**南昌航空大学**

**19学年—20学年第 2 学期 医疗软件技术基础**

**实验二**

专业名称： 生物医学工程 实验学时： 2

学 号： 18084124 姓 名： 杨振宇

实验题目：二叉树的应用－哈夫曼编码的实现

实验环境： vs2008

实验目的：

1．掌握二叉树的定义；

2．掌握哈夫曼树和哈夫曼编码算法的实现。

实验内容：

实现一个哈夫曼编码系统，系统包括以下功能：

(1) 字符信息统计：读取待编码的源文件SourceFile.txt，统计出现的字符及其频率。

(2) 建立哈夫曼树：根据统计结果建立哈夫曼树。

(3) 建立哈夫曼码表：利用得到的哈夫曼树，将各字符对应的编码表保存在文件Code.txt中。

(4) 对源文件进行编码：根据哈夫曼码表，将SourceFile.txt中的字符转换成相应的编码文件ResultFile.txt。

实验流程图

输出字符编码

结束

输入一串要压缩的字符串

统计个字符串的频率

建立哈夫曼树

实验要求：

(1) 程序要具在一定的健壮性，即当输入数据非法时，程序也能适当地做出反应。

(2) 程序要添加适当的注释，程序的书写要采用缩进格式。

实验代码：

#include <stdio.h>

#include<stdlib.h>

#include<string.h>

#include<map>

#include <fstream>

#include<iostream>

#define MAXN 50

using namespace std;

typedef char \*\*HuffmanCode;

typedef struct{

char zimu;

int weight;

int parent, lchild, rchild;

}HTNode,\*HuffmanTree;

HuffmanTree HT;//树,声明

HuffmanCode HC;//节点，声明

int m,s1,s2,start,c,f;

char \*cd;

int z[30];//统计权

int h[30];

//int weight[MAXN] ;//权重

char u[30];

char u\_1[30];

void Select(HuffmanTree HT,int n,int &s1,int &s2)//Select函数，s1,s2其实没有必要

{

int min1 =0x3f3f3f3f;

int min2 =0x3f3f3f3f;//这里不管，这样写是为了开始必然进入循环

for(int i=1;i<=n;i++)//选中最小的权的两个，返回给s1,s2

{

if(HT[i].parent) continue;//不是根节点

if(HT[i].weight < min1)

{

min1 = HT[i].weight;

s1 = i;

continue;

}

if(HT[i].weight < min2 && HT[i].weight >= min1)

{

min2 = HT[i].weight;

s2 = i;

}

}

}

void CreatHuffmanTree(HuffmanTree &HT ,int n ) //n统计字符数

{

if ( n <= 1 ) return;

int m = 2 \* n ;//字符数的两倍

HT = new HTNode[m+1] ;//指向大小

for (int a = 1 ; a <= m ; a ++ )

{

HT[a].parent = HT[a].lchild = HT[a].rchild = HT[a].weight = 0;

}

for (int j = 1 ; j <= n ; j ++ )

{

HT[j].weight = h[j]; //权值传递????

HT[j].zimu = u[j];

}

int s1 ,s2 ;

for (int i = n + 1 ; i <= m ; i ++ ) //次数与上面的相同,父亲节点的与字节点的关系2n关系

{

Select(HT ,i - 1 ,s1 ,s2 );

HT[s1].parent = i ; HT[s2].parent = i ;

HT[i].lchild = s1 ,HT[i].rchild = s2 ;

HT[i].weight = HT[s1].weight + HT[s2].weight ;//定义父节点的权

}

}

void CreatHuffmanCode(HuffmanTree HT, HuffmanCode &HC, int n)

{

HC = new char\*[n+1];//二维

cd = new char[n];//一维\0一个位置

cd[n-1] = '\0';

for(int i=1; i<=n; ++i)

{

start = n-1;

c = i;

f = HT[i].parent;

while(f!=0)//

{

--start;

if(HT[f].lchild == c) cd[start]='0';

else cd[start]='1';

c = f;

f = HT[f].parent;//向上回溯

}

u\_1[i]= HT[i].zimu;

HC[i] = new char[n-start];

strcpy(HC[i],&cd[start]);//把这个编码给HC

}

delete cd;

}

int main()

{

// freopen("in.txt", "r", stdin);//文件要和代码放在同一个文件夹

// freopen("out.txt", "w", stdout);

int r=1;

char s[MAXN];

scanf("%s",s);

int len = strlen(s);//得到节点数

memset(z,0,sizeof(z));//赋0

memset(h,0,sizeof(h));//赋0

memset(u,0,sizeof(u));//赋0

for(int i=0;i<len;i++)//统计字符个数

z[s[i]-'A'+1]++;//统计权

int cnt=0;

for(int j=1;j<=26;++j)

if(z[j])//存在

cnt++;//统计字符数

for(int k=1;k<=26;k++)

if(z[k]){

printf("%c:%d\n",'A'+k-1,z[k]);//显示权

h[r]=z[k];

u[r]='A'+k-1;

r++;

}CreatHuffmanTree(HT,cnt);//创建这个哈夫曼树

CreatHuffmanCode(HT,HC,cnt);//创建编码

printf("\n\nHuffmanCode:\n");//输出哈夫曼编码

printf(" i Char Code\n\n");

for(int t=1; t<=cnt; ++t)//这个输出有问题

printf("%2d %c %s\n",t,u\_1[t],HC[t]);

puts("");

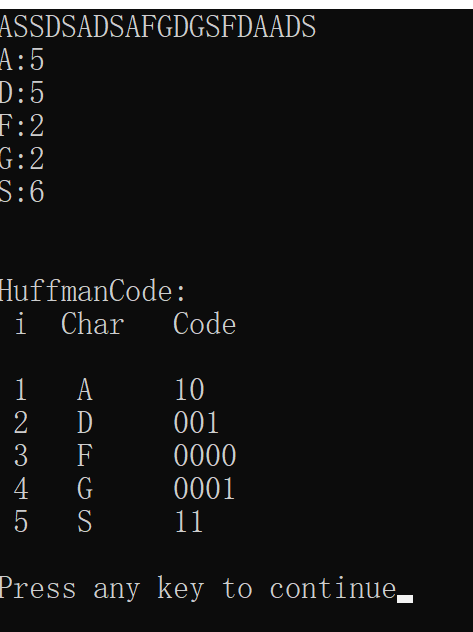
// for(int z=0;z<len;z++)

// printf("%s",HC[s[z]-'A'+1]);

return 0;

}

实验结果：



心得体会：

通过这次的实验，让我复习了前面所学的哈弗曼树，树很重要值得反复学习。认识了二叉树的重要性，在广泛的领域都有用到。