Algorithm 3 UNION(x,y)
11. If x.length > y.length
12. L1= x.set
13. L2 = y.set
14. Else
15. L1=y.set
16. L2=x.set
17. L1.tail.next = L2.head
18. z = L2.head
19. **while** z.next != NIL **do**
20. z.set = L1
21. end **while**
22. L1.tail = L2.tail
23. L1.length = L1.length + L2.length
24. **return** L1