

数据结构与算法基础课程实验报告

实验 3：查找结构的实验比较

姓名	李大鑫	院系	计算机学院	学号	1170300825	
任课教师	臧天仪		指导教师	臧天仪		
实验地点	格物		实验时间			
实验课表现	出勤、表现得分 10%		实验报告 得分 40%		实验总分	
	操作结果得分 50%					
实验目的：						
1. 掌握二叉搜索树的实现 2. 掌握二叉搜索树中的添加、删除、查找操作 3. 掌握二分查找的实现 4. 比较二叉搜索树与二分查找进行查找的时间性能						
实验内容：						
本实验要求编写程序实现 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 查找结构与折半查找的时间性能。						
实验要求：（学生对预习要求的回答）（10 分）					得分：	
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。 BST 的建立通过插入实现，BST 的操作中都需要用到 BST 遍历的基础知识，就是通过将 key 与节点进行比较然后确定是在左子树还是右子树。 删除操作需要额外的注意，因为删除之后还需要维护二叉搜索树的基本性质。 2、实现折半查找： 算法简单实现没有难度。需要注意 while 条件以及 mid 中间点的计算。 3、实验比较？ 实验比较的比较标准是平均查找长度，所谓平均查找长度就是 N 次查找中平均一次查找需要进行比较的次数，用这个标准来衡量一个查找进行的快与慢的问题。						
实验过程中遇到的问题如何解决的？（10 分）（着重从软件调试、质量保证、结果分析方面进行阐述）					得分：	
问题 1： 如果进行性能比较？首先构造 BST 的时候区分两种情况，一种是按照已经排好顺序的数字来建树，一种是按照经过一定随机之后的数字进行建树，首先随机的实现是在已经排好序的数组基础上进行交换得来的，对每一个数组元素进行交换，交换的对象是通过 random() 得到的，这样就实现了一个基本随机的序列。其次进行性能比较的时候对两种查找情况进行分别计算，一种是能够查找得到，一种是不能查找到，分别计算这两种情况下的平均查找长						

度。进行建树的数字选择 0-2048 之间的所有奇数，查找成功的情况依次使用这些奇数，查找不成功的情况使用这些范围内的偶数。这样实现了数据的平均分布，计算出来的平均查找长度是有可信度的。

问题 2:

在二叉搜索树中实现删除的思路：如果我们需要删除一个节点，我们先查找它在不在这棵树中，如果不在报错，如果在的话，我们在树中找到这个节点，删除这个节点之后我们依然需要维护二叉搜索树的基本性质，这里需要进行讨论：如果只有一棵子树，则直接将子树提上来，否则代表两棵子树都存在，这时候我们选择右子树中值最小的节点，将它作为这颗子树新的根节点，因为它是右子树中的最小节点，所以比右子树所有节点都要小，比左子树中所有节点都要大，所以这棵子树依然符合二叉搜索树的定义。对于这个过程使用一个 `deleteMin` 函数来递归查找。

本次实验的体会（结论）（10 分）

得分：

本次主要用实现了 BST 和二叉查找，然后通过比较这两种算法的查找得出了平均查找长度作为性能的衡量。

- 1) 对于 BST 的删除操作，个人感觉这种维护树的基本性质的算法十分值得借鉴。
- 2) 对于比较的结果：显然按照顺序建立的 BST 性能十分差，其查找次数与原值在建树序列中的位置成正比，平均查找次数（两种情况）为 512。随机建树的性能显然更好，平均查找次数能查找到的情况平均为 11.55 不能查找到的情况平均为 12.55，对于二分查找，能查找到的情况平均为 9，不能查找到的情况平均为 10，相比之下二分查找性能更好一些，因为我们随机得到的序列显然不是完全随机的，因此建树的时候 BST 不能保证完全平衡，这就使一次查找的深度（比较次数）会偏离 $O(\log n)$ ，而达不到二分查找 $O(\log n)$ 的效果。

思考题：

思考题 1：（5 分）

得分：

思考题 1：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？

个人认为如果在数据完全随机进行建树且查询数据完全随机的条件下，BST 的查找与折半查找差不多，两者时间复杂度都是 $O(\log n)$ ，但是如果是根据有规律的数字进行建树的话就会导致 BST 不能够满足平衡性因此深度会“深深浅浅”，当使用特定数据进行查询的时候（访问比较“深”的节点）就可能会导致性能下降的效果，比如在本次实验中，对于顺序建树，BST 最终成为一条“链”，如果每次都是查找后面的数据则会导致很差的查找性能，对于完全随机的数据进行查找依然效果效果很差。

思考题 2：（5 分）

得分：

思考题 2: 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。

数据是[0,2048]内的全部奇数, 通过程序, 可以得出折半查找查找成功的平均长度为 9.011719, 查找失败的平均长度为 10.000977, 与理论结果相比较应该为 $O(\log n)$ 其中 n 为 1024, 所以 $O(\log 1024) = 10$, 对于查找成功的情况, 在计算 `mid` 时会查找成功直接返回, 不会循环到 $L=R$, 对于查找失败的情况, 需要 `while` 到 $L=R$ 才能够停止, 所以查找失败的平均查找长度更加趋近于理论值 $O(\log n)$, 而查找成功的情况平均查找长度会低。

指导教师特殊评语:

指导教师签字:

日期: