Lesson11——huffman树

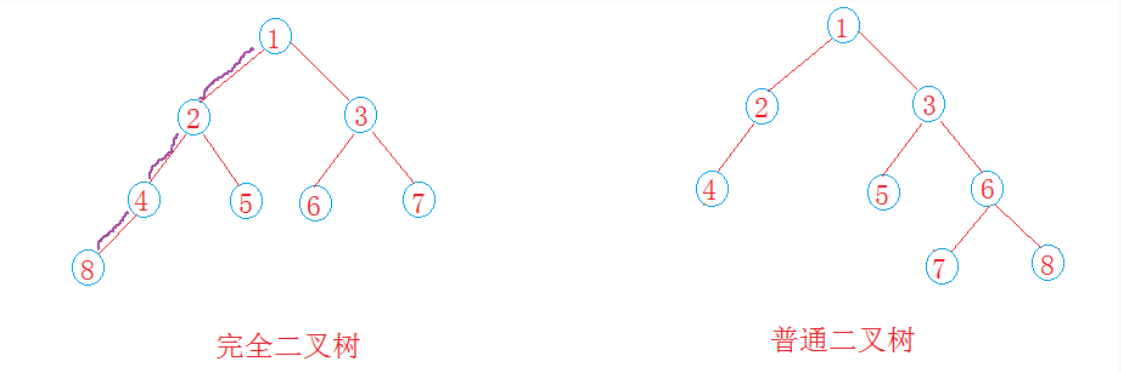
【本节目标】

- **基本概念**
- **HuffmanTree构造算法**
- **Huffman编码**
- **Huffman树的应用：文件压缩与解压缩**

基本概念

在一棵二叉树中：
定义从A结点到B结点所经过的分支序列为从A结点到B结点的路径
定义从A结点到B结点所经过的分支个数为从A结点到B结点的路径长度
从二叉树的根节点到二叉树中所有节点的路径长度之和为该二叉树的路径长度

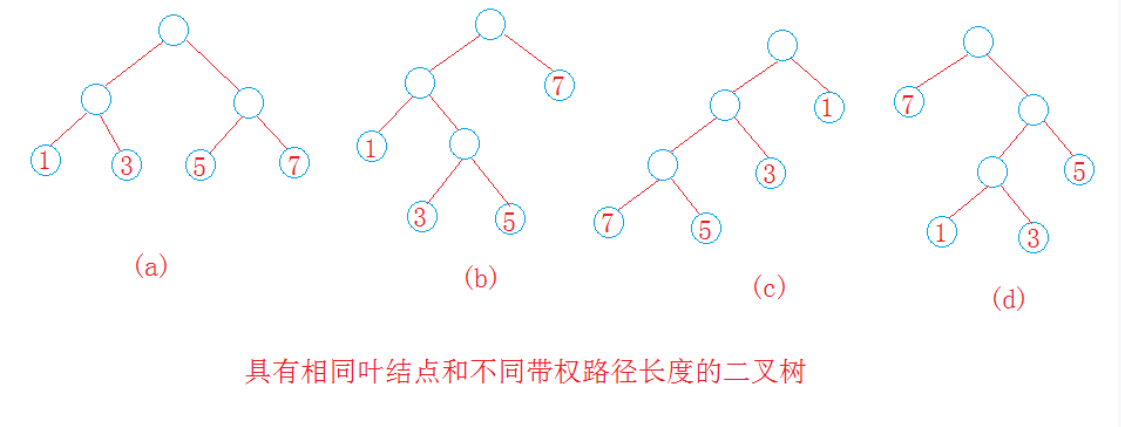
由树的定义可知：**从根节点到达树中任一结点有且仅有一条路径**
若设**根节点处于第1层，某节点处于第k层，则从根节点到其它各结点的路径长度等于该节点所处层次减1**



PL1 = 0 + 1 + 1 + 2 + 2+ 2 + 2 + 3 = 13
PL2 = 0 + 1 + 1 + 2 + 2+ 2 + 3 + 3 = 14

设二叉树有n个带权值的叶结点，定义：
从二叉树的根结点到二叉树中所有叶结点的路径长度与相应权值的乘积之和为该二叉树的带权路径长度WPL。

对于一组具有确定权值的叶结点，可以构造出多个具有不同带权路径长度的二叉树



这4棵二叉树的带权路径长度分别为：

- WPLa = 1 * 2 + 3 * 2 + 5 * 2 + 7 * 2 = 32
- WPLb = 1 * 2 + 3 * 3 + 5 * 3 + 7 * 1 = 33
- WPLc = 7 * 3 + 5 * 3 + 3 * 2 + 1 * 1 = 43
- WPLd = 1 * 3 + 3 * 3 + 5 * 2 + 7 * 1 = 29

把带权路径最小的二叉树称为Huffman树。

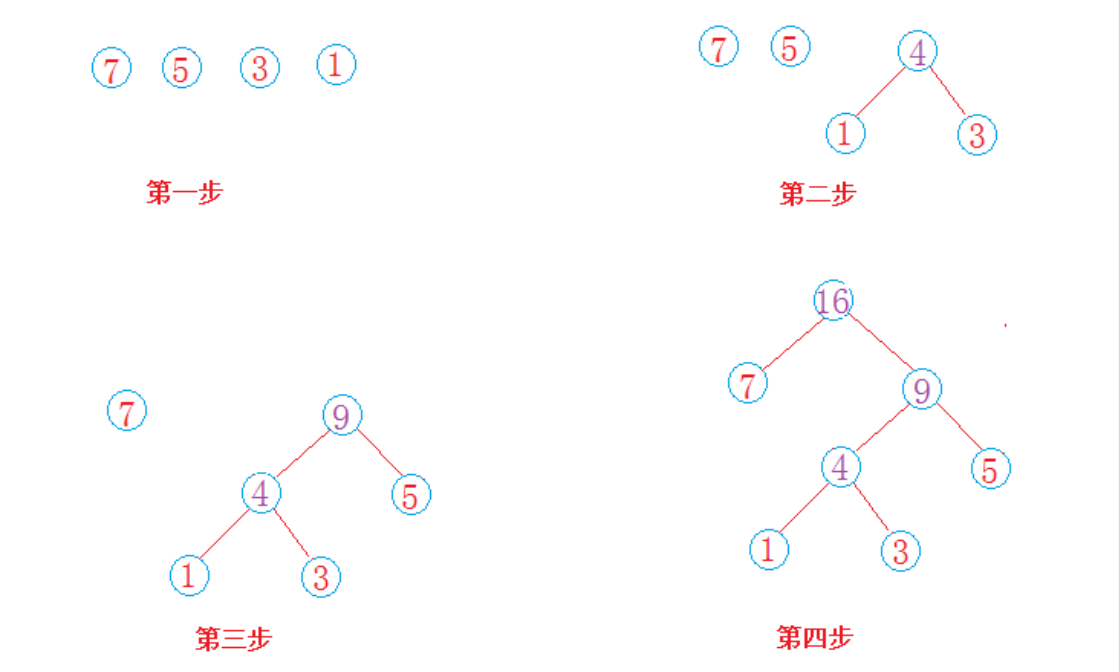
hufuman树的构造

Huffman树构造算法：

- **1. 由给定的n个权值{ w1, w2, w3, ... , wn}构造n棵只有根节点的二叉树森林F={T1, T2 , T3, ... ,Tn},每棵二叉树Ti只有一个带权值wi的根节点，左右孩子均为空。**

- **2. 重复以下步骤，直到F中只剩下一棵树为止：**
- **在F中选取两棵根节点权值最小的二叉树，作为左右子树构造一棵新的二叉树,新二叉树根节点的权值为其左右子树根节点的权值之和**
- **在F中删除这两棵二叉树**
- **把新的二叉树加入到F中**

最后得到的就是Huffman树



huffman编码

在数据通讯中经常需要将传输的文字转换成二进制字符0和1组成的二进制串，称该过程为编码。
假设需要传输ABBBCCCCCDDDDDDD，电文中只有4中字符，可有如下的编码方案：

字符	编码
A	00
B	01
C	10
D	11

等长编码

字符	编码
A	100
B	101
C	11
D	0

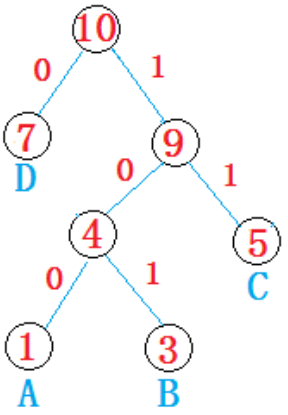
不等长编码

思考：

- **上述的不等长编码是怎么得到?**
- **能否将D的编码给成1？**

ABBBCCCCCDDDDDDD
A->1 B->3 C->5 D->7

A:100
B:101
C:11
D:0



注意：**在建立不等长编码时，任何一个字符的编码不能是另一个字符编码的前缀，这样才能保证译码的唯一性。**
而huffman树中，任何一个叶子结点的huffman编码都不可能是另一个叶节点huffman编码的前缀。
为什么:)？

文件压缩