DOI:10.16707/j.cnki.fjpc.2015.04.066

分治算法的探讨及应用

何坤金
（河海大学物联网工程学院 江苏 常州 213022）

【摘 要】考虑到分治法是算法设计中的主要算法且具有广泛的应用性，本文对分治法的基本思想、基本步骤、适用条件及其应用进行探讨。最后，结合几个典型的例子，对分治法进行算法分析和算法设计，并侧重分析了分治算法的时间复杂性。

【关键词】分治法；算法设计；时间复杂性；递归

1. 分治算法的基本思想

任何一个可以用计算机求解的问题所需的计算时间都与其规模有关[1]。问题规模越小，解题所需的计算时间往往也越少，从而也越容易计算。当我们求解一个较大的问题，有时是相当困难的，由于这些问题要处理的数据相当多，或求解过程相当复杂，使得直接求解法在时间上相当长，或者根本无法直接求出。

对于这类问题，我们往往先把它分解成几个子问题，找到求出这几个子问题的解法后，再找到合适的方法，把它们组合成求整个问题的解法。如果这些子问题还较大，难以解决，可以再把它们分成几个更小的子问题，以此类推，直至可以直接求出解为止。这就是分治策略的基本思想[2]。

分治的基本思想是将一个规模为 n 的问题分解为 k 个规模较小的子问题，这些子问题互相独立且与原问题相同。找出各部分的解，然后把各部分的解组合成整个问题的解。

2. 分治算法的步骤

分治策略的基本原理是分、治、合的策略。分是将较大规模的问题分割成多个更小规模的子问题；治是对多个子问题进行求解；合是将求出的小规模问题的解合并为一个更大规模的问题[3]。分治法解题的一般步骤：

⑴ 分解，将问题划分为一些子问题，子问题的形式与原问题一样，只是规模更小；

⑵ 求解，递归地求解出子问题。当子问题划分得足够小时，用较简单的方法直接求解；

⑶ 合并，将子问题的解逐层合并组成原问题的解

分治法的一般的算法框架如下：

```
Divide-and-Conquer(n)          //n 为问题规模
{
    1. if n==n0                //n0 为可解子问题的规模
    2. 解子问题；
    3. return(子问题的解)；
    4. for i←1 to k            //分解为较小的 K 个子问题 P1,P2,…,Pk
    5.  yi ← Divide-and-Conquer(|Pi|)      //递归解决 Pi, |Pi|为 Pi 的规模
    6.  T ← MERGE(y1,y2,...,yk)            //合并子问题
    7.  return(T)
}
```

3. 分治算法的适用条件

分治法是建基于多项分支递归的一种很重要的算法范式，它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，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。另一方面，设计分治法算法的能力需要一定时间去掌握，为了使递归能够推行，很多时候需要用一个较为概括或复杂的问题去取代原有问题。采用分治法解决的问题一般具有几个特征[4]：

⑴ 该问题的规模缩小到一定的程度就可以容易地解决，或直接解决；

⑵ 该问题可以分解为若干个规模较小的相同或相似子问题；

⑶ 利用该问题分解出的子问题的解在一定的条件下可以合并为该问题的解；

⑷ 该问题分解出的各个子问题是相互独立的，且各个子问题间不包括公共子问题。

第一条特征是绝大多数问题都可以满足的，因为问题的计算复杂性一般是随着问题规模的增加而增加。

第二条特征是应用分治法的前提它也是大多数问题可以满足的，此特征反映了递归思想的应用。

第三条特征是关键，能否利用分治法完全取决于问题是否具有第三条特征，如果具备了第一条和第二条特征，而不具备第三条特征，则可以考虑用贪心法或动态规划法。

第四条特征涉及到分治法的效率，如果各子问题是不独立的则分治法要做许多不必要的工作，重复地解公共的子问题，此时虽然可用分治法，但一般用动态规划法较好。

4. 分治法的求解与分析

分治算法通常是递归算法，算法时间复杂度的分析需要求解递归方程。在分治算法中最常见的递归方程有两类[5]：

$$T(n)=\sum_{i=1}^{k} a_i T(n-i)+f(n) \quad \text{(第 1 类)}$$

$$T(n)=a\,T(n/b)+d(n) \quad \text{(第 2 类)}$$

第 1 类方程所解问题的特征是：归约后的子问题规模比原问题呈现常数量级的减少，例如 Hanoi 塔的分治算法，该问题要求将 n 个盘子的移动归约为 2 个 n−1 个盘子移动的子问题，子问题规模只比原问题少了 1 个盘子，其递归方程为：

$$\begin{cases} T(n)=2T(n-1)+1 \\ T(1)=1 \end{cases}$$

这类方程可通过迭代、换元、递归树等方法求解，最后求得所需移动盘字数为 $T(n)=2^n-1$.

第 2 类方程是在均衡划分的情况下，a 是归约后的子问题个数，b 是子问题规模减少的倍数，$d(n)$表示归约过程和综合求解过程的总工作量，可通过迭代法、递归树和主定理等求解，有

几种常见形式：

当 $d(n)$为常数时，如果 $a=1$，于是 $T(n) = O(\log n)$ 如果 $a\neq 1$，$n^{\log_b a}$ 不是常数，$T(n)=O(n^{\log_b a})$，于是得到：

$$T(n) = \begin{cases} O(n^{\log_b a}) & a \neq 1 \\ O(\log n) & a = 1 \end{cases}$$

当 $d(n)=cn$ 时，如果 $a>b$，方程的解为 $O(n^{\log_b a})$ 如果 $a=b$，方程的解为 $O(n\log n)$ 如果 $a<b$，方程的解为 $O(f(n)) = O(n)$，于是得到：

$$T(n) = \begin{cases} O(n) & a < b \\ O(n\log n) & a = b \\ O(n^{\log_b a}) & a > b \end{cases}$$

利用第2 类方程，可直接得到二分检索算法的时间复杂度为 $O(\log n)$，二分归并排序算法的时间复杂度为 $O(n\log n)$。

5. 分治算法应用实例

分治法常用于在计算机科学中，分治法是一种很重要的算法。字面上的解释是“分而治之”，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，原问题的解即为子问题的解的合并。分治法包括数值类和非数值类两大类。数值实例常有：二分搜索、大整数乘法、快速排序、合并排序等[6]；非数值实例常有：棋盘覆盖、汉诺塔等。以下分别就数值问题和非数值问题进行分析。

数值实例：在 n 个数字中，同时求出 n 个数字中的最大和最小两个数，要求比较的次数尽可能的少。

分治法可以用较少的比较次数解决该问题：首先，将数据等分为两组，目的是分别选取其中每组中的最大值和最小值；其次，递归分解直到每组元素的个数小于等于 2，可简单找到最大、最小值；最后，回溯时合并子问题的解，在两个子问题的解中取最大和最小值，即合并为当前问题的解。

根据分治法的求解方程：$T(n)= 2T(n/2) + 2$，可得，$T(n) = 3n/2-2$

非数值实例：残缺棋盘是一个有 $2k*2k$ 个方格的棋盘，其中恰有一个方格残缺，用多个三格板，如图 1(a)所示，进行覆盖残缺棋盘，如图 1 所示。覆盖要求：两个三格板不能重叠；三格板不能覆盖残缺方格，但必须覆盖其他所有方格。

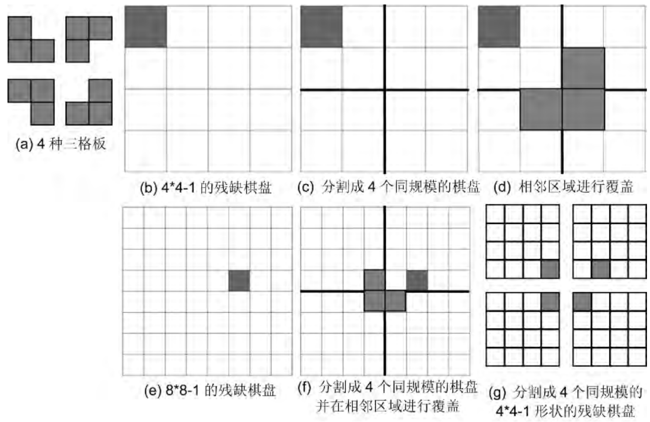

图 1　三格板和 2^k*2^k-1 的残缺棋盘

根据分治法的求解方程：$T(n)= 4*T(n/4) + d(n)$，$d(n) = c$，可得，$T(n) = O(n)$

6. 结束语

本文对分治法从基本思想、步骤、适用条件及应用进行探讨，侧重于分析了该方法的时间复杂性，将该方法应用于数值计算和非数值计算两大类。并结合实例，采用分治法进行算法设计与分析，结果表明，分治算法具有高效率，方便用户使用的特点。

参考文献：

[1] Thomas H.Cormen 等，算法导论，机械工业出版社出版，2013
[2] 马燕，数据结构中基于分治策略的排序算法探讨，延安大学学报(自然科学版)，2006
[3] 春燕，基于分治策略的快速排序算法探讨，西藏大学学报，2003
[4] 吕国英，算法设计与分析，清华大学出版社，2006
[5] 屈婉玲等，算法设计与分析，清华大学出版社，2011
[6] 于志奇等，基于分治策略的排序方法的比较研究，太原师范学院学报(自然科学版)，2008

(上接第 59 页)

用班比普通班的参与热情高。

5 结论

从对 2011 级和 2012 级的横向对比来看，对《Windows 程序设计》课程进行的应用性改造探索取得了较好的效果。实践证明，其有助于提高学生的学习兴趣和学习主动性；有助于培养学生的团结协作精神；有助于实现理论与实践相结合，提高学生的实践应用能力；从而提高学生的综合素质，进而增强学生的就业竞争力。

参考文献：

[1] 兰红,李淑芝.基于“以学生为中心”的计算机语言类课程改革探索[J].中国电力教育.2010,(10):49- 51
[2] 虞芬,邹睿娟.以学生为中心,培养学习能力——《Windows 程序设计(C#)》课程的教学改革与实践[J]. 九江职业技术学院学报.2011,(02):46- 48.
[3] 杨程,陈念年,李郁峰.游戏开发驱动的 Windows 程序设计课程教改探析[J].教学研究.2012,(02):92- 95.
[4] 刘智,张金荣,王森.深入浅出讲解“Windows 程序设计—Visual C++”课程[J].计算机时代.2012,(09):57- 59.

作者简介：

邱宁(1978-)，男，硕士，副教授，主要研究方向：计算机应用、数据挖掘。