

## 实验名称：文件压缩与解压

## 班 级： 工学部人工智能1班

## 姓 名： 吴南群

## 学 号： 20214001002

## 日 期： 2022年11月16日

## 指导老师：

1. **实验目的**:掌握哈夫曼编码和解码
2. **实验内容**：

利用哈夫曼编码进行通信可以大大提高信道利用率，缩短信息传输时间，降低传输成本。但是，这要求在发送端通过一个编码系统对待传输数据预先编码，在接收端将传来的数据进行译码（复原）。对于双工信道（即可以双向传输信息的信道），每端都需要一个完整的编/译码系统。试为这样的信息收发站写一个哈夫曼码的编/译码系统。

1. **必做内容**

一个完整的系统应具有以下功能（菜单）：

1)I初始化（Initialization）。从终端读入字符集大小n，以及n个字符和n个权值，建立哈夫曼树，并将它存于文件hfmTree中。

2)E编码（Encoding）。利用已建好的哈夫曼树（如不在内存，则从文件hfmTree中读入）, 对文件ToBeTran中的正文进行编码，然后将结果存入文件CodeFile中。

3)D译码（Decoding）。 利用已建好的哈夫曼树将文件CodeFile中的代码进行译码，结果存入文件TextFile中。

4)P打印代码文件（Print）。将文件CodeFile以紧凑格式显示在终端上，每行50个代码。同时将此字符形式的编码文件写入文件CodePrint中

5)T打印哈夫曼树（Tree Print）。将已在内存中的哈夫曼树以直观的方式（树或凹入表）显示在终端上，同时将此字符形式的哈夫曼树写入文件TreePrint

1. **选作内容**

1)修改你的系统，实现对你的系统的原程序的编码和译码（主要是将行尾符编/译码问题）

2)实现各个转换操作的源/目标文件均由用户自行选择指定。

*测试数据*

用下表给出的字符集和频度的实际统计数据建立哈夫曼树，并实现以下报文的编码和译码：“THIS PROGRAM IS MY FAVORITE”。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 字符 |  | A | B | C | D | E | F | G | H | I | J | K | L | M |
| 频度 | 186 | 64 | 13 | 22 | 32 | 103 | 21 | 15 | 47 | 57 | 1 | 5 | 32 | 20 |
| 字符 | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |  |
| 频度 | 57 | 63 | 15 | 1 | 48 | 51 | 80 | 23 | 8 | 18 | 1 | 16 | 1 |  |

提示：

1）用户界面可以设计为“菜单”方式：显示上述功能符号，再加上“Q”，表示运行Quit。请用户键入一个功能符，此功能执行完毕后再显示菜单，直至某次用户选了“Q”为止。

2）在程序的一次执行过程中，第一次执行I、D或C命令之后，哈夫曼树已经在内存了，不必再读入。每次执行中不一定执行I命令，因为文件hfmTree可能早已建好。

1. **实验环境**

ubuntu20.04 g++

1. **程序结构**

**主函数调用哈夫曼树类的成员函数，在主函数内则打开各种文件，实功能。**

1. **函数功能**

**//建树函数**

**void createHFTree(std::string ch[], int fr[], int n);**

**//输出树函数**

**void printHFTree();**

**//返回树的结构的函数**

**std::string printF(); //返回表格内容**

**//创建字母对应的编码表**

**void CreateHFCode(); //编码表**

**//编码函数**

**std::string Encodestr(std::string str); //编码**

**//解码函数**

**std::string Decode(std::string );//解码**

1. **实验数据与结果**

**实现了基本的功能和读取文件进行输入输出的功能。详细请看视频。**

1. **实验体会**

**编码的细节需要注意，容易出错，可能少一步都会出错。**