

作业 3: AVL Tree

陈科辉 Keiver Pabula

28 Octobe,2022

作业 3: AVL Tree

1 设计思路

题目要求我们制作一个平衡书,输入元素 k_1, k_2 为上下限遍历树中所有在这之间的元素,并为他们排序输出,并且还需要记录其时间来观察它的复杂性。因此我先拿了头文件 `AVLTree.h` 并为其制作 `Printelement` 函数,其中这个程序符合了 (k_1 为下限, k_2 为上限): 若节点为空,什么也不做; 其实就是在排除 $k_1 > k_2$ 的情况; 在 $k_1 < k < k_2$ 的情况下, 则将元素按顺序排列, 并将他们全部打印出来。

因为是要记录时间,所以使用时间函数。并且因为要多次实验,所以我将主函数设计为多次不同元素的排序。元素每次增加 100, 且 k_1 和 k_2 恒定, 所以元素增加的情况下, 范围不变进行排序。

2 理论分析

首先树有两类: 1 类在要求内, 第 2 类不再要求内。因为是 AVLTree 所有的点应该最多高度只差 1. 设在 k_1 到 k_2 之间有 K 个元素, 所以出输出 K 次, 所以运行 $O(K)$ 次; 除了输出的点还有那些不再 k_1 到 k_2 的点需要跟着进行判断他是否在里面, 而数的高度为 $O(\log N)$, 所以要执行 $O(\log N)$ 次, 所以总共为 $O(K + \log N)$.

3 数值结果

2

3 数值结果

通过结果可以发现符合理论分析的结果，复杂度为 $O(K+\log N)$

[illegible]

3 数值结果

3

[illegible]