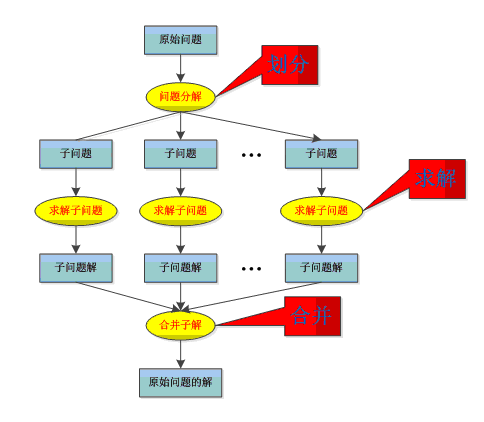
分治法，

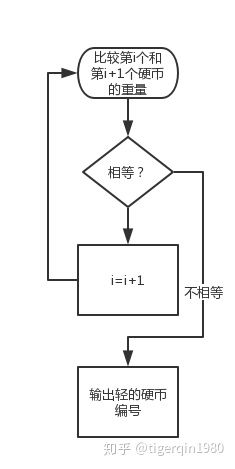
Divide-and-Conquer Algorithm

分治分治，即分而治之。分治，就是把一个复杂的问题分成两个或更多的相同或相似的子问题，再把子问题分成更小的子问题……直到最后子问题可以简单的直接求解，原问题的解即子问题的解的合并。这个技巧是很多高效算法的基础，如排序算法(快速排序，归并排序)，傅立叶变换(快速傅立叶变换)……  
直接说就是将一个难以直接解决的大问题，分割成一些规模比较小的相同的小问题，以便各个击破，分而治之。



有这样一个非常经典的问题 .  
问题：有100枚硬币，其中1枚重量与众不同，是假币，略轻一些。如果用天平秤，请问至少称几次一定能找到这枚假币。

假如我们用传统的逐枚比较法的话，显然至少需要比较99次。 流程如下:



用分治法

我们把它分为

1.将100硬币分成3份，33,33,34。

2.称量1、2份，若天平平衡，则假币必在另外34枚中。若不平衡，假币在轻的那33枚里。

3.将34枚分为11/11/12枚（或将33枚分成11\*3）。

4.称量两组11枚的硬币，若平衡，假币在12枚里（或另外的11枚）若不平衡，假币在轻的11里。

5.将11（或12）枚分成3/4/4（或4/4/4），称量4/4，方法同上。

6.将剩下的3（或4）分为1/1/1（或1/1），称量1/1，若平衡，则剩下的一枚是假币，若不平衡，轻的是假币。 若还剩4枚，出现1/1平衡，剩下2枚则称量，显然轻的是假币。

这种方法只需要5次就能解决这个问题。

3.2栗子分析

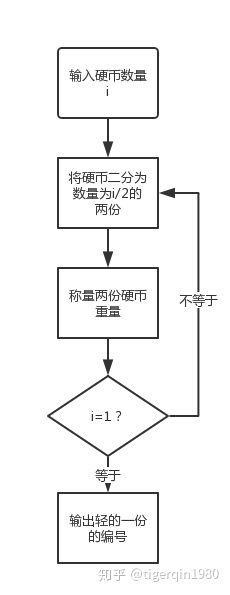
我们来看我们刚刚使用的“分治法”。

1.观察可以看到1-2，3-4，5-6步除了硬币的枚数改变了，其他的步骤完全一样。

2.观察发现这是一个子问题的分解过程，100—33-11-3，将一个大问题分解为了容易解决的小问题。

3.可以发现小问题是相互独立的。每一个33枚硬币和其他的并不相互影响。

方便起见，我们用较简单的二分法流程图来具体看一下：



可以发现这和程序设计中的递归recursio很类似，自顶而下的解决问题；

5.1流程Divide-and-Conquer！

划分问题：整个问题划分成多个无关联的子问题。  
递归求解：递归调用求解各个子问题。  
合并问题：合并子问题的解，形成原始问题的解。 我们用伪代码来具体分析~

5.2使用条件

1.该问题的规模缩小到一定的程度就可以容易地解决

2.该问题可以分解为若干个规模较小的相同问题。

3.利用该问题分解出的子问题的解可以合并为该问题的解；

4.该问题所分解出的各个子问题是相互独立的，即子问题之间不包含公共的子子问题。(非必需）