

数据结构和算法

作者: 小甲鱼

让编程改变世界

Change the world by program



赫夫曼树

- 在数据膨胀、信息爆炸的今天，数据压缩的意义不言而喻。谈到数据压缩，就不能不提赫夫曼（Huffman）编码，赫夫曼编码是首个实用的压缩编码方案，即使在今天的许多知名压缩算法里，依然可以见到赫夫曼编码的影子。
- 另外，在数据通信中，用二进制给每个字符进行编码时不得不面对的一个问题是如何使电文总长最短且不产生二义性。根据字符出现频率，利用赫夫曼编码可以构造出一种不等长的二进制，使编码后的电文长度最短，且保证不产生二义性。



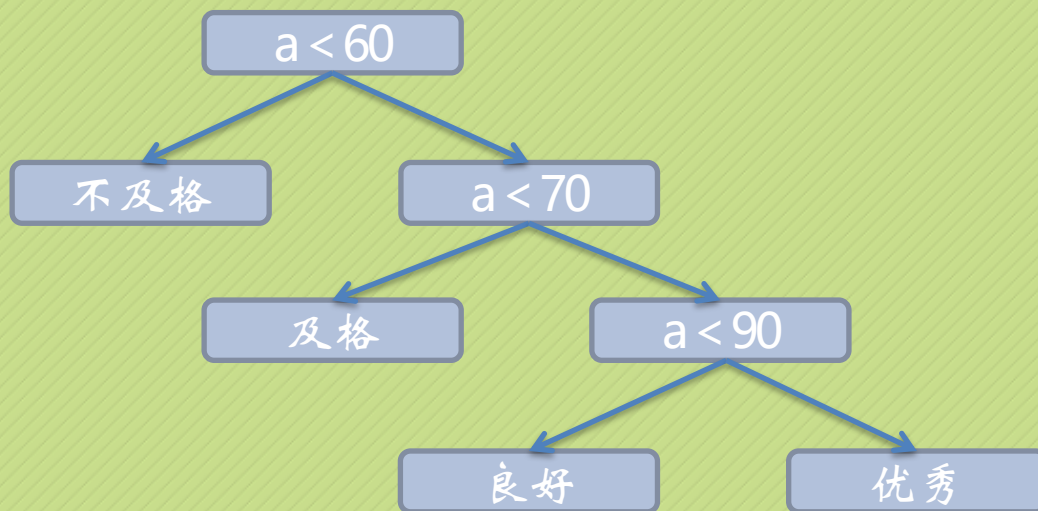
赫夫曼树

- 介绍赫夫曼编码前, 我们必须得介绍赫夫曼树。
- 什么叫做赫夫曼树呢? 我们先来看一个例子。
- 以下程序在效率上有什么问题呢?

```
if( a < 60 )  
    printf( "不及格" );  
else if( a < 70 )  
    printf( "及格" );  
else if( a < 90 )  
    printf( "良好" );  
else  
    printf( "优秀" );
```



赫夫曼树

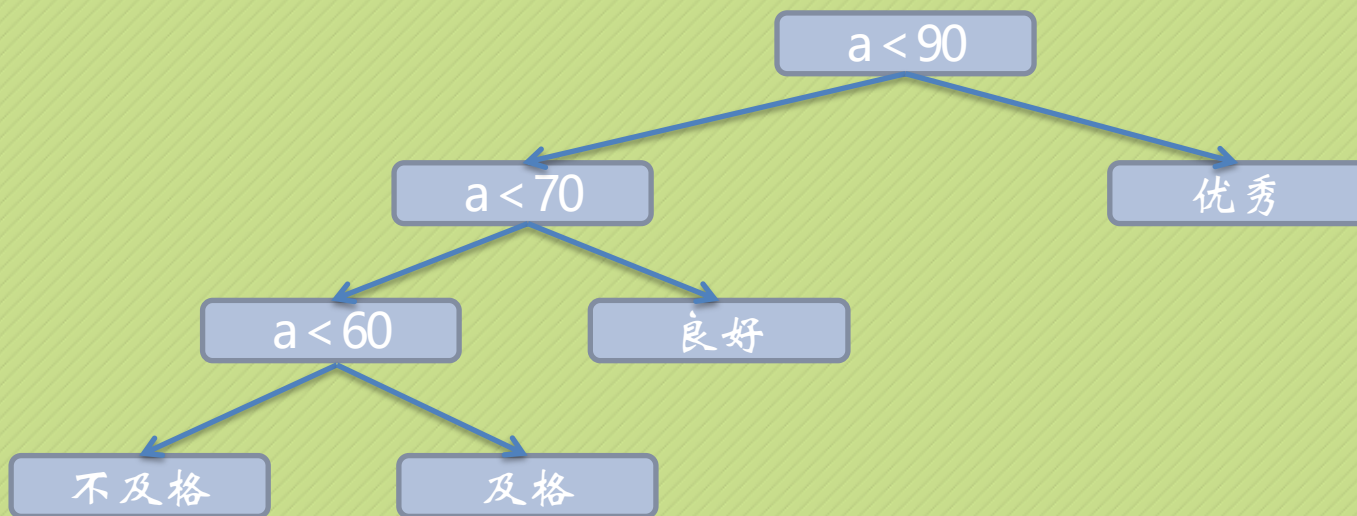


分数	0~59	60~69	70~89	90~100
所占比例	5%	15%	70%	10%



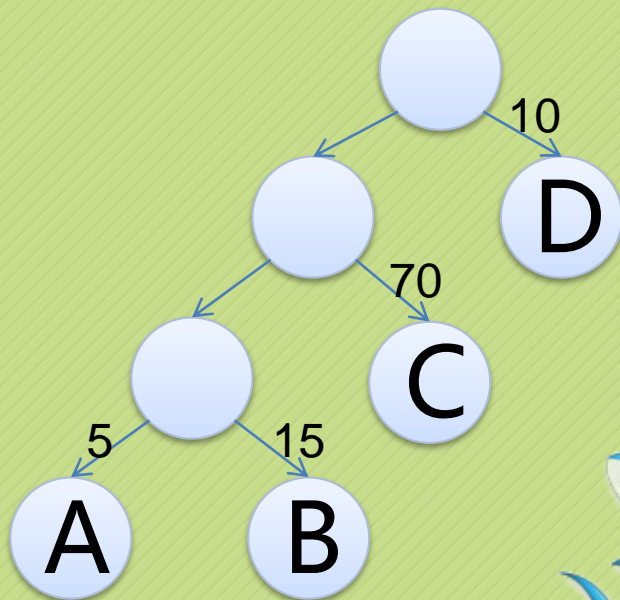
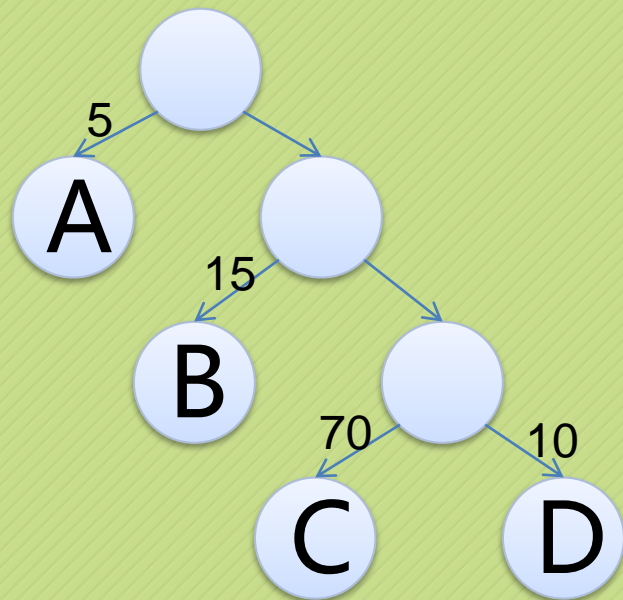
赫夫曼树

- 如果我们把判断流程改为以下，效果可能有明显的改善：



赫夫曼树定义与原理

- 我们先把这两棵二叉树简化成叶子结点带权的二叉树（注：树结点间的连线相关的数叫做权，Weight）。



赫夫曼树定义与原理

- 结点的路径长度:
 - 从根结点到该结点的路径上的连接数。
- 树的路径长度:
 - 树中每个叶子结点的路径长度之和。
- 结点带权路径长度:
 - 结点的路径长度与结点权值的乘积。
- 树的带权路径长度:
 - WPL(Weighted Path Length)是树中所有叶子结点的带权路径长度之和。



赫夫曼树定义与原理

- WPL的值越小，说明构造出来的二叉树性能越优。
- 那么现在的问题是，如何构造出最优的赫夫曼树呢？别急，赫夫曼大叔给了我们解决的办法。
- 看动画。。。。。

