哈夫曼树

1. 哈夫曼树可用来构造最优编码，用于信息传输，数据压缩等方面，哈夫曼树是一种广泛应用的二叉树。

2.哈夫曼树的基本概念：

1. 路径和路径长度

**路径：**是指从根节点到该节点的分支序列

**路径长度：**根节点到该节点所经过的分支数目

1. 节点的权和带权路径长度

赋予节点的实数称为**节点的权**

在树结构中，把从树根到某一节点的路径长度与该节点的权的乘积，称为该节点的**带权路径长度**

1. 树的带权路径长度

树中，从根节点到所有叶子节点的各个带权路径长度之和

4

2

5

7

WPL（带权路径长度）=7\*2+5\*2+2\*2+4\*2=36

4)路径长度为k结点至多有2^k个

5）完全二叉树具有最小路径长度的性质但不唯一，有些树不是完全二叉树但也有最小路径长度

6）什么样的二叉树带权路径长度最小？

答：哈夫曼树，哈夫曼树的构造思想就是尽可能把权值大的叶子节点靠近根

3.构造哈夫曼树

什么是哈夫曼树？

答：是由n个带权叶子结点构成所有二叉树中带权路径长度最短的二叉树

4.如何创建哈夫曼树

1）将数组元素从小到大排序，每个数据都是一个节点，每个节点都可以看成一颗没有左右孩子的最简单的二叉树

2）取出根节点最小的两颗二叉树

3）组成一颗新的二叉树，该新的二叉树根节点的权值是前面两颗二叉树根节点的权值之和

4）再将这颗新的二叉树以根节点的权值大小再次排序，不断重复1-2-3-4，直到数组中的元素都被处理过，就得到了一颗哈夫曼树

举例：{13,7,8,3,29,6,1}

1.排序：{1,3,6,7,8,13,29}

2-3-4操作:**注意：每次组合形成二叉树都是选最小的两个权值进行组合**

取出1,3组成新的二叉树，{4,6,7,8,13,29}

3

1

4

取出6，和4再组成新的二叉树{10,7,8,13,29}

3

1

6

4

10

接下来取7,8因为7,8组合最小{15,10,13,29}

取13,10组合{23,15,29}

取23,15{38,29}

取38,29，完成

3

1

6

4

13

10

29

38

23

8

7

15

67