# 哈夫曼树

所谓哈夫曼树，又称最优树，是一类带权路径长度最短的树。

**1、路径和路径长度**

**路径：**从树中一个结点到另一个结点之间的分支构成两个结点的路径。

**路径长度：**某条路径上边的条数。

**树的路径长度WPL**：从树根到每一个叶子结点的路径长度 乘以 叶子节点权值 的之和。

WPL= //它是叶子节点的带权路径长度之和

代表某个叶子节点的权值，一共n个叶子节点，代表从根节点到某个权值为的叶子节点的路径长度

**哈夫曼树整体的权值**：就是除了根节点外其余节点权值总和,因为根节点权值是，所以**哈夫曼树整体的权值**也是所有节点权值和减去

## 构建

给你n个节点:{}构建哈夫曼树

方法是：选取权值最小的两个节点，求它的的权值之和s=建新的节点权值是s，把权值是和的节点当它的儿子。把s当做新的节点，和其他节点继续做上述操作。每次这么做会把2个节点合并成一棵树。到最后执行n-1次后，所有节都会被合并成一棵树，算上新建的节点，最终树的节点总数是2n-1

哈夫曼树的构建需要用到二叉堆。

* 哈夫曼树的形态不唯一，根源是可能中间某个过程出现两个权值相同的节点，就可能产生2个分支。只有当：包括原本的数在内的任何过程中，都没有相同的数时，才能保证哈夫曼树形态唯一。
* n个数可以构建的哈夫曼树有n个叶子

## 哈夫曼编码

定义一个树的某个节点的左边方向为0，右边方向为1，则一个权值的哈夫曼编码定义为：对哈夫曼树的到这个权值所对应的的叶子，走的路径的方向组成的01字符串。

n个数可以构建的哈夫曼树有n个叶子,也就有n个01字符串。由于哈夫曼树不一定唯一，所以这些01字符串也不一定唯一，但哈夫曼树的带权路径长度唯一，都是相比其他构建方法来说是最小的。

# 应用：

## 编码字符串：

给定字符串str,假设它有26\*2个大写和小写字母组成，每个字母对应权值就是出现频率，之前不管字符串str是什么编码，都可以把每个字符变成哈夫曼编码表示，这样有利于压缩存储空间，也正是哈夫曼编码的作用：数据压缩。