作业4

2022111568郑乔译

二叉搜索树对于任何给定的节点，其左子树中所有节点的值都小于该节点的值，而其右子树中所有节点的值都大于该节点的值。这种性质使得BST在搜索、插入和删除操作中具有对数级别（O(log n)）的时间复杂度。

然而，BST的性能高度依赖于其构造。如果输入数据是有序的（如升序或降序），那么BST可能会退化为一个链表，即所有节点都位于同一侧（如都是右子节点），这被称为倾斜树或不平衡树。在倾斜树中，BST的性能会大大降低，因为搜索操作可能需要遍历整个树，时间复杂度接近O(n)，失去了对数级别搜索的优势。

折半查找是一种在有序数组中查找某一特定元素的搜索算法。搜索过程从数组的中间元素开始，如果中间元素正好是要查找的元素，则搜索过程结束；如果某一特定元素大于或者小于中间元素，则在数组大于或小于中间元素的那一半中查找，而且跟开始一样从中间元素开始比较。如果在某一步骤数组为空，则代表找不到。这种搜索算法每一次比较都使搜索范围缩小一半。因此，折半查找的时间复杂度为O(log n)。

因此，分析如下：

数据一：由于数据一是有序的（或具有某种规律），当用于构造BST时，树可能会退化为倾斜树。在这种情况下，BST的性能将显著下降，因为对于许多查询，树可能需要被几乎完全遍历，导致时间复杂度接近O(n)。这与折半查找在有序数组上的性能形成鲜明对比，因为折半查找始终利用有序性来有效地缩小搜索范围。

数据二：数据二是随机排列的，这意味着用于构造BST的数据没有明显的规律。在这种情况下，BST更有可能保持平衡，即树的左右子树高度相近。平衡BST的搜索性能与折半查找相似，因为两者都有效地利用了数据的排序性质来缩小搜索范围。因此，当数据随机排列时，BST和折半查找的平均查找时间性能很接近。