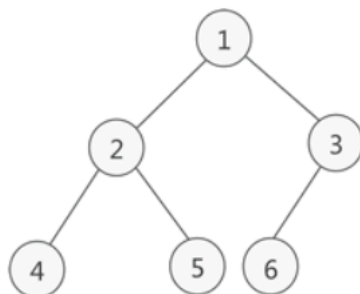


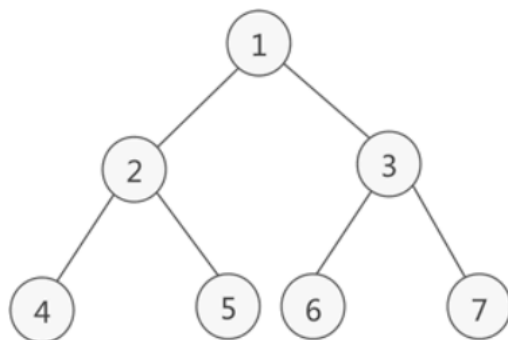
第07课哈夫曼树

哈夫曼编码的意义：信息传输设计，IC指令设计，压缩与解压等

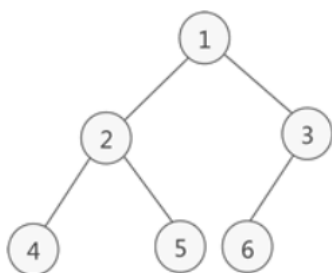
二叉树，哈夫曼树



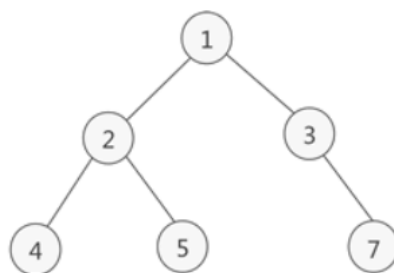
如果二叉树中除了叶子结点，每个结点的度都为 2，则此二叉树称为满二叉树。（左孩子右孩子）



如果二叉树中除去最后一层节点为满二叉树，且最后一层的结点依次从左到右分布，则此二叉树被称为完全二叉树。

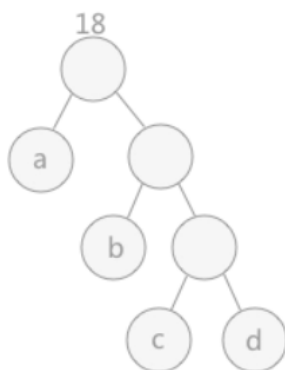


a) 完全二叉树



b) 非完全二叉树

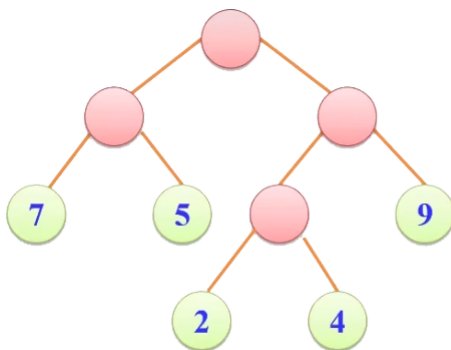
哈夫曼树：是带权路径长度最短的树



如何构建哈夫曼树 (贪心策略: 局部最优)

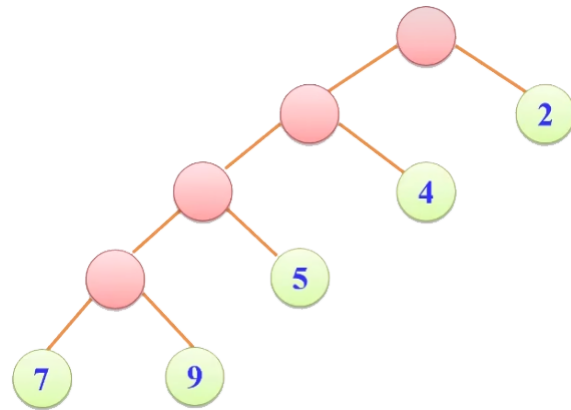
二叉树的带权路径长度WPL

$$WPL = \sum_{i=1}^n w_i l_i$$



$$WPL(T_1) =$$

$$7 \times 2 + 5 \times 2 + 2 \times 3 + 4 \times 3 + 9 \times 2 = 60$$



$$WPL(T_2) =$$

$$7 \times 4 + 9 \times 4 + 5 \times 3 + 4 \times 2 + 2 \times 1 = 89$$

构造哈夫曼树的过程:

(1) 给定的 n 个权值 $\{w_1, w_2, \dots, w_n\}$ 构造 n 棵只有一个叶结点的二叉树, 从而得到一个二叉树的集合 $F = \{T_1, T_2, \dots, T_n\}$ 。

(2) 在 F 中选取根结点的权值最小和次小的两棵二叉树作为左、右子树构造一棵新的二叉树, 这棵新的二叉树根结点的权值为其左、右子树根结点权值之和。

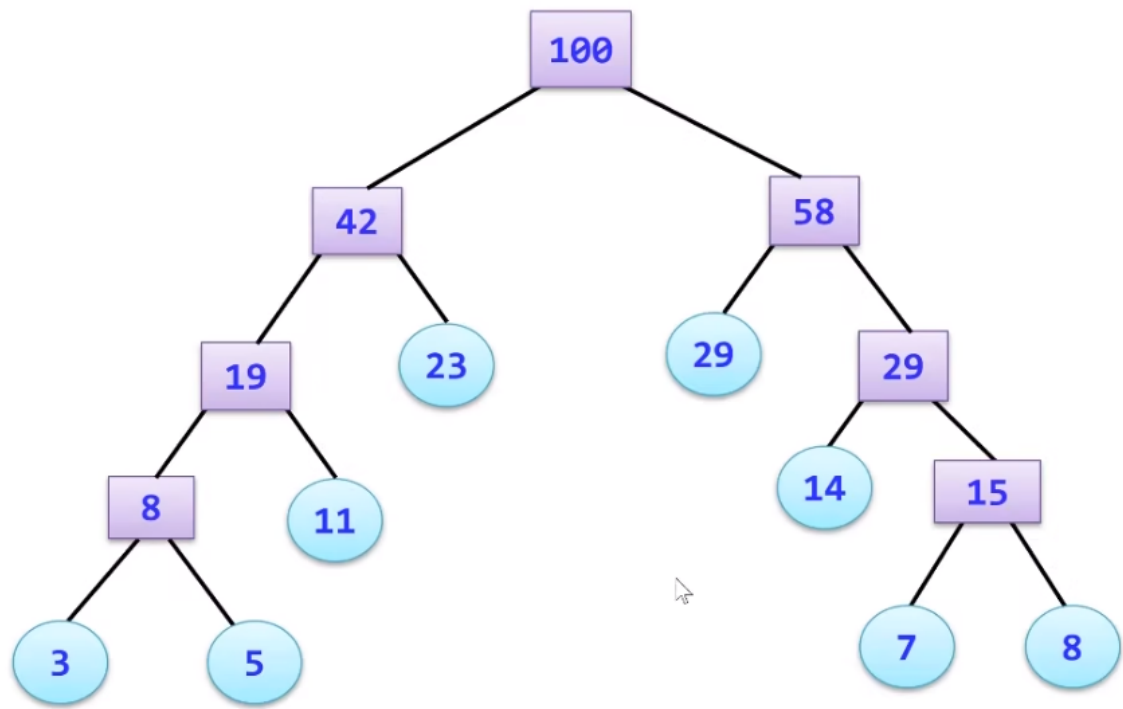
(3) 在集合 F 中删除作为左、右子树的两棵二叉树, 并将新建立的二叉树加入到集合 F 中。

(4) 重复 (2)、(3) 两步, 当 F 中只剩下一棵二叉树时, 这棵二叉树便是所要建立的哈夫曼树。

实例:

$W = \{0.05, 0.29, 0.07, 0.08, 0.14, 0.23, 0.03, 0.11\}$





证明：为什么哈夫曼树的节点个数为 $2n-1$ (n 为叶子节点个数)

讲程序：如何构建哈夫曼树（数组法）

作业任务

任务1：哈夫曼树（C语言版）

输入 n 个带权重的节点（如A, 3, B, 14, C, 8, D, 7, E, 8），生成哈夫曼树数组并显示数组

显示要求：一行显示一个数组元素，元素的分量间用逗号隔开

任务2：哈夫曼树（Python版）

输入 n 个带权重的节点（如A, 3, B, 14, C, 8, D, 7, E, 8），显示生成哈夫曼树图形