哈尔滨工业大学

**<<数据结构与算法>>**

**实验报告**

**(2023年秋季学期)**

|  |  |
| --- | --- |
| **姓名：** | **阎发祥** |
| **学号：** | **2023140004** |
| **学院：** | **计算机学院** |
| **教师：** | **赵妍妍** |

实验二 树型结构及其应用

## 一、实验内容

哈夫曼编码与译码方法

哈夫曼编码是一种以哈夫曼树（最优二叉树，带权路径长度最小的二叉树）为基础变长编码方法。其基本思想是：将使用次数多的代码转换成长度较短的编码，而使用次数少的采用较长的编码，并且保持编码的唯一可解性。在计算机信息处理中，经常应用于数据压缩。是一种一致性编码法（又称"熵编码法"），用于数据的无损压缩。要求实现一个完整的哈夫曼编码与译码系统。

## 二、实验过程及结果

本实验步骤较多，主要分为以下几个步骤：

1. 文件的读取

给定相应的文件，使用文件操作函数从中读取全文，然后统计不同字符出现的频数，计算字符权值并将其存入到数组里面

1. 哈夫曼树的构造和编码

采用三叉链表来表示哈夫曼树，给定n个权重，那么叶节点就有n个，非叶节点n-1个。首先来初始化所有节点，用-1表示该节点找不到父亲，然后就开始选择两个叶节点来构造一个新的节点，这个节点的权重为两个儿子的权重之和。选择策略是：从所有的没有父亲的节点找出两个权重最小的节点。用函数SelectMin来完成这一功能，接着比较知道构造完成，得到哈夫曼树，用0表示左儿子，1表示右儿子，这样从根节点到叶节点所经过的所有01串就是对应叶节点的哈夫曼编码。

1. 文件的压缩

得到哈夫曼树之后，将每一个字符对应哈夫曼编码存入到数组中，Encode函数实现此功能，然后用Read函数在字符串中相应标识，此时只是用一系列01串来表示原字符文件，还没有实现压缩。因此接下来每8位二进制串一组，转换成相应的ASCII码实现存储，用BitStrToStr函数实现，注意若字符不足8位要从后面补0，这样就实现了文件的无损压缩。

1. 根据压缩内容解压缩

解码是压缩的逆过程，DecodeFile函数实现这个功能，首先在二进制文件中读取相应字符，把压缩串翻译成01字符串，再根据已经存储下来的字符集一一比较，若相同则翻译成为原字符

1. 计算压缩率和对比译码内容

压缩率计算较为简单，就是用字符平均编码长度乘频率得到平均长度，才将其与8相比，减少的长度之比就是压缩率；同时可以将译码文件直接和原文件字符意义比较，即可判断是否相等。

## 三、实验心得

1.学习了c++文件操作的相应知识，巩固了之前所学，强化应用

2.对树形结构的认识有了进一步加深，熟悉了二叉树的相关操作

3.练习了简单的算法知识，比如从树中选取两个权最小的叶节点

4.强化了数组和字符串的相关操作，学习复习了一些函数的用法和传参的众多操作

5.强化之前所学链表的相关操作，体会数据结构对于解决问题的重要性

6.在实验中遇到了众多的困难，比如开始主函数的设计问题，表明面向对象，封装的思想仍然不到位，应该把大问题逐渐拆分成小问题，再逐个击破。文件知识平时操作较少，遗忘较为严重，需要及时复习补充，加以习题的练习。

测试数据如下：





