TURING 图灵程序设计丛书

Programmation efficace

高效算法

竞赛、应试与提高必修128例

[法] Christoph Dürr Jill-Jênn Vie —— 著

史世强 —— 译

中国工信出版集团

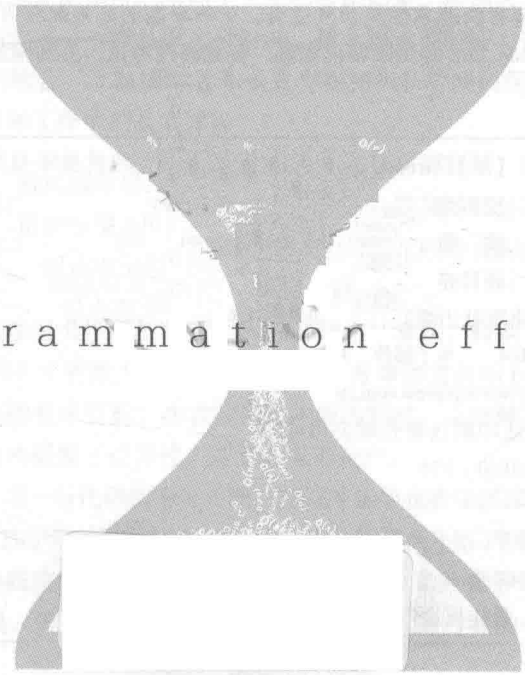
人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

高效算法

竞赛、应试与提高必修128例

[法] Christoph Dürr Jill-Jênn Vie —— 著
史世强 —— 译



Programmation efficace

人民邮电出版社
北京

图书在版编目（CIP）数据

高效算法：竞赛、应试与提高必修128例 /（法）克里斯托弗·杜尔（Christoph Dürr），（法）吉尔-让·维（Jill-Jênn Vie）著；史世强译. -- 北京：人民邮电出版社，2018.5

（图灵程序设计丛书）

ISBN 978-7-115-48085-9

Ⅰ. ①高… Ⅱ. ①克… ②吉… ③史… Ⅲ. ①计算机算法－研究 Ⅳ. ①TP301.6

中国版本图书馆CIP数据核字（2018）第050544号

内 容 提 要

本书旨在探讨如何优化算法效率，详细阐述了经典算法和特殊算法的实现、应用技巧和复杂度验证过程，内容由浅入深，能帮助读者快速掌握复杂度适当、正确率高的高效编程方法以及自检、自测技巧，是参加ACM/ICPC、Google Code Jam等国际编程竞赛、备战编程考试、提高编程效率、优化编程方法的参考书目。

◆ 著　[法] Christoph Dürr　Jill-Jênn Vie

译　史世强

责任编辑　戴　童

责任印制　周昇亮

◆ 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

三河市祥达印刷包装有限公司印刷

◆ 开本：800×1000　1/16

印张：12.75

字数：301千字　2018年5月第1版

印数：1-3 500册　2018年5月河北第1次印刷

著作权合同登记号　图字：01-2017-3131号

定价：55.00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：京东工商广登字20170147号

译者序

22 年前的秋天，我刚刚进入初中时，得到了一台中华学习机。它的 1 MHz 主频甚至赶不上现在一台 10 元钱的计算器。我从第一行用 BASIC 语言写的 IF/ELSE 开始，开启了自己的编程人生。1996 年，还是初中生的我凭着不多的算法和逻辑知识参加了国家信息学奥林匹克竞赛，当然，最后只得到了安慰奖。二十多年后，我得知当年斩获金牌的是王小川，如今搜狗的 CEO。

现在，我在一家互联网公司负责技术并管理研发团队。从自身的职业发展经历，以及在中国和法国的招聘和用人经历中，我深刻体会到了软件工程师的成就在很大程度上取决于他的专业知识视野。这是个很现实的问题。因此，我在得到翻译这本法语技术书的机会时，欣然接受了这个颇有难度的任务。

法国是一个盛产数学家的国度。不同于大家的传统印象，法国人在“浪漫”的同时，在工作和科研中非常讲究逻辑与验证——产品原型要验证，技术探索要验证。证明和实验有着同样不可或缺的地位。理论和实践的结合，让法国学界和企业界在相当长时间内保持着旺盛的生命力与创造力。这是我在法国 8 年学习和工作中的真实体验。

本书由法国国际信息学奥林匹克竞赛“国家队”辅导老师编写，凝聚了作者辅导高中生、大学生参加国际信息学奥林匹克竞赛的大量经验和技巧。书中提及的部分算法十分常见，在实际工作中也十分常用。但也有另一部分算法，例如舞蹈链算法以及一些涉及图论与匹配的算法，在中国的大学教育都不太提及。

在人工智能和深度学习大发展的今天，Python 语言、算法，特别是证明算法可靠性和高效性的能力，是进入大数据和人工智能人才市场的入场券。希望读者善用 Github 和作者准备的源代码网站，以及网上能够找到的技术资源，在尝试代码实现的同时，去理解算法复杂度的证明过程，从而彻底掌握并熟练运用这些凝聚了很多代人智慧的无形资产。

我要感谢教我写下第一行代码的哥哥史轶，支持并指导我参加国家信息学奥林匹克竞赛的湖北省十堰市东风汽车公司第四中学的陈长国老师，用大量课外知识开拓了我的见识的东风汽车公司第一中学的吴华山老师，支持我前往法国留学的父母，以及一直以来给我带来太多快乐的妻子和孩子。

由于个人水平有限，译文不能做到尽善尽美，欢迎读者通过我的个人网站 www.jetwaves.cn 与我交流。

史世强

2017 年 10 月于巴黎

序

我们编写本书的主要动力是对 Python 语言编程的热爱和对解决算法问题的激情。Python 语言能够如此打动人，是因为这种语言能让我们编写清晰而优雅的代码，把注意力集中于算法的本质步骤，而不需要过多关注复杂的语法和数据结构。同时，我们用 Python 完成编写程序后数个月再回头来读的时候，仍然可以理解自己写的代码，这一点十分有教益。作为本书的作者，我们最希望的是能接受新的挑战，其次是能经得住各种测试，因为一段程序代码只有在毫无 bug 地实现后，我们才算真正地掌握了编程技巧。我们希望用自己的热情感染读者，营造出一种氛围，鼓励大家学习和掌握扎实的算法和编程基础知识。这种学习经历往往会受到大型软件企业招聘人员的赏识，而对于软件工程师或计算机科学教育工作者来说，这对其整个职业生涯也会有所帮助。

本书按照主题而不是技术分类收录了 128 种算法。其中某些算法是常见的经典算法，另一些则不太常见。尤其在读者备战 ACM-ICPC、Google Code Jam、Facebook Hacker Cup、Prologin 和 France-ioi 等编程竞赛时，本书编写的大量问题将起到积极的辅导作用。我们希望本书能够成为算法的基础教程和高级程序设计教程的参考，或者能让学习数学和计算机专业的读者看到与众不同的进修内容。读者可以在网站 tryalgo.org（http://tryalgo.org/code/）上找到本书使用的源代码库[①]，以及用来测试代码调试结果和实现性能的链接。

感谢 Huong 和智子，如果没有这两位朋友的支持，本书是无法完成的。感谢法国综合理工学院和法国高等师范学院 Cachan 分校的学生们，他们多次通宵达旦的训练，为本书提供了很多素材。最后，感谢所有审阅手稿的朋友们，他们是 René Adad、Evripidis Bampis、Binh-Minh Bui-Xuan、Stéphane Henriot、Lê Thành Dũng Nguyễn、Alexandre Nolin 和 Antoine Pietri。本书的作者之一要特别感谢在 Tiers 高中时的老师 Yves Lemaire 先生：当年就是在这位老师的启迪下，作者才初次发现了本书 2.5 节中描述的“宝藏”。

最后，我们希望读者在碰到算法难题时，能够耐心地花时间去思考。祝愿大家能在豁然间找到解答，甚至是一个优雅的解答，享受到胜利的喜悦之情。

好，我们要开始了！

① 也可以用 PyPI 直接安装后下载查看并执行。——译者注

目录

第1章 引言

年轻人，通过本书学习编写算法，你将在编程竞赛中大显身手，顺利通过就业面试，卷起袖管大干一场，创造更多的价值。

如今人们仍然存在一种误解，错把程序员当成当代的魔术师。计算机逐渐进入企业和家庭，成为推动世界运行的重要动力。但是，仍有太多人在使用计算机的时候没能掌握足够的知识，充分发挥计算机的能力，来满足自己的需要。懂得编程可以让人们在最大程度上找到解决问题的高效方法。算法和编程成为计算机行业中必不可少的工具。掌握这些技能可以让我们在面对困难时提出有创造力、高效的解决方案。

本书介绍了多种解决某些经典问题的算法技术，描述了问题出现的场景，并用 Python 提出了简单的解决方案。正确地实现算法往往不是一件简单的事情，总需要避开陷阱，也需要应用一些技巧保证算法能够在规定时间内实现。本书在阐述算法实现时附加了重要的细节，以帮助读者理解。

最近几十年，不同级别的编程竞赛在世界范围内展开，推广了算法文化。竞赛考察的问题一般都是经典问题的变种，隐藏在难以破解的谜面背后，让参赛者们一筹莫展。

1.1 编程竞赛

在编程竞赛中，参赛者必须在规定时间内解决多个问题。问题的输入称为实例（instance）。举个例子，一个输入实例可以是最短路径问题中图的邻接矩阵。一般来讲，问题会给出一个输入实例和它的输出结果[①]。参赛者在网上将答案的源代码提交到服务器；之后，服务器的后台进程将编译并

① 用于展示思路和代码测试。——译者注

执行代码，而后测试对错。对于某些问题，源代码在执行时会被输入多个实例，并一一执行；而对于其他问题，每次执行源代码时，输入都从一个表示实例数量的整数开始。程序必须按顺序读取每个输入实例，解决问题，并输出结果。如果程序能够在指定时间内输出正确结果，那么提交的答案就可以被接受。

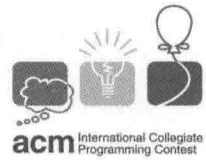


图 1.1 ACM 竞赛的图标形象地展示了解决问题的步骤。参赛团队每解决一个问题时，就会得到一个吹起的气球

我们无法列出世上所有的编程竞赛名称和竞赛网址。就算有可能，这个列表也会很快过时。但无论如何，我们在这里还是要简单介绍一下最重要的几个编程竞赛。

- ACM/ ICPC 编程竞赛

这是历史最悠久的竞赛，由国际计算机协会（ACM）从 1977 年开始举办。竞赛称为“国际大学生程序设计竞赛”（ICPC），以巡回赛的方式进行。比如，巡回赛在法国站的起点是西南欧洲地区竞赛（SWERC）。地区竞赛的前两名有资格进入全球决赛。这个竞赛的特点是每队由 3 位成员组成，共用一台计算机。参赛队在 5 个小时内从 10 个问题中尝试挑战解决尽可能多的问题。排名的第一个依据是答案被接受的数量（答案会被不公开的用例来测试）；排名的第二个依据是解决问题所耗费的时间，耗时以开始解题到提交答案的时长为准。提交一个错误答案会被罚时 20 分钟。

组成一个优秀团队有很多种方式。一般来说，至少需要一位优秀的程序员和一位优秀的数学家，以及一位擅长不同领域的专家，比如图论、动态规划等。他们需要在承受巨大压力的前提下通力合作。在竞赛中，参赛者可以用 8 磅字体打印 25 页的源代码作为参考。参赛者还可以访问 Java 应用程序编程接口（API）的在线文档，以及 C++ 的在线标准库文档。

- Google Code Jam 编程竞赛

国际计算机协会的编程竞赛仅限硕士及以下学历的学生参加，与此不同的是，Google Code Jam 编程竞赛对所有人开放。竞赛每年一度，举办历史较短，而且仅限个人参赛。每个问题通常会包含一系列简单实例，解答这些实例就可以得到一定的分数。同时，问题还包含一系列步骤复杂的实例，这需要真正找到拥有合适复杂度的算法来解决。直到竞赛结束，参赛者才能得知步骤较复杂的实例是否最终被接受了。这个竞赛的优势在于，参赛者在竞赛结束后可以查阅其他参赛者提交的解决方案，这种方式有非常强的指导作用。Facebook Hacker Cup 编程竞赛也采取类似形式。

- Prologin 编程竞赛

法国每年为 20 岁以下的学生举办一场 Prologin 编程竞赛。竞赛过程分为三步——在线筛选、

地区赛和决赛，以考察参赛者解决算法问题的能力。最终决赛是一场不同寻常的 36 小时竞赛，参赛者需要解决一个人工智能问题。每个参赛者必须编写一个遵循组织者设定规则的游戏程序，然后以循环赛的形式让游戏程序彼此对决，以此来决定参赛者的成绩排名。竞赛官网上 prologin.org 对此有详尽的解释，我们也可以在这里测试自己的算法。

- **France-ioi 编程竞赛**

France-ioi 协会旨在辅助法国初中生和高中生准备国际信息学奥林匹克竞赛。从 2012 年起，协会每年举办“河狸计算机科学竞赛”（竞赛的吉祥物是一只河狸），从初中一年级到高中三年级的学生均可参加。2014 年，全法国有 22.8 万名参赛者。协会官网 france-ioi.org 汇集了 1000 多个有代表性的现象级算法题。

除了上述竞赛以外，也有大量以筛选求职者为目的举行的编程竞赛。比如 TopCoder 网站不仅进行测试，也会对算法进行详细解释，有时讲解质量极高。如果读者希望训练编程能力，我们特别推荐 Codeforces，这是一个备受竞赛群体推崇的网站，对问题的解释总是清晰而仔细。

1.1.1　线上学习网站

很多网站提供历年各大竞赛真题，并可在线测试答案，供大家学习训练。Google Code Jam 编程竞赛和 Prologin 编程竞赛的官网也提供此类功能。但是，ACM/ICPC 每年的竞赛题目却没有统一归纳。

- **传统的线上训练和裁判网站**

下列网站 uva.onlinejudge.org、icpcarchive.ecs.baylor.edu 和 livearchive.onlinejudge.org 总结了大量的 ACM/ICPC 编程竞赛的试题和答案。

- **中国的线上训练和裁判网站**

中国目前有很多线上训练算法能力的网站，比如北京大学的 poj.org、天津大学的 acm.tju.edu.cn 和浙江大学的 acm.zju.edu.cn。相对于其他网站，这些网站更注重训练功能。

- **高级语言算法的训练和裁判网站**

spoj.com（Sphere Online Judge）网站接受用户使用更多种编程语言提交问题的解决方案，其中包括 Python。

在本书配套网站 tryalgo.com 中，读者可以找到应用本书各章讲解的知识和技巧来解决的问题，在实践中检验从书中学到的算法知识。

编程竞赛主要使用的编程语言是 C++ 和 Java 语言。Google Code Jam 编程竞赛接受所有编程语言，因为解题过程是参赛者在本地开发环境中完成的。除此之外，上面提到的线上训练和裁判网站 SPOJ 也接受 Python 语言的解答方案。为了解决因编程语言不同而导致的程序执行时间的差异问题，线上训练和裁判网站对使用不同编程语言的解答方案给出了不同的时间限制。但是，这种平衡策略并不总是准确的，而且，用 Python 语言完成的解题方案经常不能被正确执行。我们希望这种

情况在未来几年能够有所改善。某些线上训练和裁判网站仍在使用 Java 语言的老版本，导致有些很实用的类无法使用，如 Scanner 类。读者在使用这些网站的时候，应当注意版本兼容问题。

1.1.2 线上裁判的返回值

当一段代码被提交给线上训练和裁判网站的时候，会被一系列不公开的测试用例测试，测试结果和一段简要的返回值会反馈给提交者。返回代码有以下几种。

- Accepted：已接受状态

  你提交的代码在指定时间内给出了正确的结果，祝贺你！

- Presentation Error：展示错误

  程序基本能够被接受，但显示了过多或过少的空格或换行符。这种返回码很少出现。

- Compilation Error：编译错误

  你的程序在编译过程中出了错。一般来说，当你点击这条返回码的时候，就能得到错误的详细信息。你应当比较一下，裁判和自己使用的编译器版本是否有所不同。

- Wrong Answer：错误答案

  重新读一遍题吧，你肯定漏掉了什么细节。你是否确定已经检查了所有的边界条件？你是否在代码中遗留了调试代码？

- Time Limit Exceeded：执行超时

  你的解答方案可能没有达到足够优化的实现效率，或者代码的某个角落里藏着一个死循环。检查循环变量，确保循环能够终止。使用一个大规模的复杂测试用例在本地执行测试，确保你的代码性能。

- Runtime Error：运行时错误

  一般来说，这种错误源于分母为 0 的除法运算、数组下标越界，或者对一个空的堆执行了 `pop()` 方法。其他情况也会产生这条错误提示，比如在使用 Java 语言的解答方案中使用了 `assert` 断言，这种方式在编程竞赛中一般是不被接受的。

除了以上有明确意义的返回代码，没有返回代码的情况也能够或多或少地提供一些信息，帮助查找错误。以下是一个 ACM/ICPC/SWERC 竞赛中的真实案例。在一道关于图的题目中，明确指出了输入数据是连通图，但某个参赛团队对此信息不太确定，于是编写了一个测试连通性的方法：当这个方法返回 `true` 结果，即输入为连通图时，程序会进入死循环（返回执行超时错误）；而当这个方法返回 `false` 结果，即输入为非连通图时，程序会执行一个分母为 0 的除法（返回运行时错误）。这种方法可以帮助参赛者探测到某些测试用例输入的图并不是题目中的连通图，从而避免错误。[①]

① 这种方法的目的是在程序中故意留一些缺陷，从而通过返回值来猜测输入数据的具体情况。——译者注

1.2　我们的选择：Python

鉴于 Python 编程语言的可读性和使用的简易性，本书选用它来描述算法。在工业领域，Python 通常用于制作程序的原型。Python 也用于如 SAGE 这类重要的项目系统，因为其中的核心内容大多用实现速度快很多的语言编写，如 C 或 C++。

现在我们说说 Python 编程语言的一些细节。在 Python 中有四个基本数据类型：布尔型、整型、浮点型和字符串。与其他大多数的编程语言不同，Python 中的整数不受数字占用的二进制位数限制，而使用高精度计算方式。

Python 中的高级数据类型包括字典（dictionary）、列表（list）和元组（tuple）。列表和元组的区别是，元组是不可变数据，因此可以用作字典中键值对数据的键。

网络上有很多 Python 的入门教程，如官网 python.org。David Eppstein 创建了一个名为“Python 算法和数据结构”（PADS）的元件库，其中也有很好的讲解。

在编写本书代码的过程中，我们遵循了 PEP8 规范。该规范细致地规定了空格的使用方法、变量命名规则，等等。我们建议读者也遵循上述规范。

- **Python 2 还是 Python 3？**

Python 3.x 版本已经于 2008 年发布。但直到今天，由于仍有大量的类库没有迁移到 Python 3.x，使得许多开发工作还继续停留在 Python 2.x 版本。尽管如此，我们仍然选择使用 Python 3.x 来实现算法。Python 2.x 和 Python 3.x 对本书中代码的主要影响在于 print 语句的使用方式，以及整数除法的使用方式。在 Python 3.x 中，对于两个整数 *a* 和 *b*，表达式 *a*/*b* 会返回除法的浮点型的商，表达式 *a*//*b* 返回的则是两者的欧几里得商，即商的整数部分。`print` 的用法区别在于，在 Python 2.x 中 `print` 是语句，而在 Python 3.x 中 `print()` 是需要使用括号包围的参数来调用的函数。

如果程序运行存在性能问题，可以考虑使用 pypy 或 pypy3 解释器来执行，因为这都是实时编译器。也就是说，Python 代码会先被翻译为机器码，然后才被干净而迅速执行。但 pypy 的弱点在于它仍处在开发过程中，很多 Python 类库尚无法支持。

- **无穷**

Python 使用高精度计算方式进行计算，而不用二进制位数来限制整数的大小。所以，在 Python 语言中不存在哪个数可以指代正无穷大或负无穷大的值。但对于浮点数，我们可以用 `float('inf')` 和 `float('-inf')` 来指代正、负无穷大。

- **一些建议**

Python 的初学者在复制列表数据时经常犯一个错误。在下面的例子里，列表 B 只是一个指向列表 A 的引用。对 B[0] 的修改同样会修改 A[0]。

```
A = [1, 2, 3]
B = A
```

当复制一个 A 的独立副本时，我们可以使用以下语法格式：

```
A = [1, 2, 3]
B = A[:]
```

语句 [:] 用以复制一个列表。我们也可以复制一个去掉首元素的列表 A[1:]，或者去掉末尾元素的列表 A[:-1]，或者逆序的列表 A[::-1]。举例来说，下面的代码会生成一个所有行完全相同的矩阵 M，而对 M[0][0] 元素的修改会导致第一列所有元素被修改。

```
M = [[0] * 10] * 10
```

我们可以用下面两种正确的方式来初始化一个这样的矩阵：

```
M1 = [[0] * 10 for _ in range(10)]
M2 = [[0 for j in range(10)] for i in range(10)]
```

操作矩阵的简单方式是使用 numpy 模块，但我们在本书中不使用第三方类库，以便让程序代码能更方便翻译成 Java 或 C++ 代码。

另一个典型错误经常发生在使用 range 语句时。比如，下面的代码会顺序处理列表 A 中 0 至 9 号元素（包括 0 号和 9 号元素）：

```
for i in range(0, 10):                # 包括 0，不包括 10
    treat(A[i])
```

如果想逆序处理上述元素，仅反转参数是不够的。语句 range(10, 0, -1) 中的第三个参数代表循环的步长，语句会导致被处理元素中的 10 号元素被包含在内，而 0 号元素被排除在外。因此需要用以下方式来处理：

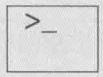
```
for i in range(9, -1, -1):            # 包括 9，不包括 -1
    treat(A[i])
```

1.3 输入输出

1.3.1 读取标准输入

在大部分编程竞赛的题目中，源数据都需要从标准输入设备来读取，并把输出显示到标准输出设备上。如果输入文件名叫 test.in，你的程序名叫 prog.py，那就可以在控制台执行以下命令，将输入文件的内容重定向到你的程序：

```
python prog.py < test.in
```



一般来说，在 Mac OS X 系统中，控制台可以用 Command + 空格，呼出 SpotLight 搜索后键入 Terminal 来打开；在 windows 系统中，使用“开始 - 执行 - cmd”；在 Linux 系统中，使用快捷键 Alt-F2[①]。

如果你想把程序的输出记录到名为 test.out 的输出文件中，使用的命令格式如下：

```
python prog.py < test.in > test.out
```

小技巧，如果你想把输出写入文件 test.out，同时还要显示在屏幕上，可以使用以下命令（注意，tee 命令在 Windows 环境下默认是不存在的）：

```
python prog.py < test.in | tee test.out
```

输入数据文件可以使用 input() 语句按行读取。input() 语句把读取到的行用字符串的形式返回，但不会返回行尾的换行符[②]。在 sys 模块中有一个类似的方法 stdin.readline()，这个方法不会删除行尾的换行符，但根据我们的经验，它的执行速度是 input() 语句的 4 倍。

如果读取到的行包含的应当是一个整数，我们使用 int 方法进行类型转换；如果是一个浮点数，我们使用 float 方法。当一行中包含多个空格分隔的整数时，我们首先使用 split() 方法把这一行拆分成独立的部分，然后用 map 方法把它们全部转换成整数。举例来说，当用空格分隔的两个整数——高度和宽度，需要在同一行内被读取时，可以使用以下命令[③]：

```
import sys
height, width = map(int, sys.stdin.readline().split())
```

如果你的程序在读取数据时遇到性能问题，根据我们的经验，可以仅使用一次系统调用，把整个输入文件读入，速度即可提升 2 倍。在下列语句中，假设输入数据中只有来自多行输入的整数，os.read() 方法的参数 0 表示标准输入流，常量 M 必须是一个大于文件大小的限值。比如，文件中包含了 10^7 个大小在 0 至 10^9 之间的整数，那么每个整数最多只能有 10 个字符，而两个整数中间最多只有两个分隔符（\r 和 \n，即回车和换行），我们可以选择 $M = 12 \cdot 10^7$。

① 使用组合快捷键是个好习惯。在 Windows 环境下，Windows 7 和 windows XP 系统都可以使用上述方式，而在 Windows 8、Windows 10 和 Windows Server 环境下，建议使用 Windows+R 组合键呼出“运行命令”窗口，再输入 cmd 打开控制台，或在 Windows 8 和 Windows 10 环境下，直接按 Windows 键打开“开始”屏幕，输入 cmd 后回车，也可以快速打开控制台。——译者注

② 根据操作系统的不同，换行符可能是 \r 或 \n，或二者皆有，但使用 input() 输入的时候不需要考虑这个问题。注意在 Python2.x 中，input() 方法的行为是不同的，同样，应当使用等价的 raw_input() 方法。

③ 以下命令中使用了 map 及管道概念。——译者注