数据结构与算法

实验报告

实验题目：利用二叉树结构实现哈夫曼编/解码器

姓 名：\_\_\_\_\_\_\_\_\_\_杨梦明\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

学 号：\_\_\_\_\_\_\_\_\_2021213300\_\_\_\_\_\_\_\_\_\_\_\_\_\_

日 期：\_\_\_\_\_\_\_\_\_\_2022.5.14\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

自我评分：\_\_\_\_\_\_\_\_\_B+\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

自我评分说明：A+，A，B+，B，B-，C，D，分别对应分数95、90、85、80、75、70、60

诚信声明

本人郑重承诺：本实验程序和实验报告均是本人独立学习和工作所获得的成果。尽我所知，实验报告中除特别标注的地方外，不包含其他同学已经发表或撰写过的成果；实验程序中对代码工作的任何帮助者所作的贡献均做了明确的说明，并表达了谢意。

如有抄袭，本人原因承担因此而造成的任何后果。

特此声明。

签名：\_\_\_\_\_\_\_杨梦明\_\_\_\_\_\_\_

日期：\_\_\_\_\_\_2022.5.14\_\_\_\_\_\_\_

1、实验简介

1. 能够对输入的任意长度的字符串 s 进行统计，统计每个字符的频度，并建立哈夫曼树。
2. 利用已经建好的哈夫曼树进行编码，并将每个字符的编码输出。
3. 根据编码表对输入的字符串进行编码，并将编码后的字符串输出。
4. 利用已经建好的哈夫曼树对编码后的字符串进行译码，并输出译码结果。
5. 计算输入的字符串编码前和编码后的长度，并进行分析，讨论赫夫曼编码的压缩效果。
6. 程序框架

（1）**SelectMin**：寻找所有元素中最小的两个值。

（2）**Reverse**:翻转字符串。

（3）**Class huffmanTree**：哈夫曼树类

（3.1）**CreateHTree**:对输入的任意长度的字符串 s 进行统计，统计每个字符的频度，并建立哈夫曼树。

（3.2）**CreateCodeTable**:利用已经建好的哈夫曼树进行编码。

（3.3）**Encode**:根据编码表对输入的字符串进行编码。

（3.4）**Decode**:利用已经建好的哈夫曼树对编码后的字符串进行译码。

（3.5）**Compress**:统计原字符串与编码后字符串的压缩比。

3、关键代码实现

3.1 **CreateHTree创建哈夫曼树**：

（1）对输入的字符串逐一进行遍历，通过链表结构存储字符，若字符在链表中存在，则其权值加一，若不存在，则链表元素加一，得到字符元素及其频度数组。

（2）根据数组初始化huffman树，创建2\*n-1个huffman树结点，并将字符与频度传入，叶子结点在放在数组头部，每个结点的左右孩子与父母都初始化为-1,。

（3）通过SelectMin函数不断遍历数组，将和最小的两个值相加并存储到数组下一位，并修改其左右孩子与父母的值，直到存满，哈夫曼树构建完成。

3.2 **CreateCodeTable利用已经建好的哈夫曼树进行编码，制作编码表**：

（1）创建容量为n的编码表数组，n为叶子结点的个数。

（2）遍历huffman树数组，自顶向下，从叶子结点开始，寻找其父母结点，若该结点为左孩子，则编码中标‘0’，若为右孩子，则编码中标‘1’，直到寻找到根结点，即父母结点值为-1的结点。

（3）通过Reverse函数将编码表数组中各个编码逆置，huffman树编码表构建完成。

3.3 **Encode对输入的字符串对照已有huffman编码表进行编码:**

（1）遍历输入的字符串s，查找已有的HcodeTable数组，若元素存在则添加其编码，最后将整段编码返回。

3.4 **Decode对输入的编码字符串进行解码**：

1. 对编码字符串进行遍历，同时从huffTree数组的最后一个元素，即从根结点开始向下查找，若编码字符为‘0’则跳到左孩子，为‘1’则跳到右孩子，直至左孩子为-1，停止查找，将得到的字符元素添加到解码字符串中，由此遍历完整个编码字符串，将得到的解码字符串返回。

4、不足

未实现菜单的交互，对压缩比的计算不够准确，内存占用方面还有待优化。

1. 心得体会

通过此次实验，深刻地了解了哈夫曼树的结构与存储原理，并自己上手进行实践，远比仅听课要印象更深，也了解到了许多编写代码时会遇到的问题，在发现问题、解决问题的过程中学习到了许多数据结构相关的知识，巩固了代码能力，对一些常用的数据结构有了进一步的认识，为今后的学习奠定了基础。