

Huffman编码与HZOJ287合并果子题目的关系

1.Huffman编码

Huffman编码算法

哈夫曼（Huffman）编码算法是基于二叉树构建编码压缩结构的，它是数据压缩中经典的一种算法。算法根据文本字符出现的频率，重新对字符进行编码。因为为了缩短编码的长度，我们自然希望频率越高的词，编码越短，这样最终才能最大化压缩存储文本数据的空间。

Huffman树

路径：若树中存在一个结点序列 k_1, k_2, \dots, k_j ，使得 k_i 是 k_{i+1} 的双亲，则称该结点序列是从 k_1 到 k_j 的一条路径。

路径长度：等于路径上的结点数减1。

结点的权：在许多应用中，常常将树中的结点赋予一个有意义的数，称为该结点的权。

结点的带权路径长度：是指该结点到树根之间的路径长度与该结点上权的乘积。

树的带权路径长度：树中所有叶子结点的带权路径长度之和，通常记作：

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

在权为 w_1, w_2, \dots, w_n 的 n 个叶子结点的所有二叉树中，带权路径长度 WPL 最小的二叉树称为Huffman树或最优二叉树。

Huffman树的构造

- 1.根据给定的 n 个权值 $\{w_1, w_2, \dots, w_n\}$ 构成二叉树集合 $F = \{T_1, T_2, \dots, T_n\}$,其中每棵二叉树 T_i 中只有一个带权为 w_i 的根结点,其左右子树为空.
- 2.在 F 中选取两棵根结点权值最小的树作为左右子树构造一棵新的二叉树,且置新的二叉树的根结点的权值为左右子树根结点的权值之和.
- 3.在 F 中删除这两棵树,同时将新的二叉树加入 F 中.
- 4.重复2、3,直到 F 只含有一棵树为止.(得到哈夫曼树)

2.HZOJ287合并果子

我们将每一堆果子距离完成合并需要的操作次数设为 l_i ,每一堆果子的重量设为 w_i ,则最终我们的目标函数为 $\min\{\sum_{i=1}^n w_i l_i\}$, 即为Huffman编码算法。