



王晓华◎著

淋漓尽致展现算法本质
广泛涵盖常用算法结构及应用

算法领域超人气著作
一本书玩转算法，尽享算法乐趣

算法的乐趣

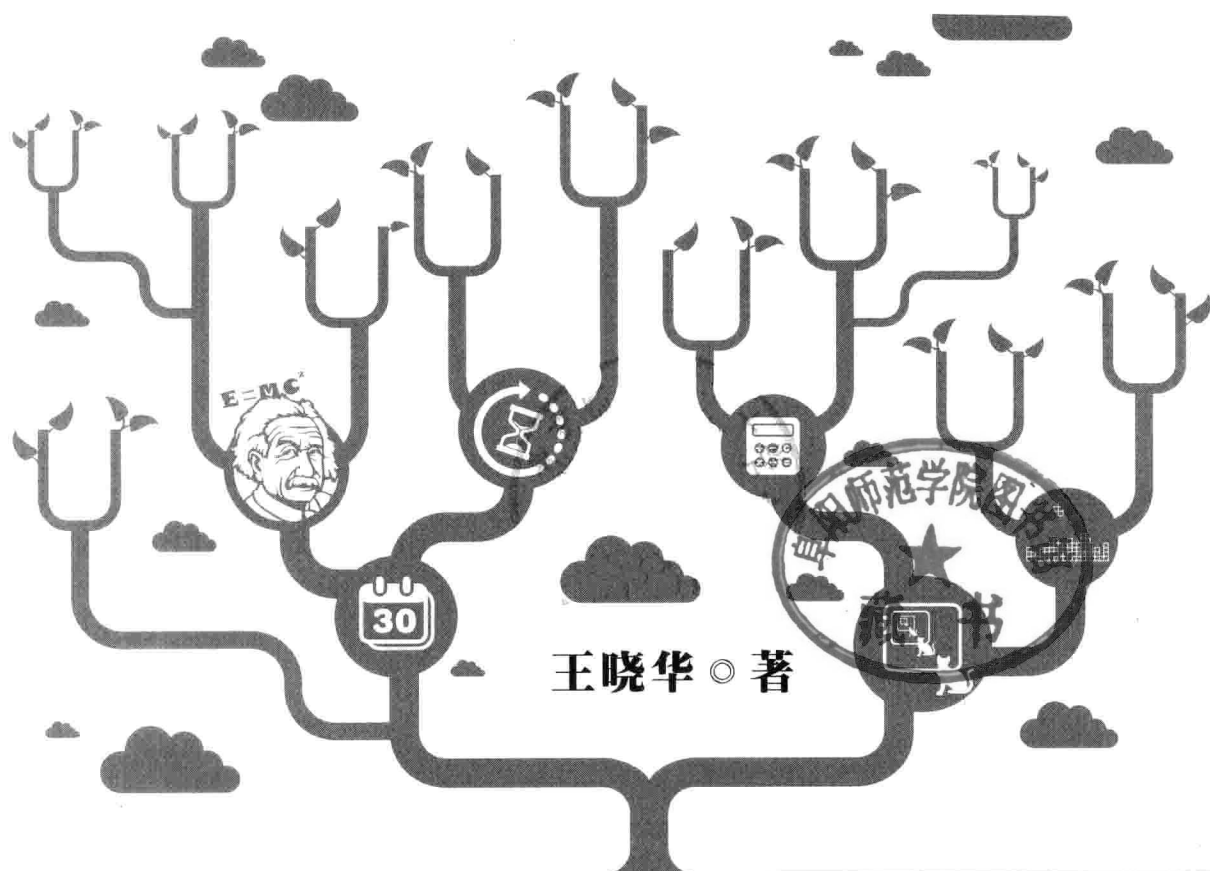


中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵原创



王晓华 著

算法的乐趣

人民邮电出版社

图书在版编目 (CIP) 数据

算法的乐趣 / 王晓华著. -- 北京 : 人民邮电出版社, 2015. 4
(图灵原创)
ISBN 978-7-115-38537-6

I. ①算… II. ①王… III. ①算法语言—程序设计
IV. ①TP312

中国版本图书馆CIP数据核字(2015)第033951号

内 容 提 要

本书从一系列有趣的生活实例出发,全面介绍了构造算法的基础方法及其广泛应用,生动地展现了算法的趣味性和实用性。全书分为两个部分,第一部分介绍了算法的概念、常用的算法结构以及实现方法,第二部分介绍了算法在各个领域的应用,如物理实验、计算机图形学、数字音频处理等。其中,既有各种大名鼎鼎的算法,如神经网络、遗传算法、离散傅里叶变换算法及各种插值算法,也有不起眼的排序和概率计算算法。讲解浅显易懂而不失深度和严谨,对程序员有很大的启发意义。书中所有的示例都与生活息息相关,淋漓尽致地展现了算法解决问题的本质,让你爱上算法,乐在其中。

本书适合软件开发人员、编程和算法爱好者以及计算机专业的学生阅读。

-
- ◆ 著 王晓华
责任编辑 王军花
执行编辑 张 霞
责任印制 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京鑫正大印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 26.25
字数: 614千字 2015年4月第1版
印数: 1~4 000册 2015年4月北京第1次印刷
-

定价: 79.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

序 一

读《算法的乐趣》的乐趣超出了我的预料。

说到算法，大部分计算机专业的同学的第一反应估计是MIT出版社的经典教材《算法导论》（*Introduction to Algorithms*）。这是一本由浅入深的好书，堪称“神书”——别看书挺厚，但是对初学者来说很难弄懂的问题也娓娓道来，让人看一遍就明白；而且作者用最简单的英语词汇和句法写书，以至于世界各地的学生们，不需要英语很好，即可读懂原版。只是看完这本大部头之后，总有一些意犹未尽的感觉——对我们日常生活中常见的比如音乐播放器里以及电子游戏里的算法并没有太多介绍。而这些正是《算法的乐趣》中主要的部分。

在Amazon上，另外两本排名靠前的经典算法教材是Jon Kleinberg的《算法设计》（*Algorithm Design*）和Steven S. Skiena的《算法设计手册》（*The Algorithm Design Manual*）。这两本出自名家之手的教材和很多教材一样，按照算法的类型或者背后的设计思路来组织内容。这是教材应该做的，“授人以鱼不如授人以渔”，传授思路而不是算法本身是教材的写作目的。可是算法最有意思的地方首先在于算法本身，因为算法是为了解决实际问题而设计的，所以让大家认识到算法奥妙的自然顺序应该是先展示有趣的问题，再展示优雅的算法，最后归纳设计思路。而这正是《算法的乐趣》吸引人的地方。

说到乐趣，总让我想起我学习和使用数学知识的经历。虽然我的学位是关于统计机器学习的，而且毕业后一直从事相关工作，但是我从小学一年级到博士第三年都对数学毫无兴趣，因为学校的老师和数学成绩好的同学都说不明白数学的用处，以至于我一直以为数学的作用只是锻炼和展示自己的聪明，博得老师的表扬，成为陈景润那样为国争光的英雄。而这些对我实在没有吸引力，而且我认为恐怕对绝大部分学生都没什么吸引力。

我认识到数学的价值，是因为在博士第三年把研究方向换到了统计机器学习。在读教材的时候，我曾想验证“数学无用”，所以费尽心力地试图写一个程序来判断一个 64×64 像素的图片里到底是数字“1”还是数字“9”，却发现无论如何也很难写一个有效的程序；可是利用教材里的数学知识却能设计和“训练”一个数学模型，准确地识别任意字符。因为体会到了数学的用处，我兴奋地用了一年的时间复习大学本科的数学课程，然后才读懂了人工智能的专业教材和论文。此后才有所创新，发表论文，到博士毕业。这个过程用了三年，而效果超过了之前19年数学教育的效果。

在这个过程中，我自然而然地开始注意数学知识的前因（比如为什么人们会关注长度、面积，怎么会有人考虑勾股定理这样的规律）以及后果（今天的数学知识能给物理学和机器智能带来什么样的帮助），也开始归纳和了解各种数学系统背后的规律，能体会哥德尔定理阐述的意思。当然，也破除了“数学是各种科学之母”之类的迷信，数学当然不是“科学之母”，而是“科学之子”，是先有物理学、力学和天文学，才有的数学；先有应用场景后有工具，先有探索后有归纳。

算法也是如此。先有工程问题需要解决，算法是解法，设计算法是寻求解法。虽然算法作为一门科学是归纳寻求解法的思路，但学习这种归纳法的前提是能体会各种具体算法的用处和效果。意识到这一点，自然也就破除了诸如“学好数学才能学好算法”之类的迷信。而把算法解决的各种有趣问题罗列出来，把算法的可爱之处展示给愿意发现和体会生活中点滴乐趣的读者们，正是《算法的乐趣》在技术价值之外的一层社会价值。

十年前，当我们坐在课堂里学习算法的时候，我们学到的是如何用人脑寻求解法，然后把解法写成程序，让计算机照着执行去解决问题。这是“经典算法”。最近十几年，随着Internet产业的兴起，Internet服务在不断取代原来由人提供的服务，这就要求机器拥有一定程度上能取代人的“智能”。在搜索引擎、推荐系统和广告系统等各个领域里，类似上述“识别数字”的问题越来越多，而人工智能和机器学习的应用也越来越深入我们的生活。机器学习算法的设计目标和“经典算法”不同——不是让人来想解法，而是让计算机从数据归纳知识——有了这些知识，计算机就能自己寻求解法。

虽然经典算法和机器学习算法之间的差别大得如同一场革命，但是由经典而入机器学习的过程却是自然而然的。比如《算法的乐趣》中介绍的曲线拟合问题，就是supervised learning（有监督学习），而音乐播放器里常用的傅里叶变换和其他时域频域变换则是unsupervised learning（无监督学习）的技术基础，棋类游戏算法是博弈论和reinforcement learning（强化学习）的经典例子。我常见有朋友从读数学教材开始探索机器学习和人工智能算法，也常看到有人不堪忍受长时间缺乏乐趣的探索以至于半途而废。如果是这样，也许不如从《算法的乐趣》开始这个探索过程。

我曾经以为从乐趣出发阐述算法的书会从西方发芽，没想到先看到了一本中文书。这真超出了我的预料。

王益

LinkedIn高级主任分析师

序 二

当图灵出版社的编辑找到我希望我为这本书写个序的时候，我和旁边的同事调侃了一句：又一本简版的《算法导论》要诞生了。但是我还是下载了附件阅读了这本书，当翻到目录的时候，我的兴趣就被燃起来了，转头和同事说，也许这是一本不错的书。

程序员到底需不需要学习算法？这个问题被争论的次数绝对不亚于“Java是不是最好的语言”“VIM和Emacs谁是最好的编辑器”“程序员是不是需要学习数学”。为了避免陷入这样的争论里，我们先对“算法”一词做个转换定义，什么是算法？下面我举几个我亲身经历的例子。

有一次我们发布了一个APP，在注册时要求用户输入自己的真实姓名，但是粗心的工程师忘记了要求用户填写自己的性别，更可怜的是在欢迎页上面明晃晃地写着“欢迎XXX先生注册XX网”，可是应用已经发布到了App Store，到底怎么办？有一位工程师提出了一个办法，我们根据已有的用户姓名和性别作为训练集，来预测新用户到底是男还是女，为了让这个错误尽快得到修复，我们使用了最简单的朴素贝叶斯分类算法，最终测试集上的预测准确率达到93%，也就是说我们解决了93%用户的体验问题。我把这类算法称为专业类算法，也就是招聘网站上算法工程师要求的算法，例如图像处理工程师、数据挖掘工程师等。

有一次我们有一个相似性搜索的需求，数据量不大，只有几万条的数据记录，没有必要用ES这样的搜索引擎。例如输入“长沙市”，也希望可以找到“我爱湖南长沙”“沙市小吃”等，且不说这个需求是否合理，我们单纯来讨论这个问题的解决方案。工程师实现的第一版是将所有字的组合全部列举出来，然后在数据库里做like操作，性能无法接受。于是我们提出了另一种解决方案：将数据库中的每个词都拆成单字，做成集合，保存在缓存中。接下来只需要对集合做交集操作，以字为单位计算词与词之间的相似性，性能问题一下就解决了。这种解题思路在《编程珠玑》中屡见不鲜，这不足以称为具体的算法，几乎都是在梳理我们的逻辑，训练我们解决问题的能力，我把这类算法称为逻辑类算法，或者技巧类算法。

还有一次，我们有个需求是帮助用户做旅游的行程规划，其中的情况比较复杂，因为除了地理位置之外，还需要包含目的地的过往用户评价、所需耗时、不同城市的住宿花费等。但是如果我们仔细分析，可以基于产品设计去拆分问题。在线部分，我们可以去使用基于路程的最短图路径算法，或者基于价格的贪心算法，也可以在综合排序处为用户选择使用了变形加权的最短图路径算法。在离线部分，由于图的节点和边都较少，可以使用穷举法来为用户找到几种不同类型的最优解。这些算法都是在《算法导论》和《数据结构》中有着详细讨论的算法，书中的每一个

算法和数据结构都是作者多年来抽象总结出的通用思路，我称之为通用类算法。

再说一个最基本的，我们做一个网站允许用户发布状态，在高峰期并发量太大，数据库不堪重负，所以我们需要将用户的插入记录先存入到消息队列中，保证用户的正常使用，然后再落地到MySQL数据库中。大家都会想到选择队列这样一种先入先出的数据结构，这也属于一种算法。

通过上面的几个例子，你会不会觉得你的身边处处都是算法？那么到底什么是算法？我们看看标准的定义：能够对一定规范的输入，在有限时间内获得所要求的输出的一系列指令都叫作算法。这个定义太抽象了，让我们简单来说，算法其实就是解决问题的思路 and 办法。那么从这一点来说，你还会说算法不重要么？

那么为什么还会有很多学生，甚至已经工作了很久的朋友还会说大学学的东西没有意义，算法没有用呢？归根结底是因为大家不知道为什么学，或者说缺乏算法的场景化。我在读大学的时候，经常做一些简单的网站，用到的技术无外乎是最简单的对数据库的增删改查，当时最大的感觉就是算法没有用。后来随着工作的深入，我开始逐步地意识到算法的重要性，逐渐地把算法捡了起来。

这本书给我最大的惊喜是没有像一般的算法书一样单纯地去讲算法和数据结构本身，那样无论语言多风趣，只要一谈到关键的问题也会马上变得无趣起来。作者在每一章都给出了一个实际的问题，然后尝试用算法去解决这一个问题，没有局限于通用类算法，而是同时涵盖逻辑类算法、通用类算法和专业类算法，真正是在训练读者解决问题的能力，而解决问题的能力，正是任何一家公司所需人才的最核心的技能。

另外，我已经在幻想作者在下一本书里可以把工作中的实际场景列举出来，更进一步地讲述工作中的算法，让每一个在校学生都可以意识到算法对于未来工作的重要性，也让每一位从业者拍案叫绝：“原来这个问题可以这样解！”让人人谈算法，人人写算法，引发软件行业的全民算法潮。

黄鑫（飞林沙）
极光推送首席科学家

序 三

如果说《啊哈！算法》是算法界的小白书，内容太少看得不过瘾，那么这本《算法的乐趣》或许可以带你一起牛逼一起飞。当我刚拿到书的目录的时候，我就很期待，因为终于有一本算法书可以系统地和大伙说一说这些我也很想与大伙说的伟大算法。

暴力盲目的搜索算法往往让计算机显得很笨甚至有点痴呆，如果你想设计一个“狡猾”的程序，那么本书中的搜索剪枝、A*寻径、博弈树以及遗传算法等将给你带来启发。快速傅里叶变换，这么霸气而又高大上的名字，其实在我们生活中的应用随处可见，家中的Wi-Fi、智能手机、电话、路由器等几乎所有内置计算机系统的东西都会以各种方式使用这个算法。RLE数据压缩算法，在文档、视频、音乐、数据存储、云计算、数据库等几乎所有应用中都有着广泛的运用。压缩算法令系统更有效，成本更低。再来说密码学算法中非常重要的RSA算法，如果没有这些算法，互联网就会变得不安全，电子交易就不会如此可信。

好玩的算法还有很多很多，历法与二十四节气的计算、华容道、井字棋、黑白棋、五子棋以及俄罗斯方块……你会惊讶地发现，再简单不过的事情背后，都藏着算法的神奇背影。不妨将本书放在案头慢慢品读，你将能看到算法如何深入我们的日常生活，如何重塑我们的世界。

你准备好了吗？接下来，这个世界算法将接管一切。

啊哈磊

《啊哈！算法》作者

致 谢

本书的示例和思考来源于我多年的资料收集和面试题目，旨在通过现实生活中的有趣实例揭示算法的作用。本书来源于我博客中的算法专栏，在写作的过程中，很多人给予了我无私的帮助，在此我要向所有帮助过我的人致以诚挚的感谢。

首先，感谢我的家人给予的无条件的支持，没有他们的理解和鼓励，本书将无法按时完稿。

其次，感谢图灵的各位编辑老师在本书策划和编写过程中给予的指导和帮助，感谢本书的排版老师让书中的图表更加清晰和规范，感谢封面设计师潘建永和书签设计师Sneezry，你们非凡的创意和优秀的设计让这本书锦上添花。感谢王益、黄鑫和纪磊对本书的认同和推荐。

最后，我要感谢本书参考资料的所有作者，我已经尽力寻找所有资料的引用根源，但是仍有可能漏掉一些内容，对于没有提到的名字的作者，我感到十分抱歉，但是仍然要感谢你们。

前言

程序员与算法，这是一个永恒的话题，无论在哪个论坛，只要出现此类主题的帖子，一定会看到两种针锋相对的观点的“激烈碰撞”。其实泡过论坛的人都知道，两种观点“激烈辩论”的惨烈程度往往可以上升到互相问候先人的高度，即使是技术论坛也不例外。在准备此书之前，我在博客的“算法系列”专栏已经陆陆续续地写了有一年多的时间，在此期间，不断有读者问我：“程序员必须会算法吗？”我实在不想让我的博客成为喷满各种口水的是非之地，所以一般不正面回答，只是笼统地说些“各行各业情况都不尽相同”之类的话，避免站队。

程序员对算法通常怀有复杂的感情，算法很重要是大家的共识，但是是否每个程序员都必须学算法是主要的分歧点。本书是想重新定义程序员对算法的理解，并不想通过说教的方式给出到底是学还是不学的结论。很多人可能觉得像人工智能、视频与音频处理以及数据搜索与挖掘这样高大上的内容才能称为算法，往往觉得算法深不可测。但是这些其实都不是具体的算法，而是一系列算法的集合，这里面既有各种大名鼎鼎的算法，比如神经网络、遗传算法、离散傅里叶变换算法以及各种插值算法，也有不起眼的排序和概率计算的算法。你必须深入地了解它们，才会领略到算法的实质——解决问题。忽视这一点，片面地或抽象地理解算法，就会使对算法的理解变得形而上学。在我的博客里就有人留言质疑：“穷举也算是算法？”且不说搜索和枚举是算法的基础设计模式之一，单就那么多的NPC问题（比如著名的汉密尔顿回路问题，至今还没有找到多项式时间的算法），实际上，从只有穷举算法和其他随机搜索算法才能求解这一点看，任何人都不能小看它。

狭隘的算法定义会将自己局限在一个小角落里，从而错过了整个色彩缤纷的算法世界。本书将带你开启一段算法之旅，在这里，你将会看到各种构造算法的基础方法，比如贪婪法、分治法、动态规划法，等等，也可以通过一个个示例看到如何应用这些算法来解决实际问题。通过对“爱因斯坦的思考题”“三个水桶等分水”“妖怪与和尚过河问题”等趣味智力题的计算机求解算法设计，你可以领会到算法设计的三个关键问题，以及对这些问题的处理方法，为以后解决这样的问题提供举一反三的基础。

在生活中，凡是有乐趣的地方就有算法。本书将介绍生活中无处不在的算法。在历法计算的章节里，你会看到霍纳法则（Horner's rule）的使用和求解一元高次方程的牛顿迭代法；音频播放器上跳动的频谱，背后是离散傅里叶变换算法；DOS时代著名的PCX图像文件格式使用的RLE压缩算法是如此简单，但是却非常有效；RSA加密算法的光环之下是朴实的欧几里得算法、蒙哥马利算法和米勒-拉宾算法；华容道游戏求解的简单穷举算法中还蕴藏着对棋盘状态的哈希算法……遗传算法神秘不可测，但是用遗传算法求解0-1背包问题只用了60多行代码。事实上，抛

开对遗传算法的深层次研究和在各种专业领域内的扩展应用,单就算法原理来说,它就是这么简单。深蓝战胜卡斯特罗之后,人类棋手在与计算机的博弈中就完全处于下风,人工智能真的这么神奇?人工智能确实是个神奇的领域,但就计算机下棋这件事来说,却并不怎么神奇,算法的基本原理简单得让人难以置信,看看第23章你就知道了。

算法之大,大到可以囊括宇宙万物的运行规律,算法之小,小到寥寥数行代码即可展现一个神奇的功能。算法是琐碎的,以至于常常被人们忽视,然而忽视算法能力的培养所带来的代价是巨大的,第1章介绍的环形队列的例子就是一个最好的说明。我面试过很多求职者,我常常会让他们手写一个算法,我的题目是这样的:有一个由若干正整数组成的数列,数列中的每个数都不超过32,已知数列中存在重复的数字,请给出一个算法找出这个数列中所有重复出现的数。我期望求职者给我一个正确的算法实现,接下来我会问这个算法的时间复杂度是什么,有没有考虑过存在一个 $O(n)$ 时间复杂度的算法。大部分求职者都知道自己的算法是 $O(n^2)$ 时间复杂度,但是都否认存在 $O(n)$ 时间复杂度的算法。事实上这个题目是可以有 $O(n)$ 时间复杂度的算法的,因为大家都忽略了一个重要的条件。这个题目并不难,但是仍有将近三分之一的面试者无法给出正确的算法,有的甚至还给我一张白纸。有人犯错误是正常现象,但是让我意外的是居然有三分之一的人写不出这个算法,算法设计的基本功被无视到这种地步是不正常的。

程序员谈到算法言必称一些高大上的词汇,但是这些专有名词大部分人是用不到的,以至于人们常常认为算法不过如此,不会又如何?这种思想变得极端就会让人忽视算法的基础设计能力,这才是最要命的。在我们维护的网络设备上,用户的数据关系错综复杂,一个对线性表进行二重循环都想不到的人又怎么可能会维护这些数据?我希望程序员们提高基础的算法能力,先从培养兴趣开始或许是一个不错的切入点。

本书挑选的算法例子,都围绕着“趣”字展开,都是简单且在生活中常见的算法,可能有些是你还没有意识到的。我上学的时候曾经做过一个MP3播放器程序,你可能觉得这主要就是利用一些音频解码算法吧?是的,这个为主要部分,但是一个功能完整的播放程序还用了很多你想不到的算法:为增加频谱显示和均衡器功能,使用了离散傅里叶变换算法;为计算频率功率谱,使用了加权平均值算法;为了匹配硬件输出设备与解码算法的性能差异,需要一个有多个缓冲区的队列管理音频数据块,这就引入了滑动窗口算法;为提供按照专辑名称或作者名称排序功能,使用了快速排序算法;为了平滑均衡器调节对音频的影响,使用了三次样条曲线插值算法;为了在两首歌曲之间切换时压制刺耳的杂音(通过填充一些舒适噪声的方式实现),还使用了正弦信号发生器算法。这些你都没有想到吧?其实还有更多的例子,比如大型项目管理软件中的工作节点排序功能和关键路径功能,背后支撑它们的却是简单的有向图拓扑排序算法。这是不是很有趣?生活中处处都是算法,程序员又怎么可能与算法绝缘?

再次重申一点,本书没有任何关于算法重要性的说教,当你看到本书时,我希望你的表情是“啊哈,原来如此!”,或者是“嗯,有意思!”,并从中获得乐趣。本书几乎所有章节都有相关算法实现和功能演示的代码,读者可以到我的博客(<http://blog.csdn.net/orbit/>)中下载,也可以到图灵社区本书主页(www.it-ebooks.com/book/1604)下载使用。

目 录

第 1 章 程序员与算法	1	3.2 分治法	30
1.1 什么是算法	2	3.2.1 分治法的基本思想	30
1.2 程序员必须要会算法吗	2	3.2.2 递归和分治，一对好朋友	31
1.2.1 一个队列引发的惨案	3	3.2.3 分治法的例子：大整数 Karatsuba 乘法算法	32
1.2.2 我的第一个算法	5	3.3 动态规划	34
1.3 算法的乐趣在哪里	7	3.3.1 动态规划的基本思想	34
1.4 算法与代码	8	3.3.2 动态规划法的例子：字符串 的编辑距离	37
1.5 总结	9	3.4 解空间的穷举搜索	40
1.6 参考资料	9	3.4.1 解空间的定义	41
第 2 章 算法设计的基础	10	3.4.2 穷举解空间的策略	42
2.1 程序的基本结构	10	3.4.3 穷举搜索的例子：Google 方 程式	44
2.1.1 顺序执行	10	3.5 总结	46
2.1.2 循环结构	11	3.6 参考资料	46
2.1.3 分支和跳转结构	13	第 4 章 阿拉伯数字与中文数字	47
2.2 算法实现与数据结构	16	4.1 中文数字的特点	47
2.2.1 基本数据结构在算法设计中的 应用	16	4.1.1 中文数字的权位和小节	48
2.2.2 复杂数据结构在算法设计中的 应用	19	4.1.2 中文数字的零	48
2.3 数据结构和数学模型与算法的关系	24	4.2 阿拉伯数字转中文数字	49
2.4 总结	25	4.2.1 一个转换示例	49
2.5 参考资料	25	4.2.2 转换算法设计	49
第 3 章 算法设计的常用思想	26	4.2.3 算法实现	50
3.1 贪婪法	26	4.2.4 中文大写数字	51
3.1.1 贪婪法的基本思想	27	4.3 中文数字转阿拉伯数字	52
3.1.2 贪婪法的例子：0-1 背包问题	27	4.3.1 转换的基本方法	52

4.3.2 算法实现	52	7.3.1 穷举所有的完美匹配	81
4.4 数字转换的测试用例	54	7.3.2 不稳定因素的判断算法	82
4.5 总结	55	7.3.3 穷举的结果	84
4.6 参考资料	55	7.4 二部图与二分匹配	84
第5章 三个水桶等分8升水的问题	56	7.4.1 最大匹配与匈牙利算法	85
5.1 问题与求解思路	57	7.4.2 带权匹配与 Kuhn-Munkres 算法	88
5.2 建立数学模型	58	7.5 总结	93
5.2.1 状态的数学模型与状态树	58	7.6 参考资料	94
5.2.2 倒水动作的数学模型	59	第8章 爱因斯坦的思考题	95
5.3 搜索算法	60	8.1 问题的答案	96
5.3.1 状态树的遍历	60	8.2 分析问题的数学模型	96
5.3.2 剪枝和重复状态判断	61	8.2.1 基本模型定义	96
5.4 算法实现	62	8.2.2 线索模型定义	98
5.5 总结	64	8.3 算法设计	99
5.6 参考资料	64	8.3.1 穷举所有的组合结果	99
第6章 妖怪与和尚过河问题	65	8.3.2 利用线索判定结果的正确性	101
6.1 问题与求解思路	66	8.4 总结	103
6.2 建立数学模型	66	8.5 参考资料	104
6.2.1 状态的数学模型与状态树	67	第9章 项目管理与图的拓扑排序	105
6.2.2 过河动作的数学模型	67	9.1 AOV网和AOE网	107
6.3 搜索算法	69	9.2 拓扑排序	108
6.3.1 状态树的遍历	70	9.2.1 拓扑排序的基本过程	108
6.3.2 剪枝和重复状态判断	70	9.2.2 按照活动开始时间排序	108
6.4 算法实现	71	9.3 关键路径算法	111
6.5 总结	72	9.3.1 什么是关键路径	112
6.6 参考资料	73	9.3.2 计算关键路径的算法	113
第7章 稳定匹配与舞伴问题	74	9.4 总结	116
7.1 稳定匹配问题	74	9.5 参考资料	116
7.1.1 什么是稳定匹配	74	第10章 RLE压缩算法与PCX图像 文件格式	117
7.1.2 Gale-Shapley 算法原理	75	10.1 RLE 压缩算法	117
7.2 Gale-Shapley 算法的应用实例	77	10.1.1 连续重复数据的处理	117
7.2.1 算法实现	77	10.1.2 连续非重复数据的处理	118
7.2.2 改进优化: 空间换时间	80		
7.3 有多少稳定匹配	81		

10.1.3 算法实现	118	11.4.2 中国农历的推算	157
10.2 RLE 与 PCX 图像文件格式	121	11.4.3 一个简单的“年历”	165
10.2.1 PCX 图像文件格式	121	11.5 总结	166
10.2.2 PCX_RLE 算法	122	11.6 参考资料	167
10.2.3 256 色 PCX 文件的解码和 显示	123		
10.3 总结	124	第 12 章 实验数据与曲线拟合	168
10.4 参考资料	125	12.1 曲线拟合	168
第 11 章 算法与历法	126	12.1.1 曲线拟合的定义	168
11.1 格里历(公历)生成算法	126	12.1.2 简单线性数据拟合的例子	168
11.1.1 格里历的历法规则	126	12.2 最小二乘法曲线拟合	169
11.1.2 今天星期几	127	12.2.1 最小二乘法原理	170
11.1.3 生成日历的算法	131	12.2.2 高斯消元法求解方程组	171
11.1.4 日历变更那点事儿	132	12.2.3 最小二乘法解决“速度与 加速度”实验	172
11.2 二十四节气的天文学计算	134	12.3 三次样条曲线拟合	173
11.2.1 二十四节气的起源	134	12.3.1 插值函数	174
11.2.2 二十四节气的天文学定义	135	12.3.2 样条函数的定义	174
11.2.3 VSOP-82/87 行星理论	137	12.3.3 边界条件	175
11.2.4 误差修正——章动	141	12.3.4 推导三次样条函数	176
11.2.5 误差修正——光行差	143	12.3.5 追赶法求解方程组	179
11.2.6 用牛顿迭代法计算二十四节 气	144	12.3.6 三次样条曲线拟合算法实 现	181
11.3 农历朔日(新月)的天文学计算	146	12.3.7 三次样条曲线拟合的效果	183
11.3.1 日月合朔的天文学定义	147	12.4 总结	184
11.3.2 ELP-2000/82 月球理论	147	12.5 参考资料	184
11.3.3 误差修正——地球轨道离心 率修正	149	第 13 章 非线性方程与牛顿迭代法	185
11.3.4 误差修正——黄经摄动	149	13.1 非线性方程求解的常用方法	185
11.3.5 月球地心视黄经和最后的 修正——地球章动	150	13.1.1 公式法	185
11.3.6 用牛顿迭代法计算日月合 朔	151	13.1.2 二分逼近法	186
11.4 农历的生成算法	152	13.2 牛顿迭代法的数学原理	187
11.4.1 中国农历的起源与历法规 则	153	13.3 用牛顿迭代法求解非线性方程的 实例	188
		13.3.1 导函数的求解与近似公式	188
		13.3.2 算法实现	188
		13.4 参考资料	189

第 14 章 计算几何与计算机图形学	190
14.1 计算几何的基本算法	190
14.1.1 点与矩形的关系	190
14.1.2 点与圆的关系	191
14.1.3 矢量的基础知识	191
14.1.4 点与直线的关系	194
14.1.5 直线与直线的关系	194
14.1.6 点与多边形的关系	196
14.2 直线生成算法	199
14.2.1 什么是光栅图形扫描转换	200
14.2.2 数值微分法	200
14.2.3 Bresenham 算法	202
14.2.4 对称直线生成算法	204
14.2.5 两步算法	205
14.2.6 其他直线生成算法	207
14.3 圆生成算法	207
14.3.1 圆的八分对称性	208
14.3.2 中点画圆法	209
14.3.3 改进的中点画圆法—— Bresenham 算法	210
14.3.4 正负判定画圆法	211
14.4 椭圆生成算法	212
14.4.1 中点画椭圆法	213
14.4.2 Bresenham 椭圆算法	215
14.5 多边形区域填充算法	217
14.5.1 种子填充算法	218
14.5.2 扫描线填充算法	223
14.5.3 改进的扫描线填充算法	229
14.5.4 边界标志填充算法	233
14.6 总结	236
14.7 参考资料	236
第 15 章 音频频谱和均衡器与傅里叶 变换算法	237
15.1 实时频谱显示的原理	237
15.2 离散傅里叶变换	238
15.2.1 什么是傅里叶变换	239
15.2.2 傅里叶变换原理	239
15.2.3 快速傅里叶变换算法的 实现	243
15.3 傅里叶变换与音频播放的实时频谱 显示	245
15.3.1 频域数值的特点分析	245
15.3.2 从音频数据到功率频谱	246
15.3.3 音频播放时实时频谱显示的 例子	248
15.4 破解电话号码的小把戏	251
15.4.1 拨号音的频谱分析	251
15.4.2 根据频谱数据反推电话 号码	252
15.5 离散傅里叶逆变换	253
15.5.1 快速傅里叶逆变换的推导	254
15.5.2 快速傅里叶逆变换的算法 实现	254
15.6 利用傅里叶变换实现频域均衡器	255
15.6.1 频域均衡器的实现原理	255
15.6.2 频域信号的增益与衰减	256
15.6.3 均衡器的实现——仿 Foobar 的 18 段均衡器	258
15.7 总结	259
15.8 参考资料	259
第 16 章 全局最优解与遗传算法	260
16.1 遗传算法的原理	260
16.1.1 遗传算法的基本概念	261
16.1.2 遗传算法的处理流程	262
16.2 遗传算法求解 0-1 背包问题	267
16.2.1 基因编码和种群初始化	267
16.2.2 适应度函数	268
16.2.3 选择算子设计与轮盘赌算 法	268
16.2.4 交叉算子设计	270
16.2.5 变异算子设计	271
16.2.6 这就是遗传算法	272

16.3 总结	272	18.3.4 数据解密算法实现	301
16.4 参考资料	273	18.4 RSA 签名与身份验证	302
第 17 章 计算器程序与大整数计算	274	18.4.1 RSASSA-PKCS 与 RSASSA-PSS 签名填充模式	302
17.1 哦, 溢出了, 出洋相的计算器程序	274	18.4.2 签名算法实现	304
17.2 大整数计算的原理	275	18.4.3 验证签名算法实现	305
17.2.1 大整数加法	276	18.5 总结	305
17.2.2 大整数减法	278	18.6 参考资料	306
17.2.3 大整数乘法	279	第 19 章 数独游戏	307
17.2.4 大整数除法与模	281	19.1 数独游戏的规则与技巧	307
17.2.5 大整数乘方运算	282	19.1.1 数独游戏的规则	307
17.3 大整数类的使用	283	19.1.2 数独游戏的常用技巧	308
17.3.1 与 Windows 的计算器程序 一决高下	283	19.2 计算机求解数独问题	308
17.3.2 最大公约数和最小公倍数	284	19.2.1 建立问题的数学模型	310
17.3.3 用扩展欧几里得算法求模 的逆元	286	19.2.2 算法实现	311
17.4 总结	288	19.2.3 与传统穷举方法的结果对 比	312
17.5 参考资料	288	19.3 关于数独的趣味话题	312
第 18 章 RSA 算法——加密与签名	289	19.3.1 数独游戏有多少终盘	313
18.1 RSA 算法的开胃菜	289	19.3.2 史上最难数独游戏	314
18.1.1 将模幂运算转化为模乘运 算	290	19.4 总结	314
18.1.2 模乘运算与蒙哥马利算法	291	19.5 参考资料	315
18.1.3 模幂算法	292	第 20 章 华容道游戏	316
18.1.4 素数检验与米勒-拉宾算 法	292	20.1 华容道游戏介绍	316
18.2 RSA 算法原理	295	20.2 自动求解的算法原理	317
18.2.1 RSA 算法的数学理论	295	20.2.1 定义棋盘的局面	317
18.2.2 加密和解密算法	296	20.2.2 算法思路	319
18.2.3 RSA 算法的安全性	297	20.3 自动求解的算法实现	320
18.3 数据块分组加密	297	20.3.1 棋局状态与 Zobrist 哈希算 法	321
18.3.1 字节流与大整数的转换	298	20.3.2 重复棋局和左右镜像的处 理	323
18.3.2 PKCS 与 OAEP 加密填充 模式	298	20.3.3 正确结果的判断条件	325
18.3.3 数据加密算法实现	300	20.3.4 武将棋子的移动	325
		20.3.5 棋局的搜索算法	328

20.4 总结	329	23.1.3 负极大值搜索算法	362
20.5 参考资料	329	23.1.4 “ α - β ”剪枝算法	363
第 21 章 A*寻径算法	330	23.1.5 估值函数	365
21.1 寻径算法演示程序	330	23.1.6 置换表与哈希函数	366
21.2 Dijkstra 算法	331	23.1.7 开局库与终局库	368
21.2.1 Dijkstra 算法原理	332	23.2 井字棋——最简单的博弈游戏	368
21.2.2 Dijkstra 算法实现	332	23.2.1 棋盘与棋子的数学模型	369
21.2.3 Dijkstra 算法演示程序	333	23.2.2 估值函数与估值算法	370
21.3 带启发的搜索算法——A*算法	335	23.2.3 如何产生走法(落子 方法)	371
21.3.1 A*算法原理	336	23.3 奥赛罗棋(黑白棋)	373
21.3.2 常用的距离评估函数	337	23.3.1 棋盘与棋子的数学模型	374
21.3.3 A*算法实现	340	23.3.2 估值函数与估值算法	377
21.4 总结	342	23.3.3 搜索算法实现	380
21.5 参考资料	342	23.3.4 最终结果	384
第 22 章 俄罗斯方块游戏	343	23.4 五子棋	385
22.1 俄罗斯方块游戏规则	343	23.4.1 棋盘与棋子的数学模型	386
22.2 俄罗斯方块游戏人工智能的算法原 理	344	23.4.2 估值函数与估值算法	388
22.2.1 影响评价结果的因素	345	23.4.3 搜索算法实现	391
22.2.2 常用的俄罗斯方块游戏人 工智能算法	346	23.4.4 最终结果	393
22.2.3 Pierre Dellacherie 评估算 法	347	23.5 总结	393
22.3 Pierre Dellacherie 算法实现	349	23.6 参考资料	393
22.3.1 基本数学模型和数据结构 定义	350	附录 A 算法设计的常用技巧	395
22.3.2 算法实现	352	A.1 数组下标处理	395
22.4 总结	358	A.2 一重循环实现两重循环的功能	396
22.5 参考资料	358	A.3 棋盘(迷宫)类算法方向遍历	396
第 23 章 博弈树与棋类游戏	359	A.4 代码的一致性处理技巧	397
23.1 棋类游戏的 AI	359	A.5 链表和数组的配合使用	398
23.1.1 博弈与博弈树	360	A.6 “以空间换时间”的常用技巧	399
23.1.2 极大极小值搜索算法	361	A.7 利用表驱动避免长长的 switch-case	400
		附录 B 一个棋类游戏的设计框架	401
		B.1 代码框架的整体结构	401
		B.2 代码框架的使用方法	403