

算法基础第三次作业

艾语晨

2020 年 10 月 27 日

目录

1 (Week 6)	2
1.1 稳定的排序算法	2
1.2 Random-Select 的最坏情况	2
1.3 二叉搜索树的最坏时间复杂度	2
1.4 BST 寻找后继的时间复杂度	2

第三次作业 (Week 6)

第 1.1 题 稳定的排序算法

稳定的排序算法：插入排序、归并排序、计数排序不稳定的排序算法：堆排序、快排拓展待排序元素为结构体，以保存其原始下标。在排序完毕之后，对于相同元素，用快速排序排序他们的下标，下标小的在前面。设共有 k 个元素是有相同的，则需要额外

第 1.2 题 Random-Select 的最坏情况

若每一次划分都极不走运地总是按照余下的元素中最大的来进行划分，而划分操作需要 $\Theta(n)$ 时间，所以总的时间复杂度为 $\Theta(n^2)$

第 1.3 题 二叉搜索树的最坏时间复杂度

因为二叉搜索树是有序排列的，所以它必须至少对任意两个数据之间都比较一次，故其时间复杂度至少为 $\Theta(n \lg n)$ 的时间。再考虑到 BST 的建树时间，故最坏情况的时间复杂度为 $\Omega(n \lg n)$ 的

第 1.4 题 BST 寻找后继的时间复杂度

第一次调用 Tree-Successor 的时间复杂度为 $O(h)$ ，而后续的调用只是对于连续的 $k - 1$ 个结点的中序遍历。这个过程只需要对连接这 n 个结点的 $2(k - 1)$ 条边进行访问，于是时间复杂度为 $O(k)$ ，于是总的时间复杂度为 $O(h + k)$