

# 哈夫曼树

2019年7月1日 15:07

哈夫曼树的定义

**带权路径长度(WPL):** 设二叉树有  $n$  个叶子结点, 每个叶子结点带有权值  $w_k$ , 从根结点到每个叶子结点的长度为  $l_k$ , 则每个叶子结点的带权路径长度之和就是:  $WPL = \sum w_k l_k$

**最优二叉树**或**哈夫曼树: WPL**最小的二叉树

哈夫曼树的创建

每次把**权值最小的两棵**二叉树合并

```
typedef struct TreeNode *HuffmanTree;
struct TreeNode{
    int Weight;
    HuffmanTree Left, Right;
}
HuffmanTree Huffman( MinHeap H )
{ /* 假设H->Size个权值已经存在H->Elements[]->Weight里 */
    int i; HuffmanTree T;
    BuildMinHeap(H); /*将H->Elements[]按权值调整为最小堆*/
    for (i = 1; i < H->Size; i++) { /*做H->Size-1次合并*/
        T = malloc( sizeof( struct TreeNode) ); /*建立新结点*/
        T->Left = DeleteMin(H);
        /*从最小堆中删除一个结点, 作为新T的左子结点*/
        T->Right = DeleteMin(H);
        /*从最小堆中删除一个结点, 作为新T的右子结点*/
        T->Weight = T->Left->Weight+T->Right->Weight;
        /*计算新权值*/
        Insert( H, T ); /*将新T插入最小堆*/
    } T
    = DeleteMin(H);
    return T;
}
```

哈夫曼树的特点:

- ☞ 没有度为1的结点;
- ☞ 哈夫曼树的任意非叶节点的**左右子树交换**后仍是哈夫曼树;
- ☞  $n$ 个叶子结点的哈夫曼树共有 **$2n-1$** 个结点;
- ☞ 对同一组权值 **$\{w_1, w_2, \dots, w_n\}$** , 是否存在**不同构**的两棵**哈夫曼树**呢?

对一组权值 **$\{1, 2, 3, 3\}$** , **不同构**的两棵哈夫曼树:

$wpl=18$

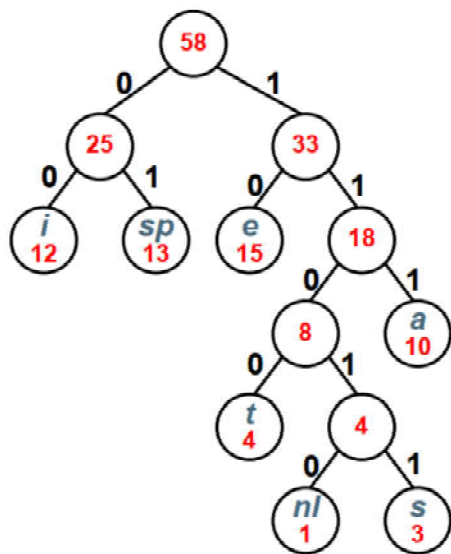
二叉树用于编码

用二叉树进行编码:

- (1) 左右分支: 0、1
- (2) 字符只在叶结点上

【例】哈夫曼编码

$C_i$	$a$	$e$	$i$	$s$	$t$	$sp$	$nl$
$f_i$	10	15	12	3	4	13	1



$a : 111$   
 $e : 10$   
 $i : 00$   
 $s : 11011$   
 $t : 1100$   
 $sp : 01$   
 $nl : 11010$

$$\begin{aligned}
 \text{Cost} &= 3 \times 10 + 2 \times 15 \\
 &\quad + 2 \times 12 + 5 \times 3 \\
 &\quad + 4 \times 4 + 2 \times 13 \\
 &\quad + 5 \times 1 \\
 &= 146
 \end{aligned}$$