HZ BOOKS
华章教育

面向CS2013计算机专业规划教材

算法设计与分析

黄宇 编著

Algorithm Design and Analysis

机械工业出版社
China Machine Press

面向CS2013计算机专业规划教材

算法设计与分析

黄宇 编著

Algorithm Design and Analysis

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

算法设计与分析 / 黄宇编著 . —北京：机械工业出版社，2017.3
（面向 CS2013 计算机专业规划教材）

ISBN 978-7-111-57297-8

I. 算… II. 黄… III. ①电子计算机 – 算法设计 – 高等学校 – 教材 ②电子计算机 – 算法分析 – 高等学校 – 教材 IV. TP301.6

中国版本图书馆 CIP 数据核字（2017）第 142226 号

本书讲授算法设计与分析的基础知识。首先介绍计算模型的基本概念；其次围绕遍历、分治、贪心、动态规划这四种经典算法设计策略，讲解排序、选择、查找、图遍历、最小生成树、最短路径等经典算法问题；最后介绍计算复杂性的基础知识。

本书主要面向计算机专业本科生，以及其他需要学习计算机科学基础知识与了解计算机程序设计背后原理的读者。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）

责任编辑：佘　洁　　责任校对：李秋荣

印　　刷：中国电影出版社印刷厂　　版　　次：2017 年 8 月第 1 版第 1 次印刷

开　　本：185mm × 260mm　1/16　　印　　张：13.5

书　　号：ISBN 978-7-111-57297-8　　定　　价：49.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991　88361066　　投稿热线：（010）88379604

购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

献给我的女儿允浛，她与这本书一起孕育和成长。

前　言

算法是计算的灵魂（spirit of computing），而算法设计与分析的基础知识是计算机科学的基石。算法设计与分析的知识内容很丰富，可以从不同视角进行组织与阐述。一种视角是关注经典的算法问题，如排序、选择、查找、图遍历等；另一种视角是关注经典的算法设计策略，包括分治、贪心、动态规划等。本书的组织兼顾问题与策略两种视角。首先按照经典的算法设计策略，将书中的主体内容分为遍历、分治、贪心、动态规划4个部分。其次在每个部分之内，又围绕经典的算法问题来阐述该部分所着重讨论的策略。

本书集中讨论抽象的即与机器、实现语言无关的算法设计与分析。为此在主体内容之前，我们首先讲解计算模型的基础知识，它是后续抽象讨论算法设计与分析的基础。另外，在本书的最后，我们介绍计算复杂性的基础知识，试图让读者在了解了各类算法问题、学习了各种算法设计和分析技术之后，对算法问题的难度有一个总体性的认识。此外，一些对于算法设计与分析较为关键的数学知识将在附录中列出。本书的内容是作者在过去多年授课的过程中逐渐积淀而成的，所以它不是对算法设计与分析知识的一个百科全书式的覆盖，而是对一些重点内容的更专注的讨论。

本书内容和组织方式的设计针对一个学期的授课展开。在内容方面，本书可以分为前后两个部分。前一部分主要围绕元素的序关系展开，讨论排序、选择、查找这3个经典的算法问题。而这3个问题的求解同时又是分治策略的典型应用。后一部分主要围绕“图”这一数学结构展开，讲授图遍历、最小生成树、最短路径等经典图算法。同时，这些图算法背后的一个核心问题是图上特定结构——最小生成树、最短路径——的优化。围绕图优化，我们展示了贪心策略与动态规划策略的典型应用。

在授课形式方面，我们将课程分为主课与辅课这两种形式。主课主要围绕典型的算法问题、经典的算法展开。而辅课则围绕算法策略来展开，在讲述若干经典问题、经典算法之后，从策略的视角回顾最近阶段的经典算法，同时补充新的素材对相应的策略进行进一步的展示。除知识讲授之外，实践也是“算法设计与分析”课程的重要组成部分。算法设计与分析课程的实践分为两类：一类是传统的习题，即

紧扣书本知识的习题，如一些简单定理的证明、紧扣算法细节的一些问题等；另一类是应用题，它需要读者对一个有一定现实意义的问题进行建模，并用书中的算法知识来求解。本书的应用题大都可以用于算法编程实现的训练。在实际授课中，我们挑选了部分应用题作为编程实现题，并基于开源的 OnlineJudge 平台进行自动评测，取得了良好的效果。

本书的素材主要源自于南京大学计算机系本科生“算法设计与分析”课程的授课内容。其中一部分素材来源于共同授课的其他老师，包括前期负责讲授主课并指导辅课教学的陈道蓄老师，以及后期共同分班讲授这门课程的钱柱中老师。还有一部分素材来自于经典的算法教科书和国外大学授课教师在其课程网站上发布的课程材料。另外，还要感谢“算法设计与分析”课程的两位助教魏恒峰和杨怡玲，他们对大量的课程资料进行了整理与提炼。最后要感谢上过这门课的学生，他们创造性的提问与解题时所犯的错误都为本书提供了宝贵的素材。

教学建议

说明：南京大学计算机系“算法设计与分析”课程的讲授采用三种不同形式，即主课、辅课（tutorial）和习题课。

- 主课围绕各个主要知识点进行专题讲授。下面列出主课的授课计划，包括每次课的主题以及所对应的书本中的章节。
- 辅课的主要内容是对前一阶段主课知识的多角度解读，以及重点内容的强化。辅课的授课往往以围绕经典例题的讨论为主，以知识点的阐述为辅。
- 习题课的讲授主要包括书上习题的讲解，以及上机评测问题的讲解。习题课的讲授可以根据具体的教学、上机、考试等情况进行相应的安排。

教学章节	教学要求	课时
主课一　准备知识 （第 1 章）	计算模型的基础知识 抽象算法设计与分析的基本概念	2
主课二　数学基础 （第 2 章）	函数渐近增长率的基本概念 简单蛮力算法的逐步改进	2
	递归方程的基本求解技术 基于 Master 定理的分治递归求解	2
辅课一	从算法设计与分析的角度重新审视数学的概念	2
主课三　排序 （第 3、6、7 章）	线性表的遍历，从蛮力排序到快速排序	2
	堆结构的维护 堆结构的应用：堆排序与优先队列	2
	合并排序 基于比较的排序的下界	2
辅课二	排序：从简单遍历到分而治之	2
主课四　选择 （第 8 章）	选择问题的简单特例 期望线性时间选择 最坏情况线性时间选择	2
主课五　查找 （第 9、10 章）	折半查找 平衡二叉树的定义及平衡性分析	2
	动态等价关系下的查找 并查集的设计与分析	2

（续）

教学章节	教学要求	课时
辅课三	分治策略中的平衡性控制技术	2
主课六　图遍历（第 4、5 章）	DFS、BFS 基本算法框架 DFS 框架深入分析	2
	有向图中的 DFS：拓扑排序、任务调度、强连通片识别	2
	无向图中的 DFS：寻找割点、寻找桥	2
辅课四	BFS 框架深入分析、图遍历的典型应用、DFS 和 BFS 各自特色比较	2
主课七　图优化（第 10～14 章）	最小生成树：Prim 算法、Kruskal 算法	2
	最短路径：给定源点最短路径、所有点对间最短路径	2
	从图优化到一般优化问题求解：贪心策略、动态规划策略的典型应用	4
辅课五	图的贪心遍历框架、经典图优化问题的各种变体	2
主课八　计算复杂性理论初步（第 15～16 章）	P 问题、NP 问题基本概念 问题间归约基本概念	2
	NP 完全问题基本概念 基本的 NP 完全性证明技术	2
辅课六	算法设计与分析——过去与未来：抽象算法设计与分析回顾、难问题求解技术简介（近似算法）、算法设计与分析前沿简介（随机算法、在线算法、并行分布式算法等）	2

目　录

第四部分 贪心策略

第五部分 动态规划

第一部分
PART ONE

计算模型

这一部分的内容是学习全书后续知识的基础。首先我们引入计算模型的概念。有了计算模型这台“抽象的计算机器”，我们可以在其上讨论抽象的——与机器、实现语言无关的——算法设计与分析。在讨论算法设计时，我们着重讨论算法正确性的严格证明，其关键挑战在于算法的输入可能有无穷多个。应对无穷多输入的主要手段是数学归纳法。在讨论算法分析时，我们根据抽象算法的特点，引入“关键操作个数”这一与机器、实现语言无关的指标，算法分析本质变成对关键操作执行的计数。算法分析的基本内容包括最坏情况时间复杂度分析和平均情况时间复杂度分析。

在给出计算模型与抽象算法设计与分析的基本概念后，我们进一步讨论与算法设计与分析相关的数学知识，着重强调从刻画计算机运作的角度，从算法设计与分析的角度，重新思考已经学习过的数学知识。我们首先建立常用的数学函数与常见的算法操作之间的关联。其次引入函数渐近增长率的概念，这是描述算法性能的基本手段。最后讨论一类特殊形式的递归方程的求解，这类递归是分析分治算法(参见第三部分的讨论)的有效工具。

CHAPTER 1

第 1 章

抽象的算法设计与分析

为了讨论与机器、实现语言无关的抽象算法设计与分析，我们必须有一台“抽象的机器”——一个计算模型——作为我们实现算法的载体。为此我们首先讨论计算模型的基本概念，然后引入 RAM 模型。在此基础上，我们进一步介绍抽象算法设计和分析的基本概念。

1.1 RAM 模型的引入

1.1.1 计算的基本概念

计算技术已经渗透到我们日常生活的方方面面，显著地改变了我们的生活。计算技术的广泛应用与巨大成功让我们不禁思考：“为什么计算机似乎无所不能。”例如，我们平时的工作、娱乐、交流都得益于计算机的支持。但是经过稍加深入的思考，我们则可以提出一个更加深入的问题：“为什么有些事情对人来说不难，而对计算机来说却非常困难；反之亦然，为什么有些事情对计算机来说很容易，而对人来说却非常困难。”例如，我们很容易理解一个人所讲的话，这对计算机来说却十分困难。再例如，从海量的歌曲信息中查找某首特定的歌曲对于计算机来说很容易，而对于人来说却非常困难；但是歌曲是由人来谱曲的，而对于计算机来说，哪怕“创作”很短的一段曲调都是非常困难的事情。

上述问题与计算技术的本质特征紧密关联。计算的首要使能技术(enabling technology)是 0/1 编码。所有需要计算机处理的信息均首先进行 0/1 比特“编码”，然后以 0/1 比特的形式进行处理，最终再从 0/1 比特“解码”为用户易于理解的形式输出。能够进行0/1 编解码的信息均可以用计算机进行处理，而计算机几乎无所不能的原因在于，很多常见的信息均可以有效地进行 0/1 编解码。例如，我们要拍一张照片发送给远在地球另一端的朋友。首先，数码相机将我们的物理影像信息 0/1 编码成特定格式的图片。这些包含影像信息的 0/1 比特则可以在计算设备上存储、

复制，还可以通过网络进行传输。同时，收到这些 0/1 比特的人也可以用特定的软件将它解码并在显示设备上显示。

0/1 编解码的“魔术”使得计算机只需要通过种类有限的、较为简单的操作就可以对信息进行处理。计算机能完成复杂的任务，其关键不在于计算机能做各种复杂的操作，而是在于计算机能组合简单的操作来完成复杂的任务。因而我们总结计算的关键特征在于：基于有限种操作的灵活组合来完成复杂的计算任务。这可以用大家熟悉的积木作一个类比。在一套积木中，积木块的种类是有限的，但是积木块的组合方式却是任意多的(假设每一种积木块都有充分多个)。搭积木的意义在于用有限种类的积木块作出各种(创造性)组合。

计算的这一本质特征也解释了计算机的擅长与不擅长：计算机并无创造能力，它只不过是根据预先指定的操作序列在依次执行。计算机的优势在于它执行操作的速度很快，不知疲倦，并且在高强度的工作负载下很少犯错(相对人而言)。这就解释了对于海量信息的简单处理(如检索海量歌曲信息)，计算机远比人类要擅长。但是对于自然语言的理解、艺术创作等“创造性”的事情，计算机就难以做到，或者说计算机必须依赖人将这些创造性的事情先转换成简单操作的组合，计算机才能按照人的安排机械地完成。

有了计算的基本概念，我们可以给算法下一个宏观的定义：算法就是一组计算机操作的序列，遵循算法的指示，计算机对任意合法输入执行一系列操作，并给出正确的输出。

1.1.2　计算模型的基本概念

通过上面对计算的基本概念的讨论，我们知道算法首先依赖于一台机器。机器能执行种类有限的操作(我们称之为指令)，但每个指令的执行次数可以是任意多的。算法描述了一种指令的序列，它能对任意合法的输入完成计算，给出正确的输出。

基于上述算法与机器二者的关系，我们又可以引出一个更进一步的问题。具体而言，假设大家有计算机程序设计的基本知识与动手经验。回顾我们的编程活动，大家会发现：我们解决计算问题时的思考和处理是机器无关的、实现语言无关的。一方面，假设你会在普通的 PC 上对一组元素进行排序，现在把计算设备换成一部手机。如果你预先了解了手机编程的基础知识，那么你对于排序的思考和理解，可以帮助你很容易在新的手机平台上完成相应的排序程序。另一方面，假设你用 C 语言实现了一个排序算法，现在我们把实现语言换成 Java。如果你充分了解 Java 语言的知识，那么你对于排序的思考和理解，可以帮助你很容易用新的 Java 语言实现同样的排序算法。

我们可以从两个不同的角度来思考上述现象背后的原因。

- 自顶向下地看：我们掌握的是一种抽象的原则，它与具体的机器和语言是无关的。换成算法的语言来说，我们是在一台抽象的机器上完成了算法的设计与分析。当熟悉机器的细节和语言的细节时，我们可以很容易地将抽象的算法“实例化”(instantiate)成一个具体机器、具体语言的算法实现(程序)。
- 自底向上地看：虽然各种计算设备千差万别，但是我们可以在汇编语言的层次上思考它们的共通之处。不考虑体系结构的不同，不同机器的主要差别是寄存器数目和字长的不同。但是这些参数虽然不同，却都在一个接近的范围内。我们可以将每一种计算设备都看成这样一种机器，即提供一组基本操作，完成数学、逻辑计算和存储器访问。不同机器上的一个操作在另外一个机器上总可以用常数个操作来模拟。这样，不同具体机器的操作代价至多相差常数倍，它们在本质上是一样的。

上面的论述中提到的“抽象的机器”就是计算模型(model of computation)，它是抽象的算法设计与分析的基础。计算模型可以执行数学运算、逻辑运算、存储器访问等基本操作。它仅存在于我们的思维之中，但是它包含了计算设备进行计算的核心运作机理。

提到计算模型，最有名的是图灵机[⊖]，它是英国数学家阿兰·图灵于1936年提出的一种抽象计算模型。但是图灵机的描述能力过于强大而不便于使用。对于讲授算法设计与分析的基础知识而言，更合适的计算模型是简单易用的RAM模型。

1.1.3　RAM模型

我们使用RAM（Random Access Machine）模型作为我们讲解抽象的算法设计与分析的计算模型。根据上面的讨论，我们可以把RAM模型理解成一台抽象的逻辑上的计算机，其基本构成如图1.1所示。RAM模型包含一个只读的输入纸带和一个只写的输出纸带。输入纸带由一个个空位组成，每个空位存储一个整数。每当一个空位的值被读入后，读写头向右移动一个空位。输出纸带同样由多个空位组成，它开始全部为空。当写指令执行的时候，RAM模型首先在当前的空位写下数值，然后读写头向右移动一个空位。一旦某个值被写到纸带上，它不能够再被更改。存储空间由一系列的寄存器 r_0，r_1，…，r_i，…构成，每个寄存器可以存储一个任意大

⊖ http://en.wikipedia.org/wiki/Turing_machine。

小的整数值。我们假设问题的规模不是特别大，能够在存储器中处理。同时我们假设计算所涉及的整数不是特别大，可以在机器的字长内表示。

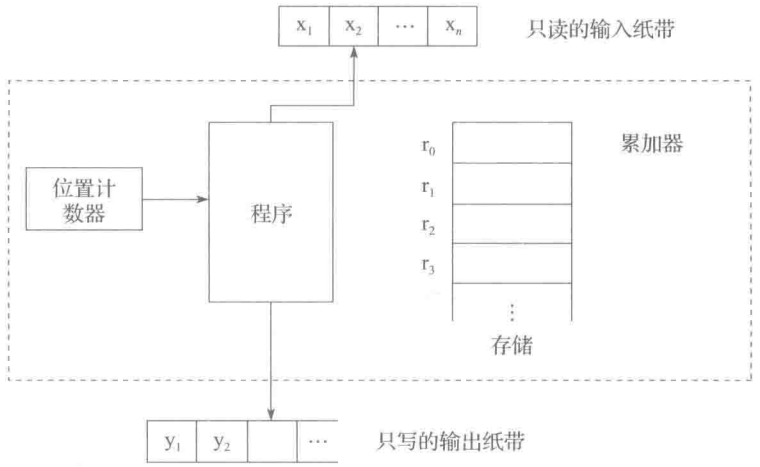

图 1.1　RAM 模型的基本构成

RAM 模型执行的程序不是存放在存储器内的，因而程序不能修改其自身。程序只是一组指令的序列，我们对 RAM 模型所执行指令的细节并不关心，只是将它们粗略地分为 3 类。

- 简单操作：RAM 模型可以执行常见的简单操作（simple operation）。我们不详细讨论每个简单操作的细节，只是假设对于现实计算机上常见的、合理的简单操作 RAM 模型都能完成。例如，在排序时，RAM 模型能完成两个待排元素的比较；在串匹配时，RAM 模型能完成两个字符的比较；在图遍历时，能对一个节点染色等。
- 复杂操作：复杂操作主要包括“循环”和“子程序调用”，在分析这些复杂操作时，需要把它们分解成简单操作的组合。
- 存储访问：存储单元的读/写是简单操作。假设 RAM 模型有充分大的存储空间，并且每次存储访问操作需要单位时间。

注意，从理论上讲我们可以将任意操作定义为简单操作。但是如果将一个实际很复杂的操作定义为简单操作，则得到的计算模型就会明显地偏离现实，未能对现实问题进行准确、有效的抽象，因而其本身就没有存在的意义。所以我们需要合理地使用定义 RAM 模型的简单操作的能力。在本书的讨论中，我们并不会显式地列出 RAM 模型所有的简单操作，而是以现实计算机的指令集为指导，针对不同的算法问题，约定合理的简单操作集。

除了约定合理的简单操作集之外，即使针对一个简单操作，当操作数的值