**作业5 查找结构与排序方法**

作业题目：BST查找结构与折半查找方法的实现与实验比较

要求编写程序实现BST 存储结构的建立（插入）、删除、查找和排序算法；实现折半查找算法；比较BST查找与折半查找方法的时间性能。

作业要求：

1. 设计BST 的左右链存储结构，并实现BST插入（建立）、删除、查找和排序算法。
2. 实现折半查找算法。
3. 实验比较：设计并产生实验测试数据，考察比较两种查找方法的时间性能，并与理论结果进行比较。以下具体做法可作为参考：
4. 第1组测试数据： n=1024个已排序的整数序列（如0至2048之间的奇数）；第2组测试数据：第1组测试数据的随机序列。
5. 以上述两组测试数据作为输入，分别建立BST查找结构。
6. 编写程序计算所建的两棵BST的查找成功和查找失败的平均查找长度（主要是改造Search算法，对“比较”进行计数），并与理论结果比较。
7. 分别以上述BST查找结构的中序遍历序列作为折半查找算法的输入，编写程序分别计算折半查找的查找成功和查找失败的平均查找长度，并与理论结果比较。
8. 以上实验能否说明：就平均性能而言，BST的查找与折半查找差不多，为什么？

可以认为，BST查找和折半查找的时间复杂度都是O(logn),但是如果根据有规律的数字建树就会导致BST不能够满足平衡性，当使用特定的数据进行查询的时候就会可能导致性能下降，但是如果给的数字不是规律的，二是随机的，比如第二组测试数据，我们发现他的平均查找时间性能和折半查找时间性能相差很小，如果我们的数字更随机一点(规律性更差)，那么我们有很强的理由认为折半查找和BST查找的时间性能差不多。

测试结果：

