# 赫夫曼树，哈夫曼树,压缩软件

1.电报发送：二战的时候大家都知道那时候普遍会应用电报，如果让你来设计一个电报的发送编码你该如何设计呢？

映射,abcd=>

A:22525

B:3333;

56 78 90

加密：而且还是一个可逆的。

2.压缩算法：给你10000个字符（每个字符1btye，也就是8bit）的文件，你怎么存储可以尽可能的节省空间呢？

10000 \* 8 =80000个空间

重复的去掉：

通信

我相信大家肯定能想到的一个思路就是用某个字符来代替（映射）。比如在压缩算法里面我们可以用二进制来代替

假设字符是 a b c d 4种

那我们假定 a=000 b=001 c=010 d=100,这样我们每个字符就变成了3bit的二进制，那么10000个字符就是30000bit，比起原来的80000bit是不是缩小了很多的存储空间？缩小了将近3倍。

100000001：dab

Abcdaaaaaaaaa：n\*3个bit位

A:0

B：101

C：110

D：100

Abcdaaaaaaaaa：010111010000000000=>abcdaaaaaa

Aaa,前缀。

但是这样做的话会有一个什么问题呢？是不是还有更优的方法呢？哈夫曼编码，也叫前缀编码

满二叉树：除了叶子节点，其他的都有两个子节点，1 2 4 8这样的节点 2^n个点

完全二叉树：除了最底层都有两个子节点，而且叶子节点是靠左连续的

计算下面三颗二叉树的带权路径长度总和：

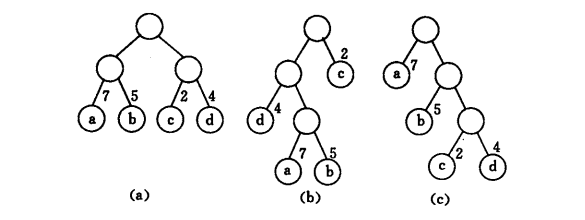
其中每个点的权重为：

a:7 b:5 c:2 d:4

WPL(a):7\*2+5\*2+2\*2+4\*2=36()

WPL(b):7\*3+5\*3+2\*1+4\*2=46()

WPL(c):7\*1+5\*2+2\*3+4\*3=35()



定N个权值作为N个叶子结点，构造一棵二叉树，若该树的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。

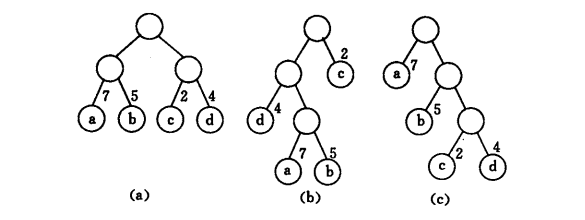
那么这个赫夫曼树和压缩又有什么关系呢？

二叉树：二叉，这时候你要想到二进制，二叉分左右嘛。左节点的边设置为0，右节点的边设置为1

在上图的最优二叉树中我们给每一条边加上一个权值，指向左子节点的边我们标记为0，指向右子节点的边标记为1，那从根节点

到叶节点的路径就是我们说的哈夫曼编码；

所以图c的赫夫曼树对应的编码就是：



A：0

B：10

C：110

D：111

核心思想：贪心算法：利用局部最优推出全局最优，把频率出现多的用短码表示，频率出现小的就用长一点。而且，任何一个字符的编码都不是另一个的前缀，在解压缩的时候，我们每次会读取尽可能长的可解压的二进制串，所以在解压缩的时候也不会产生歧义。

具体实现思路：

1.每次取数值最小的两个节点，将之组成为一颗子树。

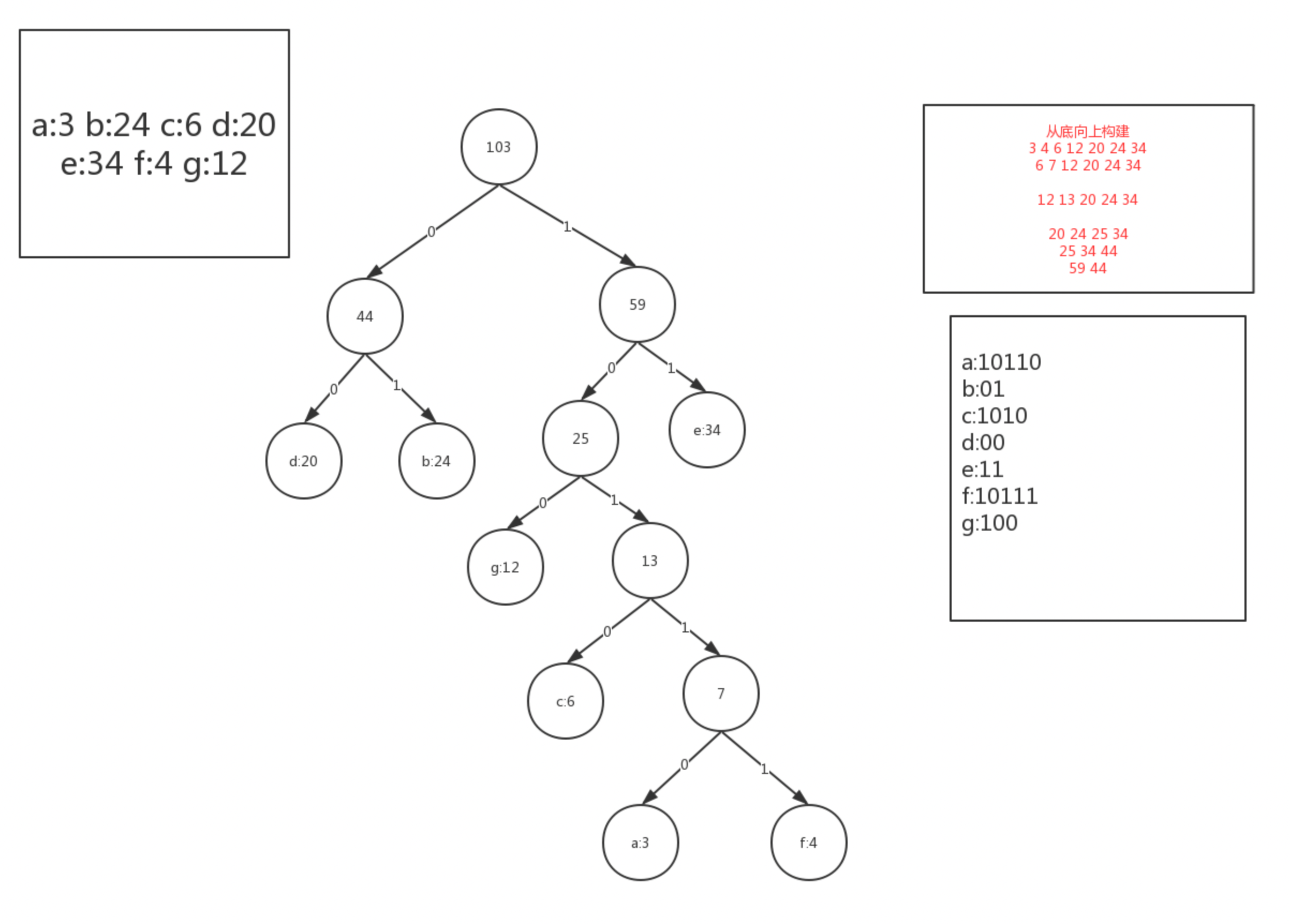
2.移除原来的两个点

3.然后将组成的子树放入原来的序列中

4.重复执行1 2 3 直到只剩最后一个点

例子： a:3 b:24 c:6 d:20 e:34 f:4 g:12

根据以上权重来实现哈夫曼树



学完赫夫曼树，现在可以回到我们的思考题，这两个问题是不是就迎刃而解了。

电报的设计：

1.电报加密后越短越好，发送快。

2.破解难

3.解码容易

4.换加密树也要快

5.可逆的。

什么叫不可逆：MD5这样的Hash加密，简单的MD5已经破解了，穷举：很大的库。Md5(paswd)=-> 接口的密码加密

从前端传到后端我要加密这个密码，Hash函数的了。后面讲

，数据传输的时候回来怎么办？

所以我们现在的很多数字通信里面很多都采用了哈夫曼编码。

压缩我相信大家就更能理解了，只要你明白了哈夫曼编码这个压缩算法其实就很容易了，相信你也可以很快就能实现出来。

有一些稀疏的文件没办法压缩太多