

数据结构与算法基础课程实验报告

实验 4：查找结构

姓名	郭梓贤	院系	计算机科学与技术学院	学号	1170500112	
任课教师	臧天仪		指导教师	臧天仪		
实验地点	格物楼 207		实验时间	周六 13:30-15:30		
实验课表现	出勤、表现得分		实验报告得分 40%		实验总分	
	10%					
	操作结果得分					
50%						

实验目的：

1. 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。

2. 实现折半查找算法。

3. 比较 BST 查找结构与折半查找的时间性能。

实验内容：

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。

实验要求：（学生对预习要求的回答）（10 分）

得分：

BST定义：

二叉排序树或是一棵空树，或者是具有下列或者是具有下列性质的二叉树：

1、若左子树不空，则左子树上所有结点的值均小于它的根结点的值；

2、若右子树不空，则右子树上所有结点的值均大于它的根结点的值；

3、左、右子树也分别为二叉排序树。

BST 的查找：

从根节点开始比较，如果大于根，则到右子树继续比较，反之，就到左子树继续比较，知道找到相应键值的结点或者不存在相应键值的结点。

比较次数不超过树的深度，树的结构与原始数据的顺序有关，计算得BST的平均查找长度为 $O(\ln n)$.

中序遍历二叉排序树可以得到一个关键字的有序序列。

BST的删除可分为三种情况：

待删除的是叶结点，直接free并将其父结点相应指针修改为空即可；如果待删除的结点只有单个子树，则直接将此子树拼接到待删除节点父结点即可；如果待删除结点有两个子树，则使用待删除结点的直接后继（或前继）替代删除的结点，然后整理指针。

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：
<p>问题 1：</p> <p>建树采用的是插入法，依次将提供的数据插入到原有的树中，使其满足二叉排序树的性质。寻找插入位置时不仅要记录其父结点，还有确定其位于父结点的左侧还是右侧。</p> <p>查找较为简单，根据与根节点的大小关系向左右继续比较直至找到结点或不存在所求结点即可。</p> <p>排序使用中序遍历输出即可，使用递归方式进行中序遍历。</p> <p>删除分为三种情况：待删除的是叶结点，直接free并将其父结点相应指针修改为空即可；如果待删除的结点只有单个子树，则直接将此子树拼接到待删除节点父结点即可；如果待删除结点有两个子树，则使用待删除结点的直接后继（或前继）替代删除的结点，然后整理指针。同样需要记录删除节点与父节点的位置关系。</p> <p>问题 2：</p> <p>统计查找次数过程中，在原有 BST 查找函数和二分查找函数中加入统计查找次数的变量，计算将所有数据元素访问一遍所使用的比较次数除以所有元素的总数得到查找每个元素的平均查找长度。因为查找长度取决于比较的次数，而从根节点的路径长度，所以计数依据比较次数。</p> <p>因为树中的数据都是奇数，所以查找成功的次数使用奇数进行测试，查找失败的次数采用偶数进行测试</p>	
本次实验的体会（结论）（10 分）	得分：
<p>查找长度结果如下：</p> <pre>average length of ordered_list BST:512.5 average length of random_list BST:12.4 average length of find failure ordered_list BST:512.5 average length of find failure random_list BST:13.4 average length of ordered_list BS:9.0 average length of find failure ordered_list BS:10.0</pre> <p>可以看出，当 BST 建树的数据已经是有序的时候，查找相当于有序表的顺序查找，所以长度平均为总长度的一半。而乱序数据建树的平均查找长度，符合 $O(\ln n)$ 的复杂程度。查找失败的平均长度同理。</p> <p>另外，BST 的建立是从完全随机的序列开始进行的，所以很有可能出现树形的不平衡导致的树的深度偏大，而二分查找生成的判定树是平衡二叉树，深度是最小的，所以二分查找的复杂度和 BST 查找的复杂度存在一个常值比例系数，平均查找长度小于 BST，当数据量比较大时，二分查找的性能优势体现得更明显。</p>	

思考题：（10 分）	
思考题 1：（5 分）	得分：
<p>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？</p> <p>实际理论计算得出的结果是：BST 平均查找长度为 $2 \cdot \ln n$，二分查找平均长度为 $\ln(n+1)/(\ln 2)-1$，实验结果也与计算值符合的比较好，虽然两者复杂度处于同一数量级，但是仍有一个比例系数约为 1.4，当查找数据量比较大时，二分查找的性能优势体现得比较明显。所以不能说 BST 的查找与折半查找差不多。</p>	
思考题 2：（5 分）	得分：
<p>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。</p> <p>实验结果查找成功平均长度 9，查找失败的平均长度为 10 理论计算结果为 $\log_2(n+1)-1=9$，符合得比较好。 查找失败时的平均长度略多于查找成功的长度，在进行查找过程中，查找判定树中存在的元素可能在较低的深度完成，而查找判定树中不存在的元素则必然要找到判定树最深层，所以查找失败的平均长度略高于理论值。</p>	
指导教师特殊评语：	
<div>指导教师签字： 日期：</div>	