6.1

(a)

因为1-Rh和1-Rl是两个常量, 所以可以在循环前提前计算好, 用于在循环中代替1-Rh和 1-Rl. 由此可以将1-Rh和1-Rl这两条指令的总执行次数由N次减少为2次.

因为两个循环的判断条件是一样的, 可以将相同条件下的指令放在一起. 由此可以将判断 BT(I)与M的大小关系的次数从2N次减少到N次.

在计算Th和Tl的过程中, 可以先求同种情况下所有人的收入之和, 最后再一起乘税率得税收之和. 由此计算Th和Tl所需的总乘法次数可以从N次减少为2次.

改进后算法如下:

(b)

由题目条件, 当时, . 所以可以在循环前定义

由此, 假设有i个, 则这条指令从i次简化为1次.

改进后算法如下:

6.6

设时间复杂度为. 易得

由master theorem,

6.9

(a)

树排序是将所有待排序元素插入二叉搜索树的过程. 在最坏情况下, 树将退化成一个链表, 新元素永远插在原链表的尾部. 此时有

所以树排序在最坏情况下是二次时间复杂度的.

(b)

待排序元素为1 2 3 4 5 6 7, 依次将这些元素插入二叉搜索树中, 会得到一条链表: 1->2->3->4->5->6->7, 此时情况同选择排序, 时间复杂度是.

(c)

归并排序将待排序的数组分成大小相同的两部分(两部分大小最多相差一), 分别进行进行归并排序，再对排好序的两部分数组进行合并. 设处理长度为的数组, 归并排序的时间是, 则最坏情况下

所以在最坏情况下, .

(d)

在归并排序过程中, 所谓的平均情况, 最坏情况只是针对两个已排序数组执行合并操作而言的, 容易知道无论两个已排序数组是何种情况, 总是需要时间进行合并, 于是由(c)可知, 归并排序在平均情况下的时间复杂度也是.

补充题：任何基于比较的排序算法，其时间复杂度的下界都是O（nlgn）的（现在无需证明，今后会被其他老师要求证明）。自学桶排序，分析这个算法的worst case时间复杂度，解释为什么这个排序算法突破的nlgn的下界？

worst case下所有待排序数据全分到一个桶里, 这时桶排序退化为基于比较的排序. 此时桶排序的时间复杂度等于任何基于比较的排序算法的时间复杂度, 其下界为.

因为桶排序不是基于比较操作的, 不受到基于比较的排序形成的决策树的限制. 注意到桶排序对于输入的数据有范围上的要求, 且要求有更多的空间, 借助这些比较排序中没有的的限制, 就可以做到突破的下界.

事实上, 假设每个桶内使用的比较排序的时间复杂度为, 桶排序的时间复杂度为, 其中M表示桶的个数. 由于需要申请额外的空间来保存元素, 并申请额外的数组来存储每个桶, 所以空间复杂度为.