实验报告

题目：哈夫曼编/译码器

班级：计算机2019211315 姓名：姜彦 学号：2019211608 完成日期：2020.11.14

1. 需求分析
2. 程序要求能够实现对发送数据进行预先编码，并在接收端将传来的数据进行译码，即为信息收发端设计一个哈夫曼编/译码函数。
3. 能够根据所输入的字符和权值建立一个Huffman树，并将其保存到文件中，而且能够将其以直观形式打印。
4. 编码待编码字符并将所编码的字符显示和保存。
5. 概要设计

需要两个个抽象数据类型，Huffman树，编码数组。

1. 的抽象数据类型定义为：

ADT HuffmanTree{

数据对象：D是具有相同特性的数据元素的集合

数据关系：R1={<ai-1,ai>|ai-1,ai∈D,i=2,…,n }

基本操作：

Initialization(HuffmanTree \*HT,Huffmancode \*HC,int \*n)

操作结果：创建并初始化哈夫曼树。

Selet(HuffmanTree HT,int n,int \*s1,int \*s2)

初始条件：哈夫曼树HT存在。

操作结果：从HT[1..,i-1]选择parent为0且weight最小的两个节点并保存到s1和s2中。

Tree\_printing(HuffmanTree HT,int idx,int level,FILE \*fp)

初始条件：哈夫曼树HT存在。

操作结果：以直观形式打印Huffman树并写入treePrint。

WriteInHmfTree(HuffmanTree HT,Huffmancode HC,int n,int m)

初始条件：哈夫曼树HT存在，编码数组HC存在。

操作结果：将Huffman树保存到hfmTree中。

Encoding(HuffmanTree HT,Huffmancode HC,int n)

初始条件：哈夫曼树HT存在，编码数组HC存在。

操作结果：生成编码数组并将ToBeTran中所要编码的字符串编码并保存到CodeFile。

ReadHfmTree(HuffmanTree \*HT,Huffmancode \*HC,int \*n)

初始条件：哈夫曼树HT为空，编码数组HC为空。

操作结果：读取Huffman树到变量HT,HC

FreeTree(HuffmanTree HT,int m);//释放Huffman树

初始条件：哈夫曼树HT存在，编码数组HC为空。

操作结果：释放Huffman树

}ADT HuffmanTree

1. 编码数组的抽象数据类型定义为：

ADT Huffmancode{

数据对象：D是具有相同特性的数据元素的集合

数据关系：R1={<ai-1,ai>|ai-1,ai∈D,i=2,…,n }

基本操作：

Initialization(HuffmanTree \*HT,Huffmancode \*HC,int \*n)

操作结果：初始化并创建一个编码数组。

Encoding(HuffmanTree HT,Huffmancode HC,int n)

初始条件：编码数组HC存在。

操作结果：生成编码数组并将ToBeTran中所要编码的字符串编码并保存到CodeFile。

findc(Huffmancode HC,char ch,int n)

初始条件：编码数组HC存在。

操作结果：在HC中找到字符ch对应的编码的指针并返回。

Decoding(Huffmancode HC,int n)

初始条件：编码数组HC存在。

操作结果：将CodeFile中代码译码并保存在TextFile中。

FreeCode(Huffmancode HC,int n)

初始条件：编码数组HC存在。

操作结果：释放编码数组。

}ADT Huffmancode

1. 本程序包含三个模块：
2. 主程序模块：

int main(void)

{

初始化；

输入数据

While(命令！=退出)

{

命令类型；

Switch选择命令；

初始化命令；break;

编码命令；break;

译码命令；break;

打印编码命令；break;

打印哈夫曼树命令；break;

输入数据；

}

销毁哈夫曼树与编码数组；

return 0；

}

1. 初始化与编/译码函数模块—将文本进行编码与译码；
2. 哈夫曼树、编码数组单元模块—定义哈夫曼树和编码数组节点的结构

模块间调用关系如下：

主程序模块

初始化与编/译码函数模块

哈夫曼树、编码数组单元模块

1. 详细设计
2. 常量定义

#define OK 1//正常返回标志

#define ERROR 0//错误返回标志

#define INFEASIBLE -1//不可用标志

#define OVERFLOW -2//溢出标志

typedef int Status;

1. Huffman树结构定义

typedef struct

{

int weight;

int parent,lchild,rchild;

}HTNode,\*HuffmanTree;

1. 编码数组定义

typedef struct //编码数组

{

char ch;

char \*code;

}HCNode,\*Huffmancode;

基本操作如下：

//哈夫曼树操作

Status Selet(HuffmanTree HT,int n,int \*s1,int \*s2);//HT[1..,i-1]选择parent为0且weight最小的两个节点

Status Initialization(HuffmanTree \*HT,Huffmancode \*HC,int \*n); //读入数据并建立Huffman树

void Tree\_printing(HuffmanTree HT,int idx,int level,FILE \*fp);//直观形式打印Huffman树并写入treePrint

Status FreeTree(HuffmanTree HT,int m);//释放Huffman树

//编码数组操作

Status Encoding(HuffmanTree HT,Huffmancode HC,int n);//生成编码数组并将ToBeTran中所要编码的字符串编码并保存到CodeFile

char \*findc(Huffmancode HC,char ch,int n);//找到字符ch对应的编码的指针主函数及其他函数

Status Decoding(Huffmancode HC,int n);//将CodeFile中代码译码并保存在TextFile中

Status FreeCode(Huffmancode HC,int n);//释放编码数组

int main(void)

{

变量初始化;

int n;//编码字符个数

FILE \*TreePrint\_fp;//直观哈夫曼树待保存的位置

HuffmanTree HT=NULL;//哈夫曼树初始化

Huffmancode HC=NULL;//编码数组初始化

while (不为退出操作时)

{

switch (操作类型)

{

case 1:

Initialization(&HT,&HC,&n);

break;

case 2:

if(HC==NULL)ReadHfmTree(&HT,&HC,&n);//若哈夫曼树为空则从文件读入

             Encoding(HT,HC,n);

Break；

case 3:

Decoding(HC,n);

Break

case 4:

Print();

Break

case 5:

TreePrint\_fp=fopen("TreePrint.txt","r+");

Tree\_printing(HT,2\*n-1,0,TreePrint\_fp);

fclose(TreePrint\_fp);

break

default:

break;

}

更新指令；

}

FreeTree(HT,2\*n-1);//释放哈夫曼树

    FreeCode(HC,n);//释放编码数组

return 0;

}

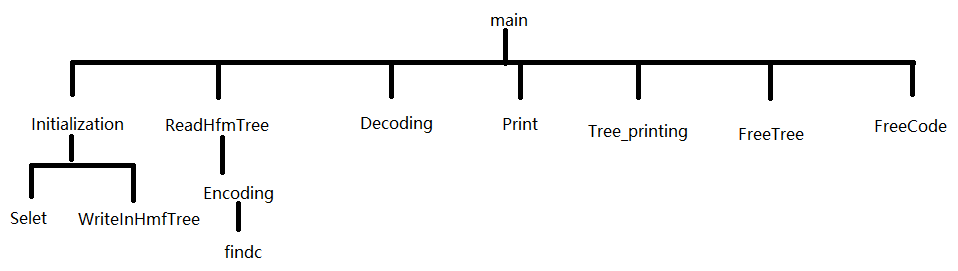
Status WriteInHmfTree(HuffmanTree HT,Huffmancode HC,int n,int m);//将Huffman树保存到hfmTree

Status ReadHfmTree(HuffmanTree \*HT,Huffmancode \*HC,int \*n);//读取Huffman树到变量

int FileLineNum(char name[]);//求文件行数

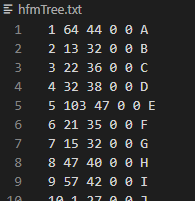
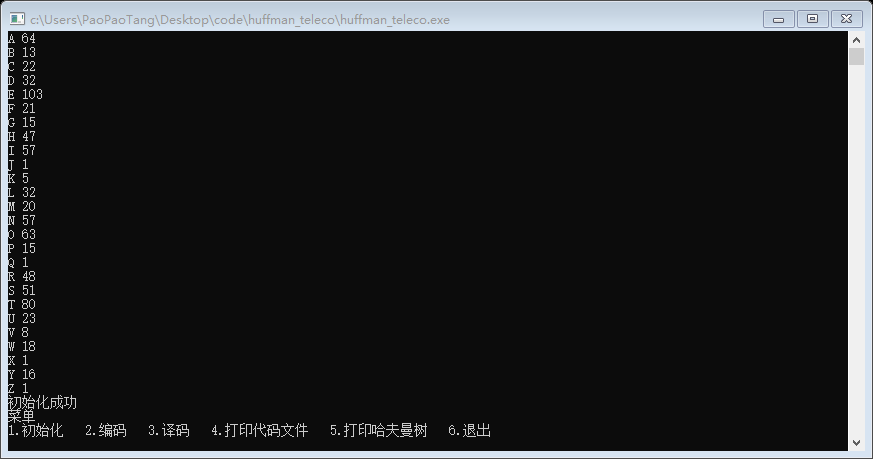
Status Print(void);//将所编译的代码打印并保存在CodePrin

1. 函数调用关系图



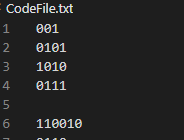
1. 调试分析
2. 当哈夫曼树为空而要进行编码的时候要先读取hfmTree中可能已经保存了的哈夫曼树，变量未赋值时进行操作会导致程序出错。
3. 哈夫曼树、编码数组等malloc生成的变量使用完后要及时销毁释放内存。
4. 该系统运行过程中文件操作较多，打开文件后要及时将文件流关闭。
5. 用户手册
6. 本程序运行环境为windows操作系统，执行文件为huffman\_teleco.exe
7. 进入程序后显示菜单界面以及对应的不同操作命令
8. 输入1后对哈夫曼树进行初始化，回车确认，然后依次输入2，3，4，5执行编码、译码、打印编码、打印哈夫曼树操作，每步操作后显示反馈结果，最后输入0后回车退出系统。
9. 测试结果

先进行初始化，输入1后，依次从终端读入字符集大小n，以及n个字符和n个权值



输入2进行编码

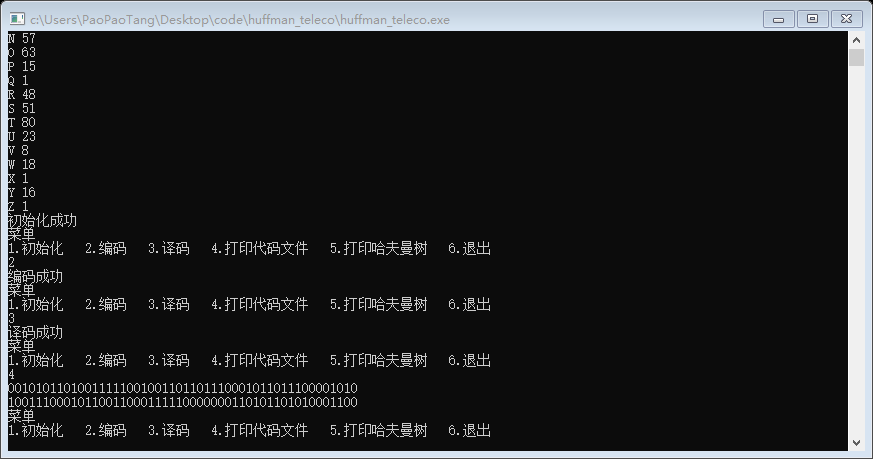




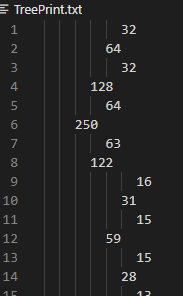
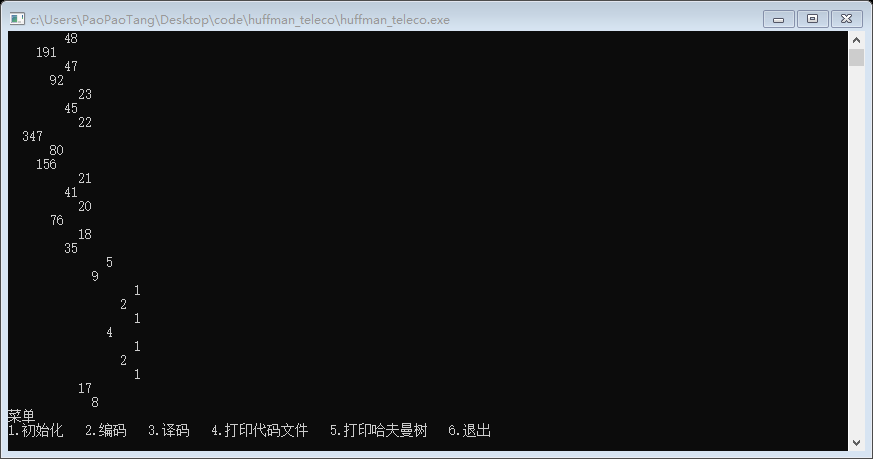
输入3进行译码



输入4打印编码



输入5打印哈夫曼树



输入6结束程序。

七、附录

源程序文件名清单：

huffman\_teleco.c //主程序

输入.txt //输入示例

hfmTree.txt//Huffman树所保存的文件

ToBeTran.txt//待编码文本

CodeFile.txt//文本编码所编码生成的编码文件

TextFile.txt//译码所生成的文本

CodePrin.txt//紧凑形式的编码文件

TreePrint.txt//直观形式Huffman树