HZ BOOKS

从无有 到无穷

第2版

算法之道

Ex Nihilo:Algorithmica Logos

Second Edition

邹恒明 著

机械工业出版社
China Machine Press

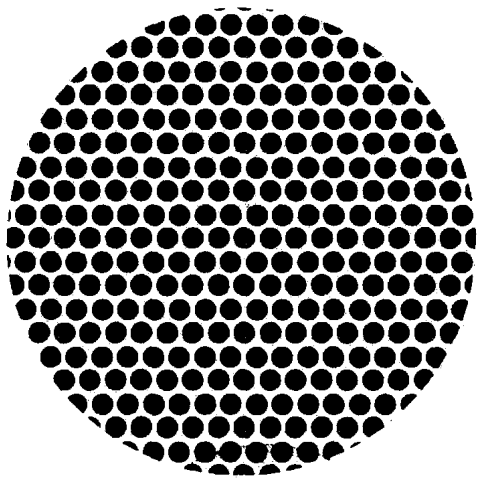
从无有 到无穷

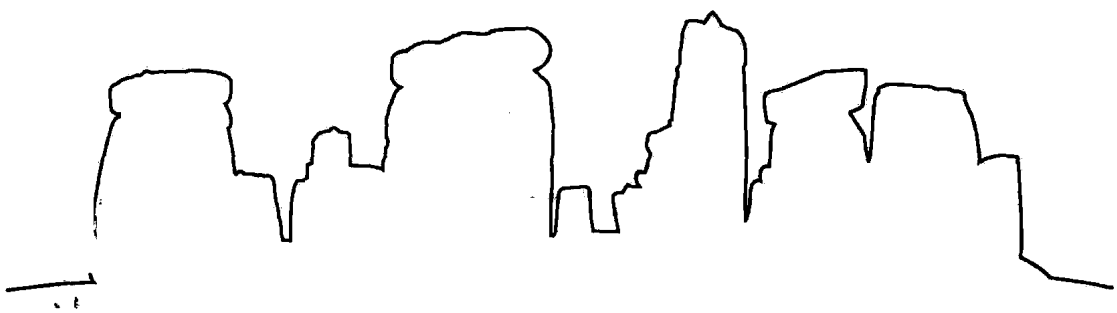
算法之道

Ex Nihilo:Algorithmica Logos

Second Edition

邹恒明 著





机械工业出版社
China Machine Press

本书追求的目标是算法背后的逻辑，是一本启示书，而不是一本包罗万象的算法大全。因此，本书甄选了那些最能展现算法思想、战略和精华，并能够有效训练算法思维的内容。本书将算法的讨论分为五篇：算法基础篇、算法设计篇、算法分析篇、经典算法篇、难解与无解篇。每篇分别讨论算法的一个方面：基础、设计、分析、经典和难解问题。第 2 版还对进程调度问题、跳转表问题、概率分析应用、遗传算法等方面进行了论述。

本书既可以作为大学本科或研究生的算法教材或参考书，也可以作为对算法有兴趣的读者提升认知深度的读物。

图书在版编目（CIP）数据

算法之道 / 邹恒明著. —2 版. —北京：机械工业出版社，2012.1

ISBN 978-7-111-37050-5

Ⅰ. 算…　Ⅱ. 邹…　Ⅲ. 电子计算机-算法理论　Ⅳ. TP301.6

中国版本图书馆 CIP 数据核字（2012）第 002416 号

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：盛思源

北京京北印刷有限公司印刷

2012 年 4 月第 2 版第 1 次印刷

186mm×240mm • 21.5 印张

标准书号：ISBN 978-7-111-37050-5

定价：59.00 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

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

求于至简，归于永恒

In Pursuit of Absolute Simplicity

谨以此书献给夫人蕾蕾，女儿雨洁、雨蓉、雨恒、雨宜。

To my wife Lily and daughter Charissa, Elizabeth, Grace, Ida.

前　言

起初神创造天地。地是空虚混沌，渊面黑暗；神的灵运行在水面上。神说："要有光。"就有了光。神看光是好的，就把光与暗分开了。神称光为昼，暗为夜。有晚上，有早晨，这是头一日。

……

神就照着自己的形象造人。

……

神说："看啊！我将遍地上一切结种子的菜蔬和一切树上所结有核的果子，全赐给你们做食物。至于地上的走兽和空中的飞鸟，以及各样在地上爬的有生命的物，我将青草赐给它们做食物。"事就这样成了。

神看着一切所造的都甚好。有晚上，有早晨，是第六日。天地万物都造齐了。

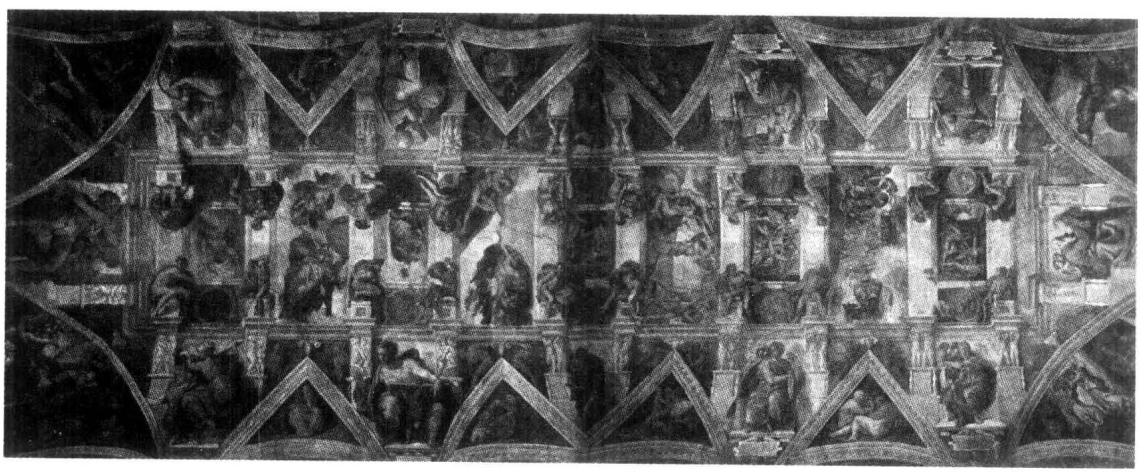


图 1　米开朗基罗创作的西斯廷教堂穹顶画《创世纪》。这幅画里隐含着算法

6天

圣经上写着：神 6 天创造天地万有，第 7 日安歇。

对于神创论者来说，这是不可怀疑的事实。但对于进化论者来说，6 天创造一切根本就不可能。

作为一本算法书，我们当然不打算加入神创论和进化论者的永无休止的争论当中去。我们关心的是这么一个问题：圣经上为什么给出的是 6 天，而不是其他的时间长度。不管是神创论者还是进化论者，弄清楚 6 这个数字的来历很可能会对己方的观点有所帮助。在这 6 天里，神将他的创作方程式重复了 6 次，每天 1 次。对于全能的神来说，他完全可以在 1 天、1 秒或者任何他所愿意的时间长度里创造天地万物，但却为什么是不多不少的 6 天呢？而不管圣经上的“1 天”是多长，这个问题都是值得讨论的。

我们知道，任何一个自然数的约数中都有 1 和它本身，而所有小于它本身的因数叫做这个自然数的真约数。例如，6 的所有真约数是 1、2、3；数字 8 的真约数是 1、2、4。如果一个数的真约数之和等于这个自然数本身，则这个自然数就称为完全数，或者完美数。例如，6=1 + 2 + 3，因此 6 是完美数；而 8≠1+2+4，因此 8 不是完美数。因此，神 6 天创造世界，暗示着该创造是完美的！

以完美数来昭示创造的完美，似乎合情合理。但问题是，完美数只有 6 这一个数吗？如果不是，为什么不使用其他的完美数呢？答案是，完美数虽然不只有 6 这一个，但确实数量稀少。一直到现在（2009 年 6 月），数学家们探索了 2 600 年，并且现代数学家们还借助了超级计算机的帮助，但也仅仅找到了 47 个完美数。其中第 1 个完美数是 6，接下来的 4 个完美数分别是： 28、496、8 128、33 550 336。而第 47 个完美数有 25 956 377 个数位（注意，是数位，不是数值），它的数值为：$2^{43\,112\,608} \times (2^{43\,112\,609} - 1)$。

完美数的稀少昭示着达到完美的难度，而神选择 6 天来创造天地万有也许是因为 6 是最小的完美数，即创造天地万有对于神来说是轻而易举的一件事情……

完美与算法

完美数由于其各种神秘属性（真约数之和等于自身只是其中的一种性质）而受到了特殊的关注。但到底哪些数是完美数则不是一件容易判断的事情。显然，按照完美数的定义，判断一个数是否是完美数的不二法则是找出它的所有真约数，然后求和看看其是否等于自身。而这种方法效率太过低下，因为这意味着因式分解，而这是十分困难的（本书后面将会讨论到这个问题）。

如果判断一个数是否是完美数就已经非常困难，那么要找出所有的完美数则更是一个难上加难的任务。因为这就意味着将所有的数进行上面描述的判断验证：因式分解。这似乎是人类不可能完成的任务。即使用世界上超快的计算机来进行计算，情况也不会有任何数量级的改善。

显然，我们需要新的解决方案，而不是发明或使用新的计算工具！研究这样的解决方案就可以归结到算法的范畴里，因为如何高效地解决问题正是算法要研究的核心课题。

有意思的是，判断和搜索完美数是算法的研究范畴，而算法本身的追求却也是“完美”（见图 2）。

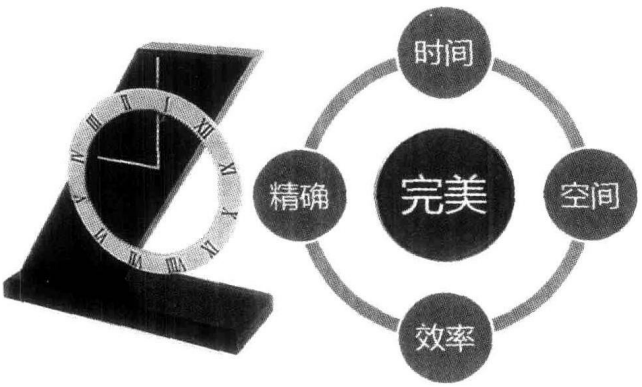


图 2 算法所追求的理想就是“完美”

算法无处不在

如果你觉得算法只是用来研究解决找出完美数之类的“漫无边际的问题”，那就大错特错了。

也许算法这个名词听上去很抽象，让人联想不到任何具体的物体。也许你会觉得算法与自己的生活并无太多关系，它只不过存在于那些闲得无聊的数学家或计算机专业人士的脑海中。

但事实真是这样吗？当然不是。如果我们告诉你算法就是解决各种问题的方法，你就不会觉得它太抽象，与生活无关了吧。事实是，算法无处不在。每个人每天都在使用不同的算法来活出自己的人生，比如你去食堂买饭会选择一个较短的队列，而有人则可能选择一个推进速度更快的队列。每天起床后，你可能先读一会儿书，再去吃早饭；另外一个人则可能先去吃早饭，然后再看书。所有这些行为都是算法或算法一部分的体现。也许运行这些算法并不在你的思想意识里，也许你并不知道算法在帮助自己的生活，但它确实存在。这些算法也许没有经过精心设计，没有经过仔细分析，但它还是算法！

2009 年 7 月 23 日下午，我在游览云南省大理市的蝴蝶泉时由于泉水边的石头很滑，在用泉水洗手时（导游金花说用该泉水洗手会带来好运）不慎滑落到蝴蝶泉水（见图 3）里面全身湿透。（据说一天至多只会有一个人滑落到泉里，可见我的运气不错！看来“蝴蝶泉边好梳妆”的歌词也许应该改为“蝴蝶泉里好冲凉”。）泉水冰冷透凉，而大理的气温又低。这样，我就面临一个是否更换全身衣服的选择。问题是，旅游团需要马上赶去登游船游览洱海风光，而若找地方或者回旅店换衣服就将赶不上游船。

如何处理这件事情就是一个算法问题：是先上游船再在船上找地方换衣服，还是找个地方换衣服而放弃游览苍山洱海。显然不同的算法有着不同的收益和代价。如果能够在游船上找到合适的地方更换衣服，则采用先上游船再换衣服的算法为佳；否则就是放弃游览的算法更好，因为如果冻病了显然就不划算了。最后，我选择了在游船上更换衣服的算法：在游船上找到了一个贵宾室更衣。

图 3 在蝴蝶泉水下洗个手也会涉及算法

算法由问题驱动

算法的发现总是由相关的问题驱动的。拿排序来说，因为生活中到处都充满次序，每个人都要接受自己在某个次序里的位置。比如，各种排名、评优、民意调查等，最后的结果都体现为一个次序！看来，“没有次序无以成方圆”并不是空穴来风！而谈到排序用的方法，人们很自然地想到了插入法。因为这种朴素的算法和人的思维方式非常类似：它就是人们打牌时整理手中扑克牌的算法。

但是随着数据量的增多，插入排序的效率缺陷迅速变为人们无法容忍的缺点。于是人们发明了归并排序、堆排序、快速排序等，这些排序的方法大大改善了速度，但是人们却并不满足于此，因此又发明了效率更高的线性排序。表 1 给出的是各种排序算法平均情况下的效率比较：最上面一行的数字代表输入的规模，如 10 表示一共有 10 个数据项，1M 表示一共有 100 万个数据项。其他格子里面的数据为相应算法在相应输入规模下完成排序所需要的时间，单位为毫秒。所有输入数据为随机产生。

表 1 部分排序算法的时间效率比较 （单位：毫秒）

排 序 算 法	10	100	1k	10k	100k	1M
冒泡排序	0.000 276	0.005 643	0.545	61	8 174	549 432
选择排序	0.000 237	0.006 438	0.488	47	4 717	478 694
插入排序	0.000 258	0.008 619	0.764	56	5 145	515 621
希尔排序/增量 3	0.000 522	0.003 372	0.036	0.518	4.152	61
堆排序	0.000 45	0.002 991	0.041	0.531	6.506	79

（续）

排序算法	10	100	1k	10k	100k	1M
归并排序	0.000 723	0.006 225	0.066	0.561	5.48	70
快速排序	0.000 291	0.003 051	0.030	0.311	3.634	39
基数排序/进制 100	0.005 181	0.021	0.165	1.65	11.428	117
基数排序/进制 1 000	0.016 134	0.026	0.139	1.264	8.394	89

注：1. 算法运行环境为 Intel 酷睿 2 双核 E8400，3.0GHz, Windows 7x64。

2. 本表数据由作者所授“数据结构”课的胡嘉斌同学测试所得。

一个个新的算法都是为了解决前面算法遗留的问题而产生的。从表 1 里的数字可以看出，一般来说，随着新的算法出现，排序效率在不断提高。不过，虽然每个算法似乎解决了前面算法的遗留问题，但新的问题也会被有意或无意地引入。例如，线性排序虽然将排序的时间复杂性降低到线性级，但各种前提条件极大地限制了其应用范围。也许这就是算法永远也不能或不会停止发展的一个原因吧。

算法是计算机的灵魂

因为人不是全能的，一个时刻只能做一件事情，所以做事情就要有一个步骤。由于算法要满足人的这种特性，因此它通常表示为一个做事情的行为序列。因此，从一般意义上说，算法就是求解问题的步骤。由于计算机的计算操作完全是一步一步进行，因此算法的上述性质用于计算机是再合适不过了，可以说算法弥漫在计算机的一切行为上。如果说操作系统是计算机的心智，那么算法就是计算机的灵魂。

理解灵魂当然不是一件容易的事情，由于它高度抽象与简洁，许多学生都望而却步。先看一个纸牌魔术（见图 4）：

1）任选一位观众将一副扑克牌充分洗好。

2）背对观众，请观众随机抽出一张牌，记住牌面，然后将这张牌放回整副牌的最上面。

3）接过牌后，洗牌几次。洗牌的时候保持最上面一张牌不动。

4）对观众说：“我来教你魔法，只要吹一口气，就能把刚才你抽的牌吹到任意位置上”。

5）请观众说出一个数字，比如说 10，然后一边吹气，一边想着刚才说的数字 10。

6）在吹完气后，请观众一张一张地将上面的牌取出放在桌上。

7）到第 10 张时，将牌翻开，发现并不是其原来抽的牌。

8）接回整副牌，并把上一个步骤里取出堆放在桌上的牌收起，仍放在整副牌的最上面。

9）然后洗牌几次，洗牌的时候保持上面放回来的那堆牌不动。

10）从观众手上拿回刚才翻开的那张牌，插入最上面 9 个位置中的任意一个。

11）对观众说：“你刚才不是在想着那个数字的时候吹的气，而是在吹气的时候想着那

个数字，而这是完全不同的两回事。我现在演示如何吹气。”对着牌吹一口气。

12）请观众从上到下数牌，到第 10 张时翻开。

13）这张翻开的牌就是观众一开始抽的那张牌。

读者看明白了上面的这个魔术了吗？这里面隐藏着一个算法。如果看懂了就可以在朋友面前一显身手了。当然，如果没有看懂也没有什么关系。算法本来就不是轻易让人看懂的嘛。

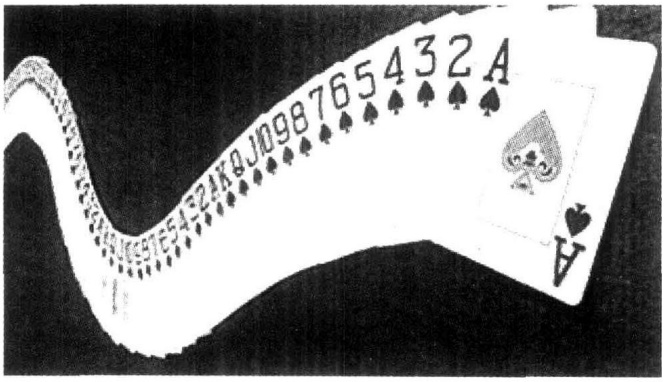


图 4　算法无处不在，就连扑克魔术都有其背后支持的算法

对于一些吹毛求疵的人来说，也许会说这个纸牌魔术不是算法。至少这与我们研究算法的人打交道的常见算法不太一样。这没有什么关系，来看下面的一段伪代码：

```
PARTITION(A, p, q)        //A是一个实数数组，p、q是该数组的上下限
x=A[p];                   //A[p]被选中
i=p;
for( j=p+ 1; j <=q; j++) {
     if(A[ j] <=x) {
          i=i+ 1;
          temp = A[i];          //以下三行交换A[i] 和A[j]的内容
          A[i] =A[j];
          A[ j ]=temp;
    }
}
temp = A[i];                    //以下三行交换A[i] 和A[p]的内容
A[i] =A[p];
A[p]=temp;
return i;
```

读者能看出来这个伪代码程序片段完成的是什么功能吗？

要分析一个算法，似乎就更难了。读者能看出下面的 C 程序片段里面“laugh++”语句执行了多少次吗？

```
for (i=1; i<=n; i*=2)
```

```
            for (j=1; j<=i; j++)
        laugh++;
```

如果这些问题读者都能回答，那么恭喜你。看来算法分析对于读者来说将是件很容易的事情，不过可能也不一定。如果你回答不出这些问题，不用担心，因为回答诸如此类的问题就是本书的目的。当然了，本书回答的远不止这么几个简单问题，而是会阐述更重要的算法精髓：算法思想、战略和分析！

本书内容安排

本书追求的目标是算法背后的逻辑。因此，它不可能是一本包罗万象的算法大全，而是一本启示书。因此，本书甄选了那些最能够展现算法思想、战略和精华，并能够有效训练算法思维的内容。本书的选材遵循的规则是：书中选取的每个算法都在某个方面具有独特性，能够彰显算法的精髓。

本书将算法的讨论分为五篇：算法基础篇、算法设计篇、算法分析篇、经典算法篇、难解与无解篇。每篇分别讨论算法的一大方面：基础、设计、分析、经典和难解问题。如图 5 所示。

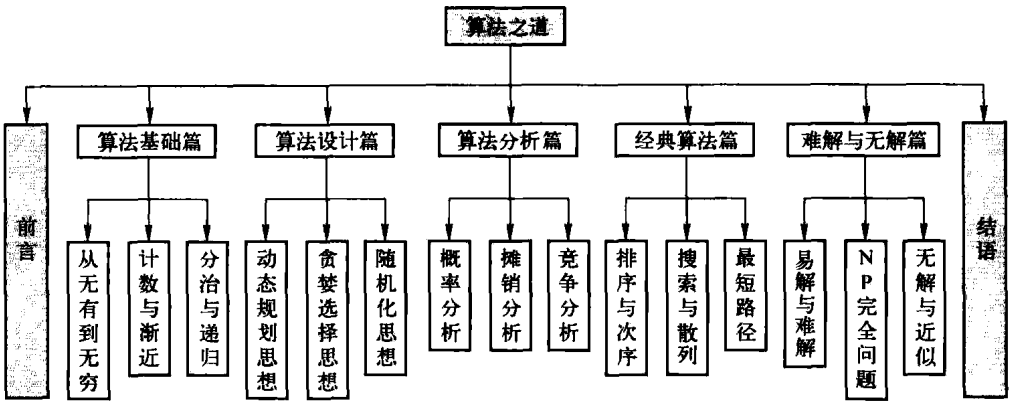


图 5 本书内容框架

本书第一篇是算法基础篇，讨论基本的算法设计思想与算法分析方法和手段，包括第 1～3 章。第 1 章讨论从无有到无穷、什么是算法、算法的表示、算法之魂、算法与计算机的关系、算法的范畴和为什么学习算法。第 2 章讨论算法的正确性、时空效率和时空特性分析、计数分析方法、算法设计、渐近表示与分析。第 3 章阐述算法设计的最基本战略：分治与递归。具体内容包括分而治之为上策、分治策略中的递归、求解递归表达式、乘法及乘方运算、矩阵乘法、斐波那契数的计算、VLSI 布线和多项式乘法。

第二篇为算法设计篇，讨论算法中常见且重要的战略或思想，包括第 4～6 章。第 4 章讨论什么是动态规划、流水线问题、最长公共子序列问题、最优二叉搜索树问题、记忆递归

法、最优子结构、重叠子问题、动态规划与静态规划之间的关系。第 5 章介绍人在生活中潜意识地遵守的一种行为：贪婪。具体内容包括什么是贪婪、贪婪选择属性、背包问题、教室规划（排课）问题、最小生成树问题、霍夫曼编码问题、标准分治、动态规划和贪婪策略的比较。第 6 章讨论人生及算法中无处不在的随机，内容包括为什么要随机化、随机的平方、拉斯维加斯算法、蒙特卡罗算法、素性测试、矩阵乘积验证器、随机化最小生成树算法。

第三篇为算法分析篇，主要介绍计数分析之上的算法分析的另外三种重要手段：概率分析、摊销分析和竞争分析，包括第 7～9 章。第 7 章包括一切都在概率中、什么是概率分析、梦幻情人的代价、概率分析结果的有效性、正确概率分析的保障、梦幻情人的概率、随机排列问题和从无穷到无有。第 8 章讨论什么是摊销分析、摊销分析与数据结构、摊销分析的方法、聚类分析、会计分析、势能分析、动态表扩张及其摊销。第 9 章讨论竞争无处不在、在线算法和离线算法、竞争力、健忘对手和优良对手、线性表更新问题、前置移动算法及其竞争分析、聚类问题及其竞争分析、竞争分析与普通算法分析的比较。

第四篇是经典算法篇，包括第 10～12 章。第 10 章包括插入排序、折半插入排序、归并排序、快速排序、随机化快速排序、排序的下限、线性排序、求最大值、求最小值、求中值、任意次序选择。第 11 章包括搜索问题定义、顺序搜索、折半搜索、常数搜索、散列搜索、散列函数选择、散列冲突的解决、随机化散列、全域散列和完美散列。第 12 章包括单源单点最短路径、单源多点最短路径、多源多点最短路径。具体算法则包括穷举搜索算法、Dijkstra 算法、Bellman-Ford 算法、Floyd-Warshall 算法和 Johnson 算法。

第五篇是难解与无解篇，包括第 13～15 章。第 13 章讨论易解与难解、决策和优化、P 和 NP、确定性与非确定性。第 14 章讨论 NP 完全问题、多项式时间规约、如何证明一个问题 *S* 是 NP 完全、库克定理、3-SAT 问题、整数规划问题、独立集问题、汉密尔顿回路问题、弱 NP 完全、强 NP 完全和中 NP 完全。第 15 章包括意志的胜利、难解问题、不可解问题、程序终结的判断、难解之题的求解、不可决定问题、难解之题的求解、智能穷举、近似算法、本地搜索、回溯策略、分支限界、贪婪近似策略、启发式搜索策略、模拟退火算法和基因／遗传算法。

本书以《创世纪》的 6 天为起点，寓意算法也有创始的一刻，将人类算法体系中优美而有代表性的内容囊括书中，最后以算法之道的随想作为结尾，构成了逻辑上一气呵成，思想上韵味深长的算法知识体系。

本书的特点

我相信，写书的目的是对读者有所启示（enlightenment），而不是用一大堆的公式或繁琐的推导来烦死或吓倒读者。虽然在一定的时候也会迫不得已地使用复杂的公式，但不必到处都引用复杂繁琐的表述，甚至将简单的东西也以繁琐的式子来表达。复杂的式子或能给部分人带来一丝莫名其妙的快感，但对于大多数读者来说，也许就是“装腔作势罢了，真可憎恶”。基于此种认识，本书将以简洁的方式来表示复杂的概念。

在我读小学的时候，有一天在街上听到一个人吆喝："快来看，快来看开膛破肚表演！"有人问："怎么开膛破肚？"卖艺的人说："用刀将活人的胸膛破开，将里面的内脏拿出来给大家看，然后再缝上。"开膛破肚？这可是难得一见的事情。于是我按照卖艺人的指引进入一个很小的黑屋里，里面已经挤进了一些人，站在屋子四周。屋中间的床架上躺着一个上身赤裸的年轻人，旁边则站着一个手拿尖刀的"屠夫"。"屠夫"先叫每个观众交了一笔钱，然后开始了他的开膛破肚表演。

只见"屠夫"将尖刀举起，对着躺在床架上的年轻人喊道："你要钱还是要命？"年轻人很坚定地回答："要钱！""屠夫"连喊了三遍，得到的回答都是"要钱"。于是"屠夫"将尖刀快速砍下，刺进了年轻人的肚脐，血从尖刀的四周往外渗出。然后"屠夫"对着四周的观众喊道："你们看这个人要钱不要命，你们给他一些钱吧。"很多人看到这个流血的场面，出于同情或害怕又向外掏钱。

令人遗憾的是，"屠夫"并没有遵守许诺将年轻人的腹部或者胸膛划开，也没有将脏器拿出来给大家看……

这是江湖杂耍的一些小伎俩，但本书对算法进行的"开膛破肚"的分析却是实打实的！这也是本书最显著的特点：**对算法的分析深入到以往没有深入的境界**。本书更关注的是算法后面的逻辑脉络，强调一个算法为什么会出现，又为什么会是现在呈现的样子。通过挖掘算法背后的思维过程，本书淋漓尽致地展现了算法的精妙绝伦。

本书的第二个特点是**结构紧凑**：摒弃臃肿繁琐的内容堆砌，通过逻辑关系将算法各部分内容进行递进演绎，形成一个层次感强、由表入里的有机体。

本书的第三个特点是**全新的角度**：也许讲的是同样的算法、相似的问题，但是本书采纳的是完全不同的角度，这种独特的视角能把我们对算法的理解带到新的高度。

本书的第四个特点是**新颖的结构**：不同一般的章节组织使条理更为清晰，内容上也包含了不少新的概念和理念。

本书的最后一个特点是**写作风格轻松活泼**：以讲故事的形式将概念和算法的精华娓娓道来，由浅入深，非常易于理解和消化。

上述这些特点赋予了本书与一般算法书或其他科技书籍的巨大不同（见图 6）。

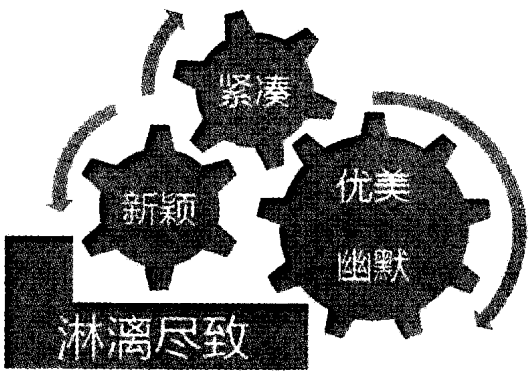


图 6 本书的特点

当然，本书也没有走另一个极端：过分强调语言的生动而忽视了严谨性。恰恰相反，本书力求兼顾这两个看似矛盾的方面。在书中我们看不到繁多的数学公式，取而代之的是精确的文字叙述。我认为，这种用严谨的语言代替数学形式化的方法更容易被读者接受。因为读者需要知道的，通常是蕴涵在各种公式或算法伪代码（或程序代码）背后的思想，而正是这些思想促成了所有精巧的算法。

本书几乎没有讲述数据结构的内容，也不会讨论算法的程序设计实现，而是集中精力论述算法本身的特点。虽然有的人认为算法与数据结构和程序设计关系密切，但是毕竟算法可以独立于它们而单独存在。事实上，早在计算机出现之前，算法就已经存在了。从相对的角度来看，算法是抽象的，数据结构与程序设计是具体的（当然从绝对意义上看，数据结构与程序设计也是很抽象的）。而越抽象就越具有普遍意义，也只有在比较抽象的层面上学习算法，才能看透算法的精妙。

当然，本书的讨论也不可能完全脱离数据结构或程序，有时候也会提到它们，有时候甚至直接给出某个算法的具体程序实现片段，但所有对数据结构和程序的论及皆点到为止，以有利于对算法的掌握和体现算法的实现为目的。而数据结构和程序设计的精妙讲解就留给“数据结构”和“程序设计”课程来完成吧。

此外，本书使用的伪代码采用类似于 C/C++的结构，每个算法的表示均以展示算法本身的思路为最高目标，并不对表示的细节进行优化，以使得逻辑清晰，方便绝大多数读者理解。

本书的使用方法

本书既可以作为大学本科或研究生的算法教材或参考书，也可以作为对算法有兴趣的读者提升认知深度的读物。如果作为教材使用，建议课程为 4 个学分，如果学时限制只有 3 个学分，建议将摊销分析（第 8 章）、竞争分析（第 9 章）和无解与近似（第 15 章）这三章内容跳过。建议课堂讲解顺序按书中安排进行，因为本书内容是按照逻辑演绎顺序环环相扣的。按这种顺序讲解条理清晰、逻辑明朗、前后连贯，学生比较容易接受。

对于一般读者，可以将本书作为一种算法思维的修养书来看，按照自己的时间和计划自行安排。或者休闲时翻看此书，斟酌算法，品味精妙，这不也是一件美事吗？

本书隐含 7 个悖论。如果读者能够发现这些悖论并找出答案，那么恭喜你！因为你有一双发现真理的眼睛。不，应该说，你有一颗发现真理的心！如果读者没有发现这些悖论，也没关系，因为能够发现这些悖论的人毕竟不多，更不用说寻找到答案了。而不管发现与否，这些悖论都不会妨碍读者对本书内容的理解。因为如果你发现了这些悖论，它们施加的是正面影响：读者对算法的理解深度会大大增加！而如果没有发现这些悖论，则对读者来说，这些悖论相当于不存在，自然也就无法对读者施加任何正面或负面的影响了。

另外，本书隐含 7 个重要的算法奥秘，但能否察觉就看个人的理解了（见图 7）。

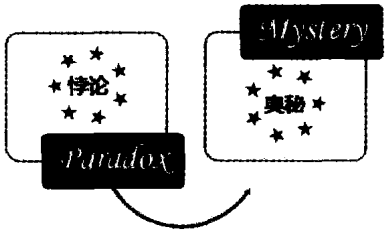


图 7 本书隐含着 7 个悖论和 7 个奥秘

逻辑演绎、生活归纳、趣味交织，入木三分地揭示算法的奥妙；新的角度、新的分析、新的境界，耳目一新地阐述算法的精华。就让我们即刻开始精彩纷呈的算法之旅，Let the Funs Begin!

目　录