信息学院

**实验报告**

|  |  |
| --- | --- |
| 专 业： | 计算机科学与技术 |
| 课程名称： | 数据结构 |
| 班 级： |  |
| 学 号： |  |
| 姓 名： |  |
| 辅导教师： | 赵海霞 |

2024年10月制

成绩：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验时间 | 2024年 月 日 时至 时 | 学时数 | 2学时 | |
| 组内成员 |  | | | |
| 分工情况 |  | | | |
| **1.实验名称**  **树和二叉树的**基本操作及应用 | | | | |
| **2.实验目的**  （1）掌握二叉树的结构特性，各种存储结构及其适用范围；  （2）掌握二叉树遍历算法。  （3）掌握二叉树的基本操作在C/C++/java/Python环境中的实现方法； | | | | |
| **3.实验内容**  **A.基础型实验:**  （1）实现程序，按满二叉树给元素编号并输入的方式构造二叉树，并在其上进行先序遍历操作。（顺序存储结构下的实现）  （2）实现程序，构造二叉树的链式存储结构，并完成递归先序（中序、后序）遍历操作。  **B.提高型实验:**  （1）实现程序，构造二叉树并计算二叉树的深度、叶子结点个数，复制二叉树、二叉树左右之树的交换等操作。  （2）建立哈夫曼树，实现哈夫曼树编码和解码。 | | | | |
| 4.实验步骤（包括结构类型定义、设计思想与实现步骤）  (1) 满二叉树的顺序存储实验：   * 结构定义：使用一维数组存储树节点 * 设计思想：   1. 根节点位于数组下标1   2. 对于任意节点i，其左子节点为2i，右子节点为2i+1   3. 父节点为i/2 * 实现步骤：   1. 创建固定大小的数组   2. 按层次顺序输入节点值   3. 实现先序遍历算法   (2) 二叉树的链式存储实验：   * 结构定义：   struct TreeNode {  char data;  TreeNode \*left;  TreeNode \*right;  };   * 设计思想：使用指针链接节点，实现递归遍历 * 实现步骤：   1. 构造二叉树节点   2. 实现三种遍历方法   3. 释放内存空间   B. 提高型实验的实现步骤：  (1) 二叉树操作实验：   * 设计思想：递归实现各种操作 * 实现步骤：   1. 计算深度：递归比较左右子树深度   2. 统计叶子：递归统计无子节点的节点   3. 复制二叉树：递归复制每个节点   4. 交换左右子树：递归交换每个节点的左右子树   (2) 哈夫曼树实验：   * 结构定义：   struct HuffmanNode {  char data;  int frequency;  HuffmanNode \*left;  HuffmanNode \*right;  };   * 设计思想：   1. 使用优先队列构建哈夫曼树   2. 使用map存储编码表 * 实现步骤：   1. 构建哈夫曼树   2. 生成编码表   3. 实现编码解码功能 | | | | |
| **5.调试过程及实验结果**  （详细记录在调试过程中出现的程序测试数据的选择；记录执行的结果。截图形式呈现））。 | | | |
| **6.总结**（对上机实践结果进行分析，问题回答，上机的心得体会及改进意见）。  **实验分析：**   1. 顺序存储适合完全二叉树，空间利用率高 2. 链式存储更灵活，但需要额外的指针空间 3. 递归算法在树的操作中非常高效 4. 哈夫曼编码可以实现有效的数据压缩   **心得体会：**   1. 理解了树结构在实际应用中的重要性 2. 掌握了递归算法的设计方法 3. 加深了对数据压缩原理的理解   **改进建议：**   1. 实现非递归版本的树操作 2. 优化哈夫曼编码的存储方式 | | | |
| **7.附录**（源程序代码等） | | | |