5.2 哈夫曼树与哈夫曼编码

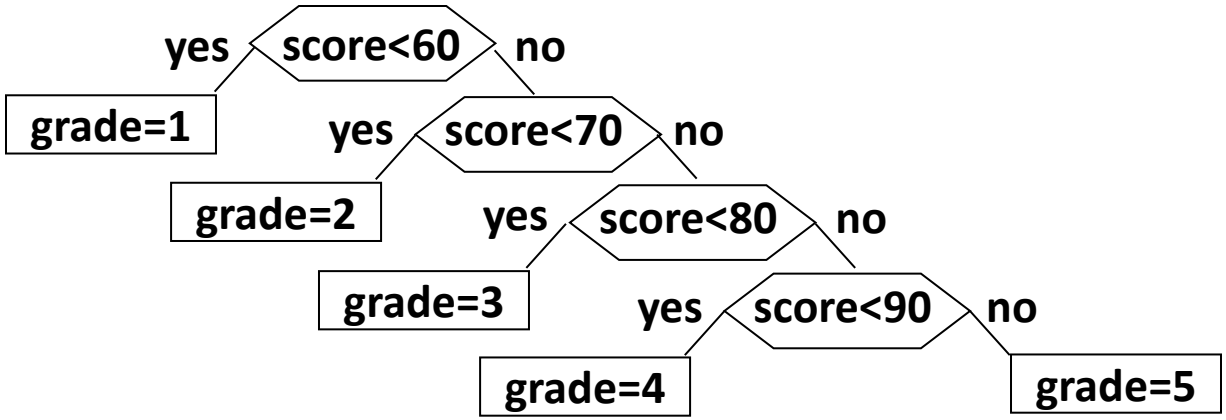
计算机现在存储，一个字占一个字节，总共占8位，最高位为0.

什么是哈夫曼树（Huffman Tree）

[例] 将百分制的考试成绩转换成五分制的成绩

```
if( score < 60 )  grade =1;
else if( score < 70 ) grade =2;
else if( score < 80 ) grade =3;
else if( score < 90 ) grade =4;
else grade =5;
```

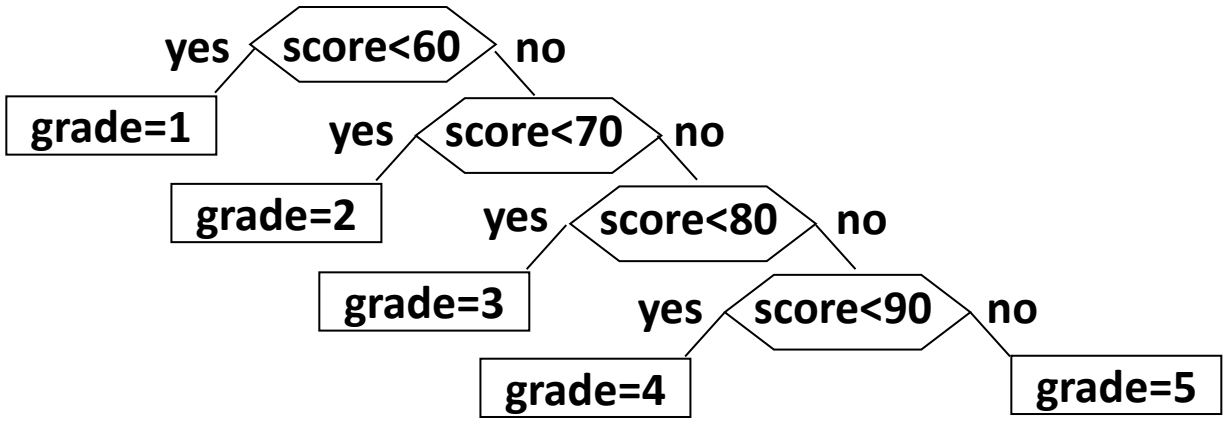
- 判定树:



- 如果考虑学生成绩的分布的概率：

分数段	0-59	60-69	70-79	80-89	90-100
比例	0.05	0.15	0.40	0.30	0.10

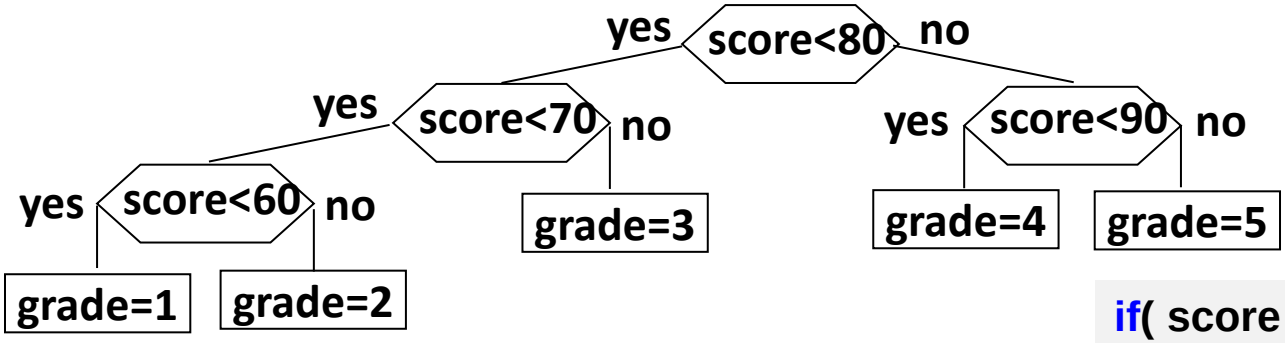
➢ 查找效率：0.05× 1+0.15 ×2+0.4× 3+0.3 ×4+0.1× 4 = 3.15



- 如果考虑学生成绩的分布的概率：

分数段	0-59	60-69	70-79	80-89	90-100
比例	0.05	0.15	0.40	0.30	0.10

- 修改判定树：




```
if( score < 80 )
{     if( score < 70 )
          if( score < 60 ) grade =1;
          else grade = 2;
      else grad=3;
}else if( score < 90 ) grade =4;
else grade =5;
```

效率: 0.05× 3+0.15 ×3+0.4× 2+0.3 ×2+0.1× 2
= 2.2

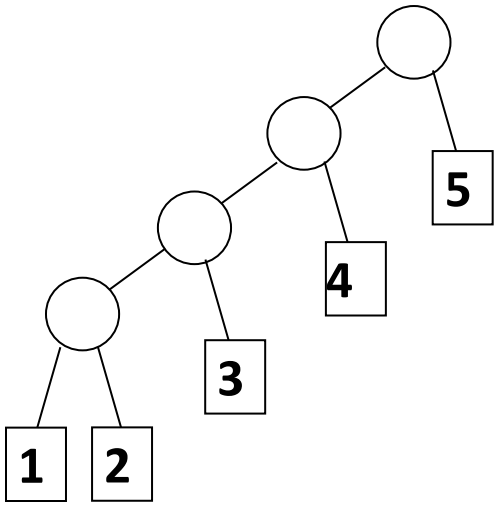
如何根据结点不同的查找频率构造更有效的搜索树？

❖ 哈夫曼树的定义

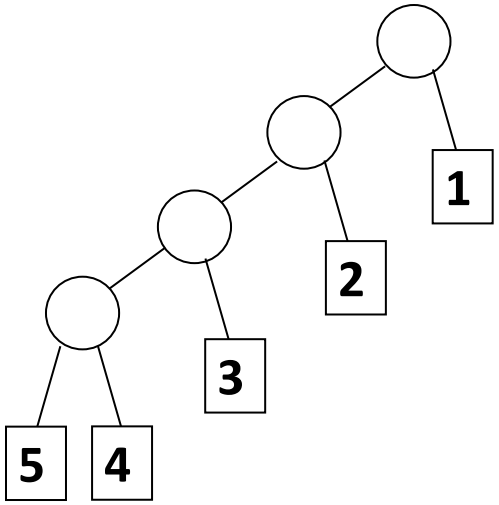
带权路径长度(WPL)：设二叉树有n个叶子结点，每个叶子结点带有权值 w_k，从根结点到每个叶子结点的长度为 l_k，则每个叶子结点的带权路径长度之和就是：$$WPL=\sum_{k=1}^{n} w_k l_k$$

最优二叉树或哈夫曼树：WPL最小的二叉树

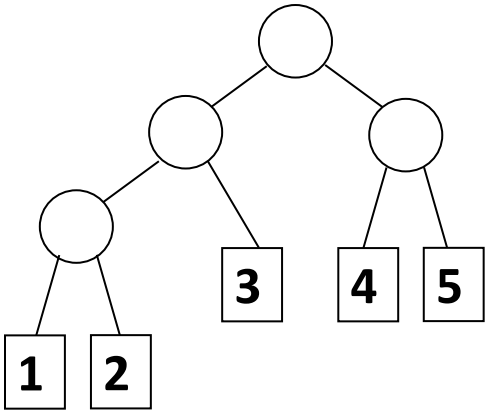
〖例〗有五个叶子结点，它们的权值为{1,2,3,4,5}，用此权值序列可以构造出形状不同的多个二叉树。



WPL = 34



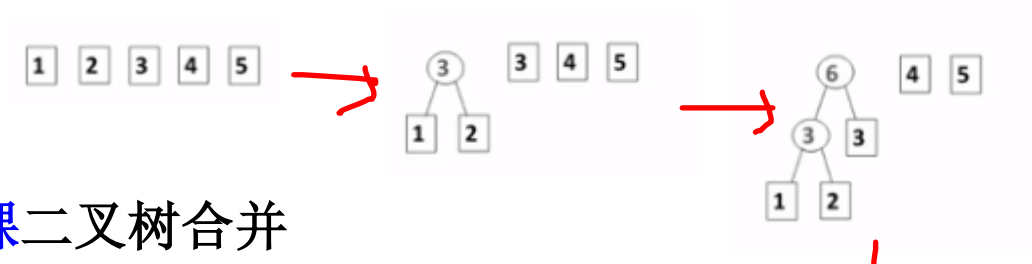
WPL = 50



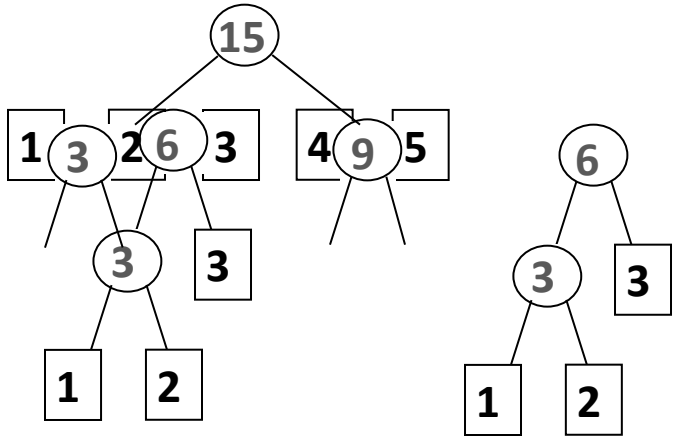
WPL = 33

WPL = 5×1 + 4×2 + 3×3 + 1×4 + 2×4 = 34

哈夫曼树的构造



☞ 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;
}
```

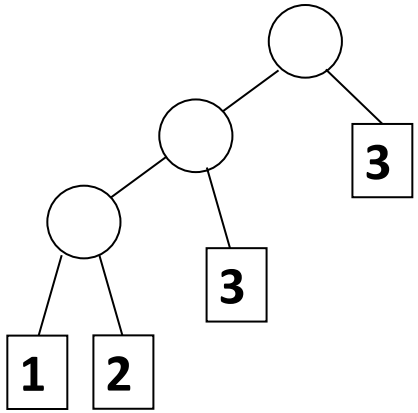
程序的难点在于如何选取两个最小的？那么就可以利用堆！。如果能把节点构造成最小堆，从里面挑两个最小的，并在一起，形成一个新的节点，再把它插入堆中

整体复杂度为O(N logN)

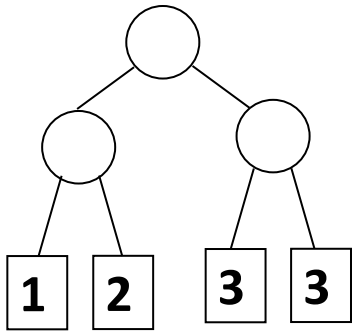
❖ 哈夫曼树的特点:

- ☞ 没有度为1的结点；
- ☞ n个叶子结点的哈夫曼树共有2n-1个结点；
- ☞ 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；
- ☞ 对同一组权值{w_1 ,w_2 , …… , 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



WPL = 18

哈夫曼编码

- 给定一段字符串，如何对字符进行编码，可以使得该字符串的编码存储空间最少？

[例] 假设有一段文本，包含58个字符，并由以下7个字符构：a，e，i，s，t，空格（sp），换行（nl）；这7个字符出现的次数不同。如何对这7个字符进行编码，使得总编码空间最少？

【分析】

（1）用等长ASCII编码：58 ×8 = 464位；

（2）用等长3位编码：58 ×3 = 174位；

（3）不等长编码：出现频率高的字符用的编码短些，出现频率低的字符则可以编码长些？

怎么进行不等长编码？

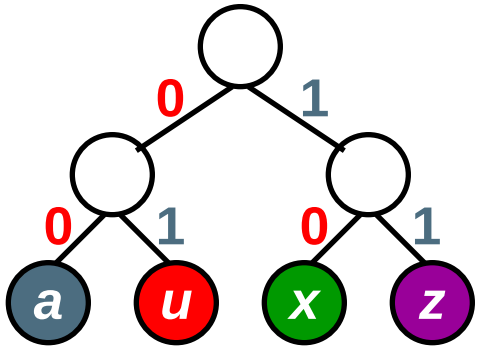
如何避免二义性？

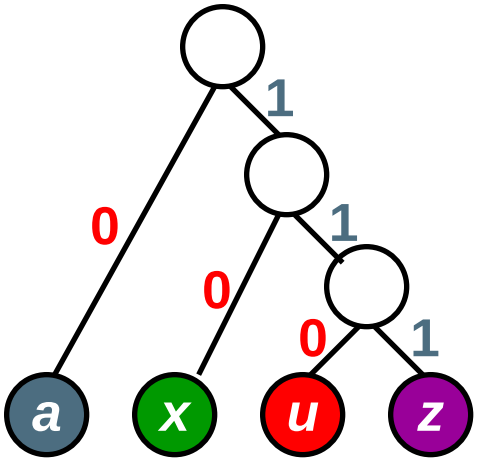
- 前缀码*prefix code*：任何字符的编码都不是另一字符编码的前缀
  - 可以无二义地解码

❖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

四个字符的频率: a:4, u:1, x:2, z:1





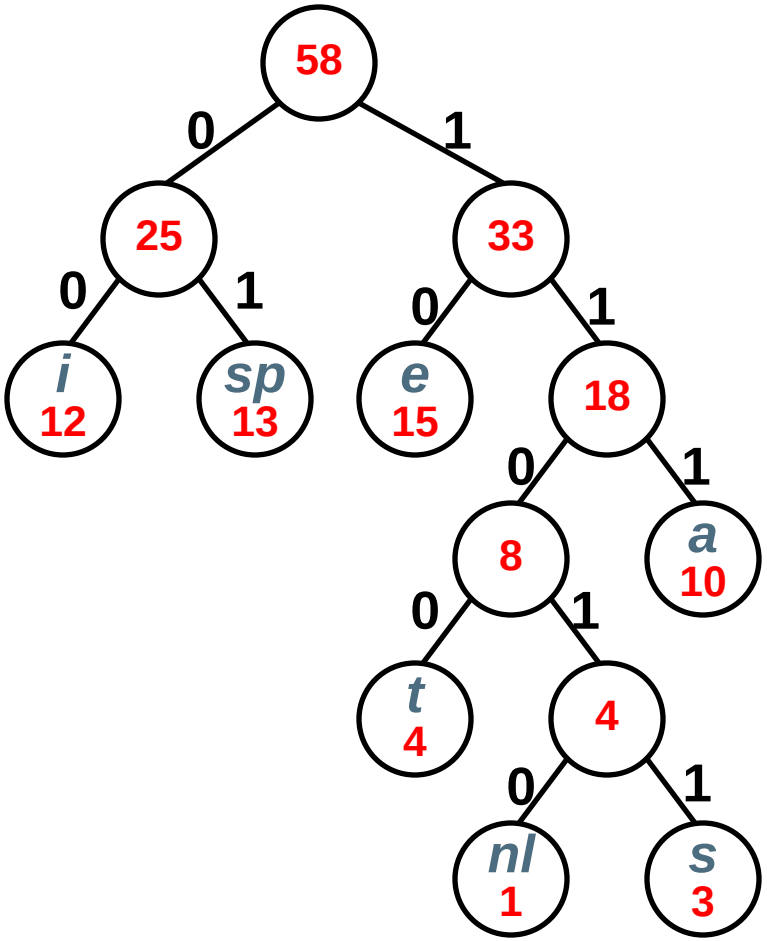
Cost **(*aaaxuaxz* → 00010110010111)**
= 1×4 + 3×1 + 2×2 + 3×1 = 14

Cost **(*aaaxuaxz* → 00000010010011)**
= 2×4 + 2×1 + 2×2 + 2×1 = 16

怎么构造一颗编码代价最小的二叉树？

〖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

Cost = 3×10 + 2×15
+ 2×12 + 5×3
+ 4×4 + 2×13
+ 5×1
= 146

浙江大学计算机科学与技术学院