**\_\_\_\_2019\_\_\_\_\_\_级\_\_\_\_\_信1901-1\_\_\_\_\_班\_\_\_\_\_2020\_\_年\_\_\_\_11\_\_\_月\_\_13\_\_\_日**

**姓名\_\_\_\_\_李朴凡\_\_\_\_\_\_\_ 学号\_\_\_\_\_\_20194311\_\_\_\_\_\_ 电话\_\_\_\_\_18333135009\_\_\_\_\_\_\_\_**

**1.实验题目**

利用哈夫曼编码进行通信可以大大提高信道利用率，这要求在发送端通过一个编码系统 对待传输预先编码，在接收端将传来的数据进行译码。对于双工通道，每端都需要一个完整 的编/译码系统。

**2．需求分析**

用cpp进行编码设计。

程序可以做到对哈夫曼树的创建以及编码译码等工作。

键盘中输入或者从文件中读入哈弗曼树；键盘中输入或者从源文件中读入需要编码的源文，然后将源文根据哈弗曼树的权值，译码为二进制代码

**3．概要设计**

首先要定义树的结构体、构建哈夫曼树，然后要将字符转化为整数值，

将输入的字符存入树中。根据字符对应的值对哈夫曼树进行查找，将其翻译或编码。

const int n=maxn //maxn表示叶子数目

const int m=2\*n-1 //m为森林中树的棵数 class tree

{

float weight; //权值

int parent; //双亲

int lch, rch; //左，右孩子

}

tree hftree[m+1]; //规定从第一个元素hftree[1]开始使用数组元素，故定义长

度为m+1而不为m

结构类型：

typedef struct{char data; int weight; int parent; int lchild; int rchild;}huffnode; typedef struct{char cd[MAX]; int start;}huffcode

主程序

int main()

{初始化：输入字符代码以及权值。

编制哈夫曼码：根据权值建立二叉树, 输出相应的根节点到叶结点的路径,便是哈夫曼编码。

编码：输入字符,输出哈夫曼码。 译码：输入哈夫曼,输出字符代码。 退出：结束进程,退出程序。 return 0;}

**4．详细设计**

typedef struct

{ char CH;//字符

int weight;//权值

int parent, lchild, rchild;//双亲，左孩子，右孩子

}DATA;//树的结构体

typedef struct

{ char code[30];

int cnt;

}codetype;

**5．调试分析**

哈夫曼树构造时要注意左右子树的创建

**6．使用说明**

首先：输入你的哈夫曼编码有几种字符

其次：输入每个字符的权值

然后：输入你想翻译的字符串及需要翻译的编码

**7．测试结果**

. 