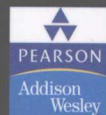




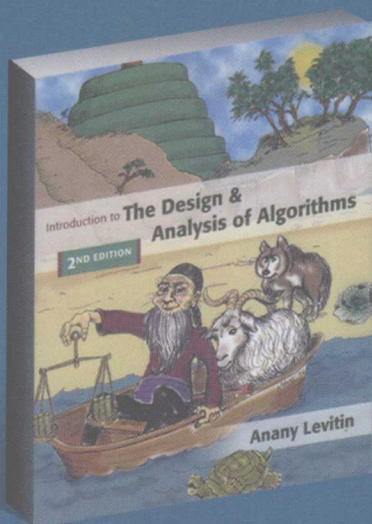
国外经典教材·计算机科学与技术



# 算法设计与分析基础

(第2版 影印版)

Introduction to the Design and  
Analysis of Algorithms  
Second Edition



- 经典原著，原味赏析
- 角度新颖，独树一帜
- 注重实践，学练结合
- 教辅齐全，获取便捷

(美) Anany Levitin 著



清华大学出版社

国外经典教材 • 计算机科学与技术

# 算法设计与分析基础

(第2版 影印版)

(美) Anany Levitin 著

江苏工业学院图书馆  
藏书章

清华大学出版社

北 京

English reprint edition copyright © 2007 by PEARSON EDUCATION ASIA LIMITED and TSINGHUA UNIVERSITY PRESS.

Original English language title from Proprietor's edition of the Work.

Original English language title: Introduction to the Design and Analysis of Algorithms, 2nd Edition by Anany Levitin, Copyright © 2007

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Pearson Education, Inc.

This edition is authorized for sale and distribution only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong, Macao SAR and Taiwan).

本书影印版由 Pearson Education, Inc. 授权给清华大学出版社出版发行。

**For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macao SAR).**

仅限于中华人民共和国境内(不包括中国香港、澳门特别行政区和中国台湾地区)销售发行。

北京市版权局著作权合同登记号 图字: 01-2007-5026

版权所有, 侵权必究。侵权举报电话: 010-62782989 13501256678 13801310933

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签, 无标签者不得销售。

#### 图书在版编目(CIP)数据

算法设计与分析基础(第2版 影印版) = Introduction to the Design and Analysis of Algorithms, Second Edition / (美) 莱维丁(Levitin, A.) 著. 一影印本. —北京: 清华大学出版社, 2007.11

(国外经典教材·计算机科学与技术)

ISBN 978-7-302-16402-9

I. 算… II. 莱… III. ① 算法设计—高等学校—教材—英文 ② 算法分析—教材—英文 IV. TP301.6

中国版本图书馆 CIP 数据核字 (2007) 第 169247 号

责任编辑: 汤涌涛

封面设计: 久久度文化

责任印制: 孟凡玉

出版发行: 清华大学出版社 地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编: 100084

[c-service@tup.tsinghua.edu.cn](mailto:c-service@tup.tsinghua.edu.cn)

社 总 机: 010-62770175 邮购热线: 010-62786544

投稿咨询: 010-62772015 客户服务: 010-62776969

印 刷 者: 清华大学印刷厂

装 订 者: 北京鑫海金澳胶印有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 37.5

版 次: 2007 年 11 月第 1 版 印 次: 2007 年 11 月第 1 次印刷

印 数: 1~4000

定 价: 59.00 元

---

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 022609-01

# 出版说明

近年来，我国的高等教育特别是计算机学科教育，进行了一系列大的调整 and 改革，急需一批门类齐全、具有国际先进水平的计算机经典教材，以适应当前我国计算机科学的教学需要。通过使用国外先进的经典教材，可以了解并吸收国际先进的教学思想和教学方法，使我国的计算机科学教育能够跟上国际计算机教育发展的步伐，从而培育出更多具有国际水准的计算机专业人才，增强我国计算机产业的核心竞争力。为此，我们从国外知名的出版集团 Pearson 引进这套“国外经典教材·计算机科学与技术”教材。

作为全球最大的图书出版机构，Pearson 在高等教育领域有着不凡的表现，其下属的 Prentice Hall 和 Addison Wesley 出版社是全球计算机高等教育的龙头出版机构。清华大学出版社与 Pearson 出版集团长期保持着紧密友好的合作关系，这次引进的“国外经典教材·计算机科学与技术”教材大部分出自 Prentice Hall 和 Addison Wesley 两家出版社。为了组织该套教材的出版，我们在国内聘请了一批知名的专家和教授，成立了一个专门的教材编审委员会。

教材编审委员会的运作从教材的选题阶段即开始启动，各位委员根据国内外高等院校计算机科学及相关专业的现有课程体系，并结合各个专业的培养方向，从 Pearson 出版的计算机系列教材中精心挑选针对性强的题材，以保证该套教材的优秀性和领先性，避免出现“低质重复引进”或“高质消化不良”的现象。

为了保证出版质量，我们为这套教材配备了一批经验丰富的编辑、排版、校对人员，制定了更加严格的出版流程。本套教材的译者，全部来自于对应专业的高校教师或拥有相关经验的 IT 专家。每本教材的责编在翻译伊始，就定期不间断地与该书的译者进行交流与反馈。为了尽可能地保留与发扬教材原著的精华，在经过翻译、排版和传统的三审三校之后，我们还请编审委员或相关的专家教授对文稿进行审读，以最大程度地弥补和修正在前面一系列加工过程中对教材造成的误差和瑕疵。

由于时间紧迫和受全体制作人员自身能力所限，该套教材在出版过程中很可能还存在一些遗憾，欢迎广大师生来电来信批评指正。同时，也欢迎读者朋友积极向我们推荐各类优秀的国外计算机教材，共同为我国高等院校计算机教育事业贡献力量。

清华大学出版社

# 国外经典教材·计算机科学与技术

## 编 审 委 员 会

主任委员:

孙家广 清华大学教授

副主任委员:

周立柱 清华大学教授

委员(按姓氏笔画排序):

王成山	天津大学教授
王 珊	中国人民大学教授
冯少荣	厦门大学教授
冯全源	西南交通大学教授
刘乐善	华中科技大学教授
刘腾红	中南财经政法大学教授
吉根林	南京师范大学教授
孙吉贵	吉林大学教授
阮秋琦	北京交通大学教授
何 晨	上海交通大学教授
吴百锋	复旦大学教授
李 彤	云南大学教授
沈钧毅	西安交通大学教授
邵志清	华东理工大学教授
陈 纯	浙江大学教授
陈 钟	北京大学教授
陈道蓄	南京大学教授
周伯生	北京航空航天大学教授
孟祥旭	山东大学教授
姚淑珍	北京航空航天大学教授
徐佩霞	中国科学技术大学教授
徐晓飞	哈尔滨工业大学教授
秦小麟	南京航空航天大学教授
钱培德	苏州大学教授
曹元大	北京理工大学教授
龚声蓉	苏州大学教授
谢希仁	中国人民解放军理工大学教授

# 读者赞誉

本书用一种全新的视角来审视算法分析这一领域。它不仅介绍那些排序和查找算法，而且教授算法设计和分析的根本思路，这样大家在需要的时候可以自己设计新算法。尽管本书只是一本入门教材，但它所奠定的基本思想可以让我们受用一生。

——Kevin J. Schmidt

就像作者在前言中所说，教授算法有两种方法。一种按照问题的类型对算法分类，另一种则根据设计技术来分类。第一类教科书会为排序、查找、图等问题安排独立的章节，这种书占绝大部分。它们更像是工具书，程序员会从中挑选合适的算法，做适当修改后来解决手头的问题。但这样的话我们能够利用的算法是有限的。如果有一个特定的问题需要一些全新的算法，我们就无能为力了。因为大多数算法书并没有教我们如何设计算法，也没有告诉我们对于特定的问题，什么算法设计技术是最适用的。但 Anany Levitin 的这本书填补了这个空白。他把排序、查找、图的各种算法按照不同的设计技术来分类，将蛮力法、分治法、贪婪技术等主要设计技术教给我们。结果许多面目迥异的算法竟然可以归纳为同一设计技术。例如冒泡排序、凸包问题、旅行商问题、背包问题都可以用蛮力法来求解。这样一来，在我们设计一个新算法的时候，我们一开始就可以知道它是哪种类型的问题，应该如何进行处理。

本书用一种全新的角度来教授算法的设计与分析，读来充满乐趣。而且每章后面的习题也非常精彩，大家赶紧买一本据为己有吧！

——Sameer Borate

虽然本书出版了没几年，但俨然已是一本经典教材了。这绝不是说它已是昔日黄花了，恰恰相反，本书的独特结构为算法教材做出了一个很好的榜样。从那以后，算法教材才改变了类似百科辞典的陈旧面貌。在本书面世之前，绝大多数算法书都是按照查找算法、图处理算法这样松散的关系把算法组织在一起的。但本书把算法按照几个重要的主线进行分组，使得我们学习算法设计基础知识的时候要容易许多。由于本书的巨大成功，近来越来越多的新算法教材也纷纷采纳了这种做法。大家千万不要错失本书！

——Saturnino Garcia

如果你一直想把 Cormen 的《算法导论》从窗户扔出去，现在终于可以如愿以偿了！本书就是一本非常优秀的算法导论教材。尽管没有 Cormen 那么面面俱到，但可读性非常强！

——Karl F. Schramm

最近学完了本书，我觉得它写得非常好，结构也很合理。大多数算法教材都难以阅读并且过于数学化。本书则更注重根据实际应用对算法做文字分析，而不是沉溺于那些理论证明。每章后面的习题对于理解书本内容非常有帮助，书后的习题提示也都非常到位。学习本身简直是一种乐趣，但也不乏挑战。

——William Franklin

# 前言

一个人接受科技教育的最大收获，是那些能够受用一生的通用智能工具<sup>①</sup>。

——George Forsythe, *What to do till the computer scientist comes*, 1968

无论是计算科学还是计算实践，算法都在其中扮演着重要角色。由于这一事实，这门学科中出现了大量的教材。它们在介绍算法的时候，基本上都选择了以下两种方案中的一种。第一种方案是按照问题的类型对算法分类。这类教材安排了不同的章节分别讨论排序、查找、图等算法。这种做法的优点是，对于解决同一问题的不同算法，它能够立即比较这些算法的效率。其缺点在于，由于过于强调问题的类型，它忽略了对算法设计技术的讨论。

第二种方案围绕着算法设计技术来组织章节。在这种结构中，即使算法来自于不同的计算领域，如果它们采用了相同的设计技术，就会被编成一组。从各方获得的信心(例如[BaY95])使我相信，这种结构更适合于算法设计与分析的基础课程。强调算法设计技术有三个主要原因。第一，学生们在解决新问题时，可以运用这些技术设计出新的算法。从实用的角度看，这使得学习算法设计技术成为一种很有价值的努力。第二，学生们会试图按照算法的内在设计方法对已知的众多算法进行分类。计算机科学教育的一个主要目的，就是让学生们知道如何发掘不同应用领域的算法间的共性。毕竟，每门学科都会倾向于把它的重要主题归纳为几个甚至一个规则。第三，依我看来，算法设计技术作为问题求解的一般性策略，在解决计算机领域以外的问题时，也能发挥相当大的作用。

遗憾的是，无论是从理论还是从教学的角度，传统的算法设计技术分类法都存在一些严重的缺陷。其中最显著的缺陷就是无法对许多重要的算法进行分类。由于这种局限性，这些书的作者不得不在按照设计技术进行分类的同时，另外增加一些章节来讨论特殊的问题类型。但这种改变将导致课程缺乏一致性，而且很可能会使学生感到迷惑。

## 算法设计技术的新分类法

传统算法设计技术分类法的缺陷令我感到失望，它激发我开发一套新的分类法[Lev99]，这套分类法就是本书的基础。以下是这套新分类法的几个主要优势。

- 新分类法比传统分类法更容易理解。它包含的某些设计策略，比如蛮力法、减治法、变治法、时空权衡和迭代改进——它们几乎从不曾被看作重要的设计范例。
- 新分类法很自然地覆盖了许多传统方法无法分类的经典算法(欧几里得算法、堆排序、查找树、散列法、拓扑排序、高斯消去法、霍纳法则等，不胜枚举)。所以，新分类法能够以一种连贯的、一致的方式表达这些经典算法的标准内容。

---

<sup>①</sup> 译注：George Forsythe 认为，在这些工具当中，最重要的三项依次是自然语言、数学和计算机科学。

- 新分类法很自然地容纳了某些设计技术的重要变种(比如, 它能涵盖减治法的 3 个变种和变治法的 3 个变种)。
- 在分析算法效率时, 新分类法与分析方法结合得更好(参见附录 B)。

## 设计技术作为问题求解的一般性策略

在本书中, 主要将设计技术应用于计算机科学中的经典问题(这里惟一的创新是引入了一些数值算法的内容, 我们也是用同样的通用框架来表述这些算法的)。但把这些设计技术看作问题求解的一般性工具时, 它们的应用就不仅限于传统的计算问题和数学问题了。有两个因素令这一点变得尤其重要。第一, 越来越多的计算类应用超越了它们的传统领域, 并且有足够的理由使人相信, 这种趋势会愈演愈烈。第二, 人们渐渐认识到, 提高学生的问题求解能力是高等教育的一个主要目标。为了满足这个目标, 在计算机科学课程体系中安排一门算法设计和分析课程是非常合适的, 因为它会告诉学生如何应用一些特定的策略来解决问题。

虽然我并不建议将算法设计和分析课程变成一门教授一般性问题求解方法的课程, 但我的确认为, 我们不应错过算法设计和分析课程提供的这样一个独一无二的机会。为了这个目标, 本书包含了一些和谜题相关的应用。虽然利用谜题来教授算法课程绝不是我的创新, 但本书打算通过引进一些全新的谜题来系统地实现这个思路。

## 如何使用本书

我的目标是写一本既不泛泛而谈, 又可供学生们独立阅读的教材。为了实现这个目标, 本书做了如下努力。

- 根据 George Forsythe 的观点(参见引言), 我试图着重强调那些隐藏在算法设计和分析背后的主要思想。在选择特定的算法来阐述这些思想的时候, 我并不倾向于涉及大量的算法, 而是选择那些最能揭示其内在设计技术或是分析方法的算法。幸运的是, 大多数经典算法满足了这个要求。
- 第 2 章主要分析算法的效率, 该章将分析非递归算法的方法和分析递归算法的典型方法区别开来。这一章还花了一些篇幅介绍算法经验分析和算法可视化。
- 书中系统地穿插着一些面向读者的提问。其中有些问题是经过精心设计的, 而且立即提供答案, 目的是引起读者的注意或引发疑问。其余问题的用意是防止读者走马观花, 不能充分理解本书的内容。
- 每一章结束时都会对本章最重要的概念和结论做一个总结。
- 本书包含大约 700 道习题。有些习题是为了给大家练习, 另外一些则是为了指出书中正文部分所涉及内容的重要意义, 或是为了介绍一些书中没有涉及到的算法。有一些习题利用了因特网上的资源。较难的习题数量不多, 会在教师用书中用一种特殊的记号标注出来(因为有些学生可能没有勇气做那些标有难度的习题, 所以本书没有对习题标注难度)。谜题类的习题用一种特殊的图标做标注。
- 本书所有的习题都附有提示。除了编程练习, 习题的详细解法都能够在教师用书



中找到,符合条件的教师可以填写书后的教师证明表,发传真到 010-62791865,以获得教师用书(也可联系 Addison-Wesley 公司的当地销售代表,或者访问 [www.aw.com/irc](http://www.aw.com/irc))。本书的任何读者都可以在 [www.aw.com/cssupport](http://www.aw.com/cssupport) 上找到 PowerPoint 格式的幻灯片文件。

## 第 2 版的变化

第 2 版中最重要的变化是新增了第 10 章,专门讨论迭代改进。增加这章有许多原因。

- 无论是从理论的还是从实践的角度看,许多计算机科学中最重要的算法,都是基于迭代改进技术的。
- 迭代改进策略是贪婪策略的重要替代,而贪婪技术在第 1 版中已有介绍。
- 鉴于第 6 章已经讨论了线性规划,那么继续介绍它的主要算法(单纯形法)也是很自然的。
- 新分类法的设计技术中,只有迭代改进没有收录在第 1 版。因此,出于对完整性的追求,也应该把它增补进来。

新的这一章包含 4 小节,涉及以下主题:单纯形法、最大网络流、二分图的最大基数匹配、稳定配对问题。和本书的其他各章一样,这章的每一小节都可以独立于其他小节来学习。

第 2 版还有两处增补值得一提,首先是 5.6 节减可变规模算法中,增加了一个关于拈游戏(game of Nim<sup>①</sup>)的子节。其次是扩充了旅行商问题近似算法的篇幅,因为近来这一领域取得了一系列的进展。

我还增加了大约 80 道习题(新习题的列表可以在本书网站 [www.aw-bc.com/info/levitin](http://www.aw-bc.com/info/levitin) 中找到)。其中大约一半习题包含在原有章节中,而且大部分属于谜题。根据我收到的反馈来判断,在习题中包含谜题很受读者欢迎,其中既包括教师也包括学生。和第 1 版一样,习题中只包含与所讨论主题相关的谜题。

## 读者所需的知识背景

本书假定读者已经学习了离散数学的标准课程和一门基础性的编程课程。有了这样的知识背景,读者应该能够掌握本书的内容而不会遇到太大的困难。尽管如此,1.4 节、附录 A 和附录 B 仍然对基本的数据结构,必须用到的求和公式和递推关系分别进行了复习和回顾。只有 3 个小节(2.2 节、11.4 节和 12.4 节)会用到一些简单的微积分知识,如果读者缺少必要的微积分知识,完全可以跳过这 3 个涉及微积分的小节,这并不会妨碍对本书其余部分的理解。

---

① 译注:“拈”本是中国发明的,它是一种民间游戏。Nim 是“拈”的广东话发音。

进度安排

如果打算开设一门围绕算法设计技术来讲解算法设计和分析理论的基础课程，可以采用本书作为教材。但要想在一个学期内完成该课程，本书涵盖的内容可能过于丰富了。大体上来说，跳过第 3~12 章的部分内容不会影响读者对后面部分的理解。本书的任何一个部分都可以安排学生自学。尤其是 2.6 节和 2.7 节，它们分别介绍了经验分析和算法可视化，这两小节的内容可以结合练习<sup>①</sup>布置给学生。

下面给出了一种针对一个学期课程的教学计划，这是按照 40 课时的集中教学来设计的。

课 次	主 题	小 节
1	课程简介	1.1~1.3
2, 3	分析框架： $O$ 、 $\Theta$ 和 $\Omega$ 符号	2.1, 2.2
4	非递归算法的数学分析	2.3
5, 6	递归算法的数学分析	2.4, 2.5(+附录 B)
7	蛮力算法	3.1, 3.2(+3.3)
8	穷举查找	3.4
9~10	分治法：合并排序、快速排序、折半查找	4.1~4.3
11	分治法的其他例子	4.4 或 4.5 或 4.6
12~14	减一算法：插入排序、DFS 和 BFS、拓扑排序	5.1~5.3
15	减常量算法	5.5
16	减变量算法	5.6
17~19	实例化简、预排序、高斯消去法、平衡查找树	6.1~6.3
20	改变表现：堆和堆排序 或者霍纳法则和二进制幂	6.4 或
		6.5
21	问题化简	6.6
22~24	时空权衡：串匹配、散列法、B 树	7.2~7.4
25~27	动态规划算法	8.1~8.4(选 3 节)
28~30	贪婪算法：Prim 算法、Kruskal 算法、Dijkstra 算法、哈夫曼算法	9.1~9.4
31~33	迭代改进算法	10.1~10.4(选 3 节)
34	下界的参数	11.1
35	决策树	11.2
36	$P$ 、 $NP$ 和 $NP$ 完全问题	11.3
37	数值算法	11.4(+12.4)

<sup>①</sup> 译注：“练习”的原文为“project”，一般应该翻译成“项目”，但国外一般将布置在课后完成的、较大的、要求实际演练的习题称为 project，国内没有相应的称呼，所以姑且译为“练习”。

续表

课 次	主 题	小 节
38	回溯法	12.1
39	分支界限法	12.2
40	$NP$ 困难问题的近似算法	12.3

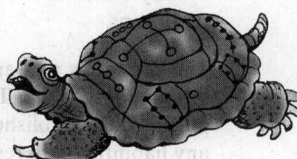
Anany Levitin  
anany.levitin@villanova.edu

Introduction to **The Design &  
Analysis of Algorithms**

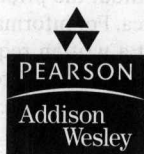
**2<sup>ND</sup> EDITION**

# Introduction to **The Design & Analysis of Algorithms**

**2ND EDITION**



**Anany Levitin**  
*Villanova University*



Boston San Francisco New York  
London Toronto Sydney Tokyo Singapore Madrid  
Mexico City Munich Paris Cape Town Hong Kong Montreal

Publisher	<i>Greg Tobin</i>
Acquisitions Editor	<i>Matt Goldstein</i>
Project Editor	<i>Katherine Harutunian</i>
Production Supervisor	<i>Marilyn Lloyd</i>
Marketing Manager	<i>Michelle Brown</i>
Project Management & Composition	<i>Windfall Software, using ZzT<sub>E</sub>X</i>
Technical Art	<i>George Nichols</i>
Copyeditor	<i>Richard Camp</i>
Cover Design	<i>Joyce Cosentino Wells</i>
Cover Illustration	<i>Jennifer M. Kohnke</i>
Senior Prepress Supervisor	<i>Caroline Fell</i>
Printer	<i>Courier Stoughton</i>

Access the latest information about Addison-Wesley titles from our World Wide Web site:  
[www.aw.com/cs](http://www.aw.com/cs)

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Addison-Wesley was aware of a trademark claim, the designations have been printed in initial caps or all caps.

The programs and applications presented in this book have been included for their instructional value. They have been tested with care but are not guaranteed for any purpose. The publisher does not offer any warranties or representations, nor does it accept any liabilities with respect to the programs or applications.

#### **Library of Congress Cataloging-in-Publication Data**

Levitin, Anany.

Introduction to the design & analysis of algorithms / Anany Levitin.

p. cm.

Includes bibliographical references and index.

ISBN 0-321-35828-7 (alk. paper)

1. Computer algorithms. I. Title: Introduction to the design and analysis of algorithms.

II. Title.

QA76.9.A43L48 2005

005.1-0dc22

2005033620

Copyright © 2007 by Pearson Education, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. Printed in the United States of America. For information on obtaining permission for use of material in this work, please submit a written request to Pearson Education, Inc., Rights and Contracts Department, 75 Arlington Street, Suite 300, Boston, MA 02116, fax your request to 617-848-7047, or e-mail at <http://www.pearsoned.com/legal/permissions.htm>.

ISBN 0-321-35828-7

1 2 3 4 5 6 7 8 9 10—CRS—09 08 07 06

---

# Preface

The most valuable acquisitions in a scientific or technical education are the general-purpose mental tools which remain serviceable for a life-time.

—George Forsythe, "What to do till the computer scientist comes," 1968

**A**lgorithms play the central role in both the science and practice of computing. Recognition of this fact has led to the appearance of a considerable number of textbooks on the subject. By and large, they follow one of two alternatives in presenting algorithms. One classifies algorithms according to a problem type. Such a book would have separate chapters on algorithms for sorting, searching, graphs, and so on. The advantage of this approach is that it allows an immediate comparison of, say, the efficiency of different algorithms for the same problem. The drawback of this approach is that it emphasizes problem types at the expense of algorithm design techniques.

The second alternative organizes the presentation around algorithm design techniques. In this organization, algorithms from different areas of computing are grouped together if they have the same design approach. I share the belief of many (e.g., [BaY95]) that this organization is more appropriate for a basic course on the design and analysis of algorithms. There are three principal reasons for emphasis on algorithm design techniques. First, these techniques provide a student with tools for designing algorithms for new problems. This makes learning algorithm design techniques a very valuable endeavor from the practical standpoint. Second, they seek to classify multitudes of known algorithms according to an underlying design idea. Learning to see such commonality among algorithms from different application areas should be a major goal of computer science education. After all, every science considers classification of its principal subject as a major if not the central point of its discipline. Third, in my opinion, algorithm design techniques have considerable utility as general problem solving strategies, applicable to problems beyond computing.

Unfortunately, the traditional classification of algorithm design techniques has several serious shortcomings from both theoretical and educational points of view. The most significant of these shortcomings is the failure to classify many important algorithms. This limitation has forced the authors of textbooks to depart from the design technique organization and to include chapters dealing with specific problem types. Such a switch leads to a loss of course coherence and almost unavoidably creates confusion in students' minds.

## **New Taxonomy of Algorithm Design Techniques**

My frustration with the shortcomings of the traditional classification of algorithm design techniques has motivated me to develop a new taxonomy of them [Lev99], which is the basis of this book. Here are the principal advantages of the new taxonomy:

- The new taxonomy is more comprehensive than the traditional one. It includes several strategies—brute force, decrease-and-conquer, transform-and-conquer, space and time tradeoffs, and iterative improvement—that are rarely if ever recognized as important design paradigms.
- The new taxonomy covers naturally many classic algorithms (Euclid's algorithm, heapsort, search trees, hashing, topological sorting, Gaussian elimination, Horner's rule, to name a few) that the traditional taxonomy cannot classify. As a result, the new taxonomy makes it possible to present the standard body of classic algorithms in a unified and coherent fashion.
- It naturally accommodates the existence of important varieties of several design techniques. (For example, it recognizes three variations of decrease-and-conquer and three variations of transform-and-conquer.)
- It is better aligned with analytical methods for efficiency analysis (see Appendix B).

## **Design Techniques as General Problem Solving Strategies**

Most applications of the design techniques in the book are to classic problems of computer science. (The only innovation here is the inclusion of some material on numerical algorithms, which are covered within the same general framework.) But these design techniques can be considered general problem solving tools, whose applications are not limited to traditional computing and mathematical problems. Two factors make this point particularly important. First, more and more computing applications go beyond the traditional domain, and there are reasons to believe that this trend will strengthen in the future. Second, developing students' problem solving skills has come to be recognized as a major goal of college education. Among all the courses in a computer science curriculum, a

course on the design and analysis of algorithms is uniquely suitable for this task because it can offer a student specific strategies for solving problems.

I am not proposing that a course on the design and analysis of algorithms should become a course on general problem solving. But I do believe that the unique opportunity provided by studying the design and analysis of algorithms should not be missed. Toward this goal, the book includes applications to puzzles and puzzle-like games. Although using puzzles in teaching algorithms is certainly not a new idea, the book tries to do so systematically by going well beyond a few standard examples.

## Textbook Pedagogy

My goal was to write a text that would not trivialize the subject but would still be readable by most students on their own. Here are some of the things done toward this objective.

- Sharing the opinion of George Forsythe expressed in the epigraph, I have sought to stress major ideas underlying the design and analysis of algorithms. In choosing specific algorithms to illustrate these ideas, I limited the number of covered algorithms to those that most clearly demonstrate an underlying design technique or analysis method. Fortunately, most classic algorithms satisfy this criterion.
- In Chapter 2, which is devoted to efficiency analysis, the methods used for analyzing nonrecursive algorithms are separated from those typically used for analyzing recursive algorithms. The chapter also includes sections devoted to empirical analysis and algorithm visualization.
- The narrative is systematically interrupted by questions to the reader. Some of them are asked rhetorically, in anticipation of a concern or doubt, and are answered immediately. The goal of the others is to prevent the reader from drifting through the text without a satisfactory level of comprehension.
- Each chapter ends with a summary recapping the most important concepts and results discussed in the chapter.
- The book contains about 700 exercises. Some of them are drills; others make important points about the material covered in the body of the text or introduce algorithms not covered there at all. A few exercises take advantage of Internet resources. More difficult problems—there are not many of them—are marked with a special symbol in the Instructor's Manual. (Because designating problems as difficult may discourage some students from trying to tackle them, problems are not marked in the book itself.) Puzzles, games, and puzzle-like questions are marked in the exercises with a special icon.
- The book provides hints to all the exercises. Detailed solutions, except for programming projects, are provided in the Instructor's Manual, available

