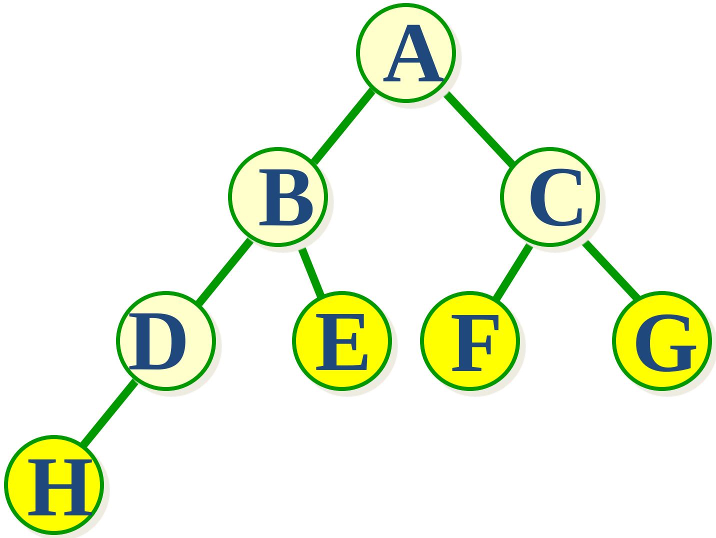
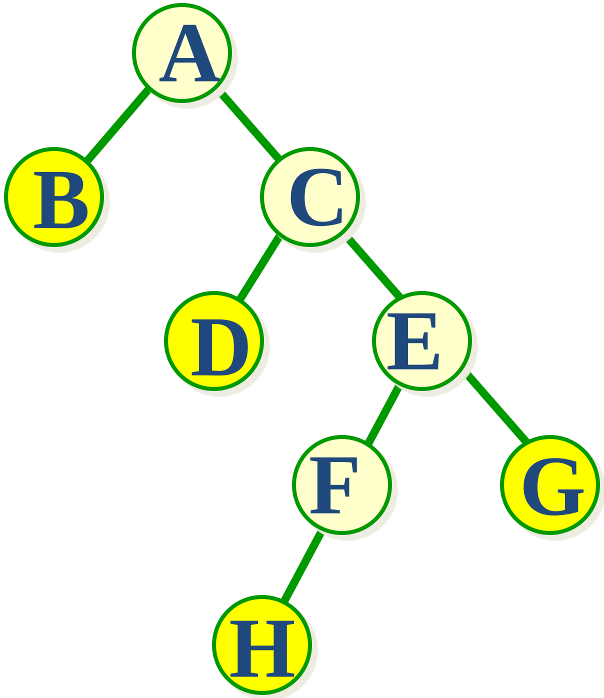
哈夫曼树

路径长度 (Path Length)

- 两个结点之间的路径长度：连接两结点的路径上的分支数。
- 树的外部路径长度（EPL）：各叶结点到根结点的路径长度之和。
- 树的内部路径长度IPL：各非叶结点到根结点的路径长度之和。
- 树的路径长度 PL = EPL + IPL。





IPL = 0+1+1+2 = 4

EPL = 2+2+2+3 = 9

PL = 13

IPL = 0+1+2+3 = 6

EPL = 1+2+3+4 = 10

PL = 16

- n 个结点的二叉树的路径长度不小于下述数列前 n 项的和，即

$$PL = \sum_{i=1}^{n} \lfloor \log_2 i \rfloor$$

$$= 0+1+1+2+2+2+2+3+3+\cdots$$

- 其路径长度最小者为

$$PL = \sum_{i=1}^{n} \lfloor \log_2 i \rfloor$$

- 完全二叉树满足这个要求。

带权路径长度 (Weighted Path Length, WPL)

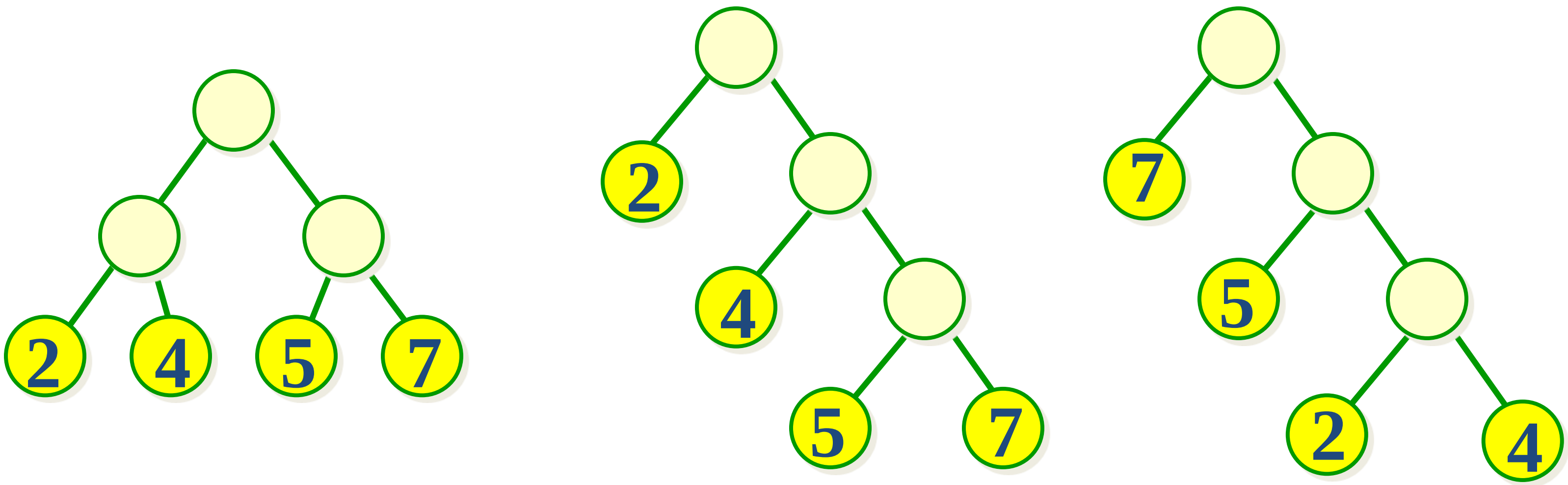
- 在很多应用问题中为树的叶结点赋予一个权值，用于表示出现频度、概率值等。因此，在问题处理中把叶结点定义得不同于非叶结点，把叶结点看成“外结点”，非叶结点看成“内结点”。这样的二叉树称为扩充二叉树。
- 扩充二叉树中只有度为 2 的内结点和度为 0 的外结点。根据二叉树的性质，有 n 个外结点就有 $n-1$ 个内结点，总结点数为 $2n-1$。

- 若一棵扩充二叉树有 n 个外结点，第 i 个外结点的权值为w_i，它到根的路径长度为l_i，则该外结点到根的带权路径长度为w_i*l_i。
- 扩充二叉树的带权路径长度定义为树的各外结点到根的带权路径长度之和。

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

- 对于同样一组权值，如果放在外结点上，组织方式不同，带权路径长度也不同。

具有不同带权路径长度的扩充二叉树



WPL = 2*2+
4*2+5*2+
7*2 = 36

WPL = 2*1+
4*2+5*3+
7*3 = 46

WPL = 7*1+
5*2+2*3+
4*3 = 35

带权路径长度最小

哈夫曼树

- 带权路径长度达到最小的扩充二叉树即为哈夫曼（**Huffman**）树。
- 在**Huffman**树中，权值大的结点离根最近。

Huffman树的构造算法

1. 由给定 n 个权值 $\{w_0, w_1, w_2, \ldots, w_{n-1}\}$，构造具有 n 棵扩充二叉树的森林 $F = \{ T_0, T_1, T_2, \ldots, T_{n-1} \}$，其中每棵扩充二叉树 T_i 只有一个带权值 w_i 的根结点, 其左、右子树均为空。

2. 重复以下步骤, 直到 *F* 中仅剩一棵树为止：

a) 在 *F* 中选取两棵根结点的权值最小的扩充二叉树， 做为左、右子树构造一棵新的二叉树。置新的二叉树的根结点的权值为其左、右子树上根结点的权值之和。

b) 在 *F* 中删去这两棵二叉树。

c) 把新的二叉树加入 *F*。

哈夫曼树的构造

1.构造要求

给定 n 个权值 { w1, w2, ..., wn}，构造 具有 上述权值的n 个叶子结点的扩充二叉树。

（1）权值愈大，离根愈近

根据上述的定义，哈夫曼树是一棵带权路径长度最小的扩充二叉树，即它具有最小的WPL值，

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

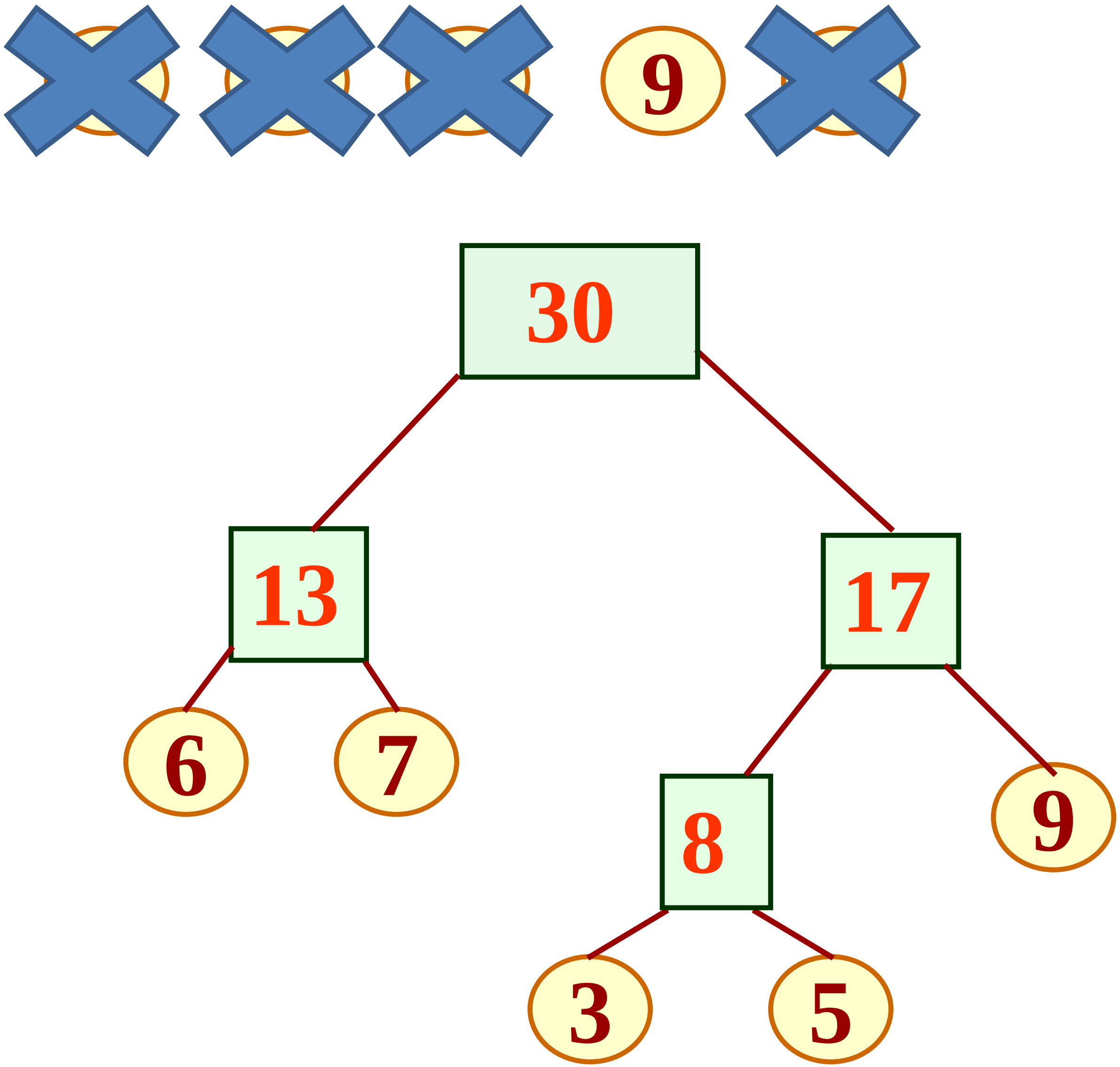
其中w_i指的是第i个外结点的权值，l_i指的是外结点到根结点的路径长度。由于w_i的值是确定的，因此为使WPL值最小，我们希望每个*li*的值都尽可能小。最小的l_i值可以为1，但我们不可能使所有叶结点的对应l_i值都为1，那样就不能构成二叉树了。因此，为使总和最小，我们希望W值大的结点的*l*值尽可能地小，而W值小的结点的*l*值可以稍大些，也即权值愈大，离根愈近。

（2）扩充二叉树中没有度为1的结点

（3）从下向上构造

由于已知叶子结点的权值，因此需要从叶子向上构造，最后得到的是根结点。

例如：已知权值 W={ 5, 6, 3, 9, 7 }构造哈夫曼树。



哈夫曼编码

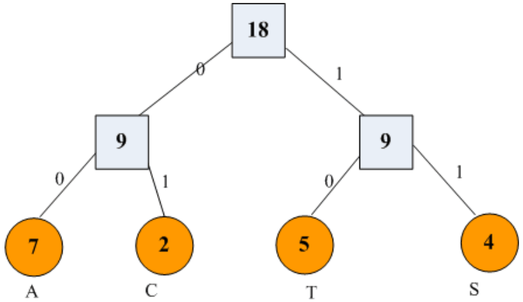
- 主要用途是实现数据压缩。设给出一段报文：

  CAST CAST SAT AT A TASA

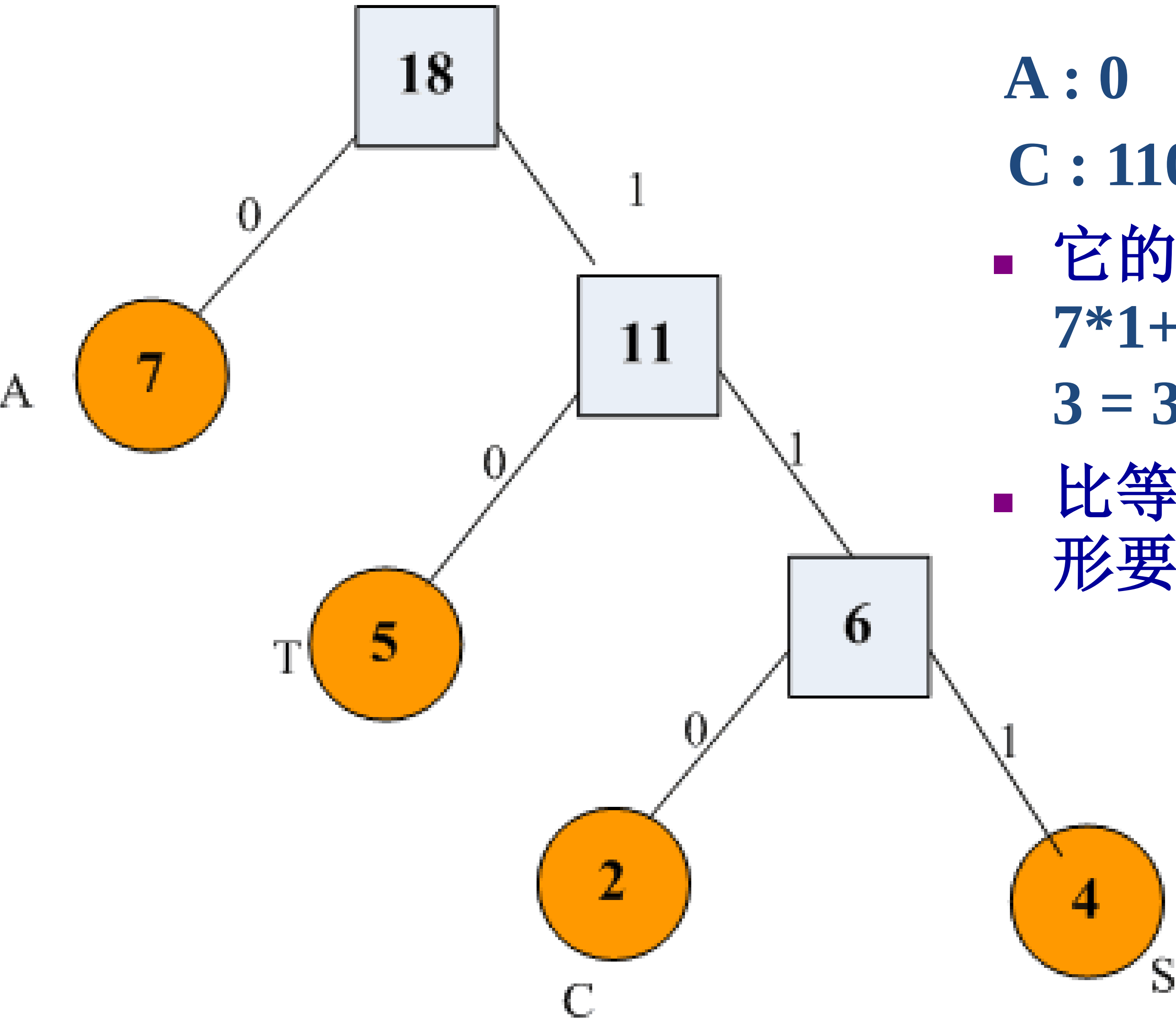
- 字符集合是 { C, A, S, T }，各个字符出现的频度（次数）是 W＝{ 2, 7, 4, 5 }。
- 若给每个字符以等长编码（2位二进制足够）

  A : 00　T : 10　C : 01　S : 11

- 则总编码长度为 (2+7+4+5) * 2 = 36。



- 能否减少总编码长度，使得发出同样报文，可以用最少的二进制代码？
- 若按各个字符出现的概率不同而给予不等长编码，可望减少总编码长度。
- 各字符出现概率为{ 2/18, 7/18, 4/18, 5/18 },化整为 { 2, 7, 4, 5 }。以它们为各叶结点上的权值, 建立Huffman树。左分支赋 0，右分支赋 1，得Huffman编码(变长编码)。

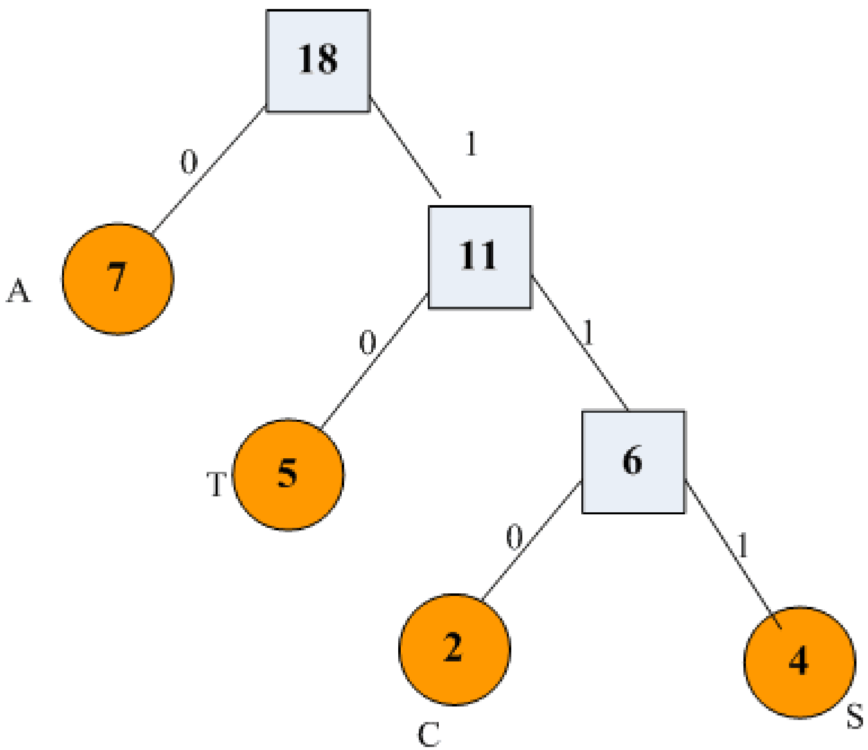


A : 0　T : 10

C : 110　S : 111

- 它的总编码长度：7*1+5*2+(2+4)*3 = 35。
- 比等长编码的情形要短。

- 总编码长度正好等于Huffman树的带权路径长度WPL。
- Huffman编码特点：任一个字符的二进制编码不是其他字符编码的前缀。
- 解码时不会混淆。



课后思考

- 在上例中，如果对这四个字符进行如下编码，A : 0 T : 1 C : 10 S : 11
- 这样总编码长度更短，这种编码方法可行吗？

课后作业

- 给定权值集合{15,3,2,6,9,16}，构造对应的哈夫曼树，并计算树的带权路径长度。