**《哈夫曼树编码解码》需求分析**

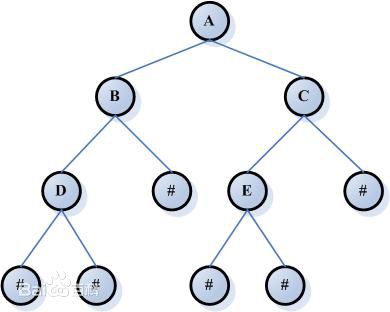
**一、实验内容**

运用哈夫曼编码的相关知识对任意文本文件进行编码、解码。

**二、需要用的数据结构以及实现思路**

需要用到二叉树、HuffMan树、递归等数据结构

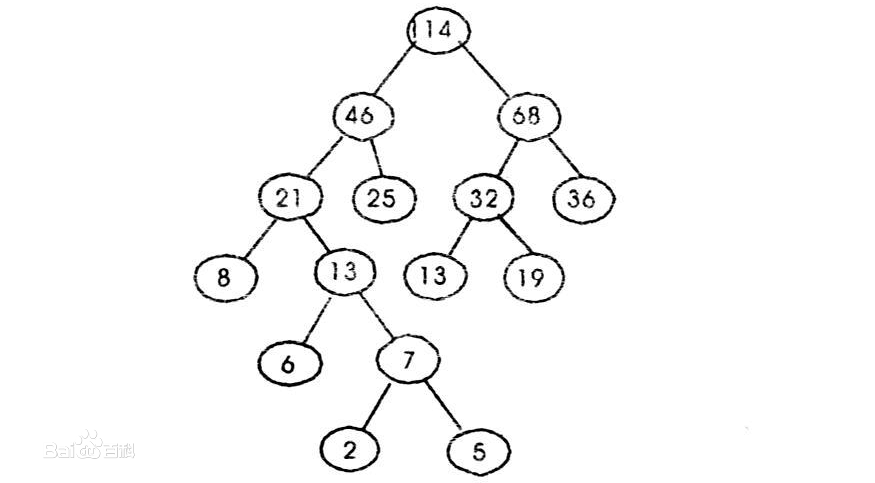
* **二叉树**



在计算机科学中，二叉树是每个节点最多有两个子树的树结构。通常子树被称作“左子树”（left subtree）和“右子  
树”（right subtree）。

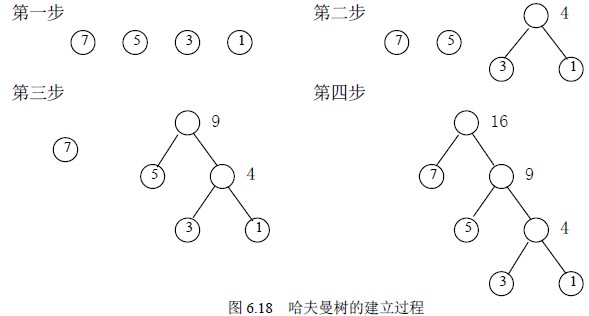
//二叉树节点  
public class TreeNode {  
    public int value;  
    public int parent;  
    public TreeNode leftchild;  
    public TreeNode rightchild;  
    public TreeNode(int v,int p, TreeNode l, TreeNode r){  
        this.value = v;  
        this.parent = p;  
        this.leftchild =l;  
        this.rightchild = r;  
   }  
}

* **HuffMan**



给定n个权值作为n个叶子结点，构造一棵二叉树，若带权路径长度达到最小，称这样的二叉树为最优二叉树，也称为哈夫曼树(Huffman Tree)。哈夫曼树是带权路径长度最短的树，权值较大的结点离根较近

* + **哈夫曼树的构造**



①根据给定的n个权值｛w1， w2， w3， w4......wn｝构成n棵二叉树的森林 F=｛T1 ， T2 ， T3.....Tn｝，其中每棵二叉树只有一个权值为wi 的根节点，其左右子树都为空。

②在森林F中选择两棵根节点的权值最小的二叉树，作为一棵新的二叉树的左右子树，且令新的二叉树的根节点的权值为其左右子树的权值和。

③从F中删除被选中的那两棵子树，并且把构成的新的二叉树加到F森林中。

④重复2 ，3 操作，直到森林只含有一棵二叉树为止，此时得到的这棵二叉树就是哈夫曼树。

//HuffMan树节点类  
  public class HuffmanNode implements Comparable, Cloneable {  
    protected int key;              // 权值  
    protected HuffmanNode left;     // 左孩子  
    protected HuffmanNode right;    // 右孩子  
    protected HuffmanNode parent;   // 父结点  
​  
    protected HuffmanNode(int key, HuffmanNode left, HuffmanNode right, HuffmanNode  
       parent) {  
         this.key = key;  
         this.left = left;  
         this.right = right;  
         this.parent = parent;  
   }  
 }

* + - **HuffMan编码**

从根节点开始，左子加0，右子加1，最终得到每个根节点的huffman编码

**三、HuffMan编码的好处**

Huffman编码处理的是字符以及字符对应的二进制的编码配对问题，分为编码和解码，目的是压字符对应的二进制数据长度。我们知道字符存贮和传输的时候都是二进制的(计算机只认识0/1)，那么就有字符与二进制之间的mapping关系。字符属于字符集(Charset), 字符需要通过编码(encode)为二进制进行存贮和传输，显示的时候需要解码(decode)回字符，字符集与编码方法是一对多关系(Unicode可以用UTF-8,UTF-16等编码)。理解了字符集，编码以及解码，满天飞的乱码问题也就游刃而解了。以英文字母小写a为例, ASCII编码中，十进制为97，二进制为01100001。ASCII的每一个字符都用8个Bit(1Byte)编码，假如有1000个字符要传输，那么就要传输8000个Bit。问题来了，英文中字母e的使用频率为12.702%，而z为0.074%，前者是后者的100多倍，但是确使用相同位数的二进制。可以做得更好，方法就是可变长度编码，指导原则就是频率高的用较短的位数编码，频率低的用较长位数编码。Huffman编码算法就是处理这样的问题。

**四、实验目标**

**算法：使用huffman算法根据要编码的文件中字符出现的频率生成对应的哈夫曼编码对文本文件进行编码和解码**

**展示：通过GUI图形用户界面应用的形式展示出编码前后的信息，并保存到本地。**

**语言：所有代码使用java语言，使用的编译器有Intellij IDEA 、 Android Studio 和 Atom**

**源码：最终源码打包为压缩包并上传到github**

github地址:<https://github.com/wangtianrui/CodeAndDecodeByHuffman>

**五、实验分工**

| **名字** | **学号** | **负责的模块** | **组内担当的位置** |
| --- | --- | --- | --- |
| 王天锐 | 1607094155 | HuffMan树的搭建（节点的获取和链接）、界面的可视化 | 组长 |
| 陈志彬 | 1607094152 | 文件读写操作、大致逻辑和文档的书写 | 组员 |
| 何德庆 | 1607094153 | HuffMan编码类搭建encode方面 | 组员 |
| 郑豪 | 1607094154 | HuffMan编码类搭建decode方面 | 组员 |

**六、时间安排**

1月4日前完成需求分析报告。

1月4日-1月10日，完成相关程序编码工作。

1月11日，撰写课程设计说明书。

1月12日，课程设计验收。

**七、参考资料和获取途径**

数据结构（c语言版）第3版

疯狂Java讲义（第三版）

CSDN

Github

第一行代码