|  |  |  |  |
| --- | --- | --- | --- |
| **《数据结构与算法》实验报告** | | | |
| **学生姓名** | 曾钰城 | **院（系）** | 计算机科学与技术 |
| **学 号** | 1173710105 | **专 业** | 软件工程 |
| **实验时间** | 2018年12月19日（周三） | **实验地点** | 格物213室 |
| **实验项目** | **实验4/4：**查找结构的实验比较 | | |
| **实验目的：**将课程的基本原理、技术和方法与实际应用相结合，训练和提高学生组织、存储和处理信息的能力，以及复杂问题的数据结构设计能力和程序设计能力，培养软件设计与开发所需要的实践能力。  **实验要求：**灵活运用基本的数据结构和算法知识，对实际问题进行分析和抽象；结合程序设计的一般过程和方法为实际问题设计数据结构和有效算法；用高级语言对数据结构和算法进行编程实现、调试，测试其正确性和有效性。 | | | |
| **实验内容：BST查找结构与折半查找方法的实现与实验比较**  **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。**  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：  （1）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。  （2）按上述两组序列的顺序作为输入顺序，分别建立BST。  （3）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。  （4）以上述BST的中序遍历序列作为折半查找的输入，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。  （5）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？ | | | |
| 数据结构定义：  //结点  struct node  {  int data; //关键字  struct node \*lchild; //左子树  struct node \*rchild; //右子树  };  //根节点  typedef struct  {  struct node \* child; //根节点  }ROOT; | | | |
| 算法设计与分析（要求画出核心内容的程序流程图）：  建立二叉排序树：    插入节点：    删除一个节点：    找子树最小节点（返回该节点的信息，并且删除节点）：    查找节点信息：    查找二叉树排序（中序遍历）：    查找二叉树性能分析：    建立折半查找列表：    选择排序：    折半查找：    折半查找性能分析： | | | |
| 实验测试结果及结果分析：  本程序实现的功能：     1. 建立二叉查找树：      1. 建立二叉查找列表：      1. 在查找二叉树中插入一个节点：      1. 在查找二叉树中删除一个节点：      1. 在查找二叉树中查找一个节点：      1. 用查找二叉树排序：      1. 分析两种输入序列的查找二叉树的查找时间性能     当随机序列输入时，构造的查找二叉树性能介于满查找二叉树与线性查找列表之间，故ASL介于log2(n+1)-1到(n+1)/2之间，将n=1024代入得到log2(1024+1)-1=9，(n+1)/2=512.5，故9<=ASL<=512.5，由实验数据得到  ASL=11.528，平均查找失败长度为每个失败点的深度求和，再取平均，平均查找失败长度 数值上等于 平均查找成功长度 + 1，故查找失败平均长度为12.528  当顺序序列输入时，构造的查找二叉树退化为线性查找列表，故  平均成功查找长度公式为：    将1024代入得，得到平均成功查找长度为：512.5  平均查找失败长度为每个失败点的深度求和，再取平均，平均查找失败长度 数值上等于 平均查找成功长度 + 1，故查找失败平均长度为513.5   1. 分析折半查找的查找时间性能   折半查找性能与满二叉查找树排序一样，可以得出以下公式：    理论上查找成功平均长度为：log2(n+1)-1  查找失败平均长度为：log2(n+1)    这里n=1024，故代入上式，查找成功平均长度为9  平均查找失败长度为每个失败点的深度求和，再取平均，平均查找失败长度 数值上等于 平均查找成功长度 + 1，故查找失败平均长度为10  问题：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？  不一定，当输入序列为有序时（升序或者降序），构造出的查找二叉树会退化为线性列表，平均查找长度ASL接近(n+1)/2，性能不及折半查找  当输入序列比较混乱时，构造出来的查找二叉树会逐渐逼近满二叉树，平均查找长度ASL接近log2(n+1)-1，性能接近折半查找，一下是实验数据：   |  |  |  | | --- | --- | --- | | 实验次数 | 查找成功平均长度 | 查找失败平均长度 | | 1 | 12.242 | 13.242 | | 2 | 11.302 | 12.302 | | 3 | 11.897 | 12.890 | | 4 | 13.485 | 14.485 | | 5 | 11.562 | 12.562 | | 6 | 11.906 | 12.906 | | 7 | 11.728 | 12.728 | | 8 | 11.422 | 12.422 | | 9 | 11.525 | 12.525 | | 10 | 11.777 | 12.777 |   每次实验结果的ASL都接近log2(1024+1)-1≈9。  综上所述：  当输入序列为有序序列时，BST的查找性能远不及折半查找  当输入序列为无序序列时，BST的查找与折半查找差不多 | | | |
| 问题及解决方法：  指针问题：在函数中改变指针的指向，解决方法：在函数参数中改用指针的指针，传参时传入指针的地址。 | | | |
| 源程序名称：search.c | | | |

注意：正文文字为宋体小4号，图中文字为宋体5号。行距为多倍行距1.25。

源程序与此报告打包提交，压缩包采用学号命名。