

Python
Algorithms

Apress®

Python 算法教程

[挪威] Magnus Lie Hetland 著
凌杰 陆禹淳 顾俊 译

- 精通Python基础算法，畅销书作者最新力作

 中国工信出版集团

 人民邮电出版社
POSTS & TELECOM PRESS

Python
Algorithms

Python 算法教程

[挪威] Magnus Lie Hetland 著
凌杰 陆禹淳 顾俊 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Python算法教程 / (挪威) 赫特兰 (Hetland, M. L.)
著; 凌杰, 陆禹淳, 顾俊译. — 北京: 人民邮电出版
社, 2016. 1 (2016. 3重印)
ISBN 978-7-115-40483-1

I. ①P… II. ①赫… ②凌… ③陆… ④顾… III. ①
软件工具—程序设计—教材 IV. ①TP311.56

中国版本图书馆CIP数据核字(2015)第247097号

版 权 声 明

Python Algorithms: Mastering Basic Algorithms in the Python Language, Second Edition

by Magnus Lie Hetland, ISBN: 978-1-4842-0056-8

Original English language edition published by Apress Media.

Copyright ©2014 by Apress Media.

Simplified Chinese-language edition copyright ©2015 by Post & Telecom Press.

All rights reserved.

本书中文简体字版由 Apress L.P.授权人民邮电出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

版权所有, 侵权必究。

-
- ◆ 著 [挪威] Magnus Lie Hetland
 - 译 凌 杰 陆禹淳 顾 俊
 - 责任编辑 陈冀康
 - 责任印制 张佳莹 焦志炜
 - ◆ 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>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 20.75
字数: 497 千字 2016 年 1 月第 1 版
印数: 3001-5000 册 2016 年 3 月北京第 2 次印刷
著作权合同登记号 图字: 01-2013-6325 号

定价: 69.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316
反盗版热线: (010)81055315

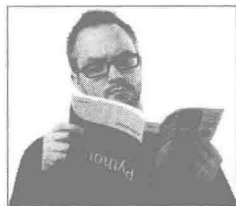
内容提要

Python 是一种面向对象、解释型计算机程序设计语言，其应用领域非常广泛，包括数据分析、自然语言处理、机器学习、科学计算以及推荐系统构建等。

本书用 Python 语言来讲解算法的分析和设计。本书主要关注经典的算法，但同时会为读者理解基本算法问题和解决问题打下很好的基础。全书共 11 章。分别介绍了树、图、计数问题、归纳递归、遍历、分解合并、贪心算法、复杂依赖、Dijkstra 算法、匹配切割问题以及困难问题及其稀释等内容。本书在每一章结束的时候均有练习题和参考资料，这为读者的自我检查以及进一步学习提供了较多的便利。在全书的最后，给出了练习题的提示，方便读者进行查漏补缺。

本书概念和知识点讲解清晰，语言简洁。本书适合对 Python 算法感兴趣的初中级用户阅读和自学，也适合高等院校的计算机系学生作为参考教材来阅读。

作者简介



Magnus Lie Hetland 是一位经验丰富的 Python 程序员，他自 20 世纪 90 年代以来就一直在使用该语言。同时，他还是挪威科技大学的副教授，有着数十年的算法教学经验，是《Beginning Python》^①一书的作者。

① 译者注：此书最初版本名为 Practical Python，现已更新至第 2 版，其中译本《Python 基础教程》已于 2010 年 7 月由人民邮电出版社出版。

技术评审人简介



Stefan Turalski 是一位非常乐于分享实际经验的程序员。这些经验不仅有软件方面的，还包括一些解决方案，以及攀登陡峭学习曲线的经验。

他有着十年以上的从业经验，曾针对知识管理、嵌入式网络、医疗、电力及天然气交易等数十个领域的问题构建过解决方案，并且近些年还将业务扩展到了金融领域。

虽然这位朋友一直专注于代码优化与系统集成，但其涉猎的编程语言可不少（或者说他已经被淹没在其中了），也曾滥用过一些开源的、商业的软件框架、库以及服务等。

Stefan 目前正在伦敦的一家金融机构研究一套具有高扩展性、低延迟的当日风险评估系统（intraday risk valuation system）。因此，他最近的兴趣主要集中在响应式编程（reactive programming）相关领域，例如 F#、Clojure、Python、OpenCL、WebGL 等。

Stefan 至今都无法相信自己在 Magnus Lie Hetland 的这本超级著作的第二版修订过程中提供了不少足以令人信赖的帮助。现在他只希望自己（和您）在学习作者带来的那些算法问题时所损失的脑细胞能早日得到恢复，并变得更聪明一些。

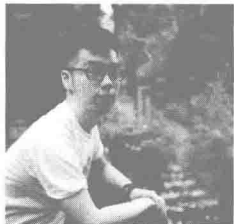
译者简介



凌杰 1981年生，浙江大学远程教育学院“荣誉学员”、“2012年度十大远程骄子”。目前为自由开发者、技术译者。精通多门编程语言，拥有丰富的软件开发及测试经验。个人崇尚黑客文化，支持开源运动，时常出没于国内外各种技术社区，曾担任上海交通大学饮水思源BBS的技术区区长，并兼任该区C/C++板板主多年。近些年来还参与了多项技术相关的外文翻译工作，译作包括《JavaScript面向对象编程指南》、《元素模式》等。



陆禹淳 1987年生，软件工程师。《JavaScript面向对象编程指南（第2版）》译者，上海交通大学饮水思源BBS水源先生，技术智囊，WebDevelop板块以及Algorithms板块的板主。对JavaScript以及Python等语言经验丰富。维护一个诞生于20年以前的、由C++写的BBS终端项目，是BBS社区“思源湖”的站长。业余时间喜欢冒充及“欺负”学生，去参加平均年龄24岁左右的黑客马拉松比赛。目前在物联网行业工作。



顾俊 1989年生，上海交通大学工学学士，在复旦大学获得计算机软件与理论专业硕士学位。拥有多年使用Java以及Python等语言的开发经验。参与过多项企业项目开发以及软件设计比赛。热衷于开源项目，熟悉Hadoop生态系统，对于各类新型数据库、分布式系统以及大数据分析有着浓厚的研究兴趣和丰富的实践经验。目前在互联网行业，主要从事数据研发相关的工作与研究。

译者序

在计算机的世界中，算法本质上是我们对某一个或者某一类问题的解决方案。也就是说，如果我们想用计算机来解决问题的话，就必须将问题的解决思路准确而完整地描述出来，同时计算机也要能理解这个描述。这需要我们这些程序员将整个描述转化成一系列清晰的指令，这些指令要能接收满足一定规范的输入，并在有限的时间内产生出相应的输出。我们通常将这些指令称为程序，而算法则是程序的灵魂。

然而，程序光有灵魂是不够的。例如，诚然搜索算法可以用来解决搜索类问题，但我们通常是不会为搜索而搜索的。所有搜索算法在进入程序的时候，都要面对一些实质性的内容，比如新闻信息、论文存档，而这些内容往往都有具体的存储系统，如数据库、文件系统等。这些系统本身也有各自的数据结构，如图、树、列表等，所以算法并不是我们写程序时唯一要考虑的问题。在很多现实情况下，它甚至还不是主要问题。

所以，如果用 C 或 C++ 来进行编程教学的话，我们对于算法设计的专注力很容易被数据结构这种更为基础的细节干扰。毕竟用这些语言实现并使用好数据结构本身就已经很复杂了。因此，如果想专注于算法教学，就需要一种不太需要程序员在数据结构上花太多时间的编程语言。

在相当长的一段时间里，我们在进行算法设计的时候通常用的是一种更接近于人类语言的“伪代码”。这种代码足够抽象，能让我们专注于算法的表达，但遗憾的是，它们无法在计算机中执行，我们最终还是要将其翻译成真正的编程语言。这很管用，但显然不够优雅。我们需要一种既能在抽象层面接近于这些“伪代码”，又能在计算机上像 C 那样通用的语言。Python 就是这样一种语言。

这本书就是一本用 Python 来进行算法设计教学的书，本书的作者在其上一本著作《Python 基础教程》中已经展现了其丰富的教学经验和实力，我本人亦从中受益匪浅。能翻译他的后续作品，我深感荣幸。但翻译一本算法书所需要付出的时间和精力还是远远超出了我的想象。而且，当我译了近八个月，终于快将本书第 1 版译完的时候，它的第 2 版又出版了。于是我不得不找了两个朋友（见译者简介），他们的工作是重新针对第 2 版对译稿进行校对、增改以及最后两章的初译。我很感谢他们给予我的帮助。除此之外，我还要感谢我的好朋友、《深入解析 Windows 操作系统》（第 6 版）的译者范德成工程师，他全程参与了本书的校对，对译稿进行了严格审阅，提供了很多宝贵建议。

祝愿这本书能给读者们带来帮助，同时也希望你们阅读快乐。

发·杰

2015 年 5 月 30 日

于新安江畔

致 谢

我感谢每个直接或间接成就了这本书的人。首先要感谢的当然是 Arne Halaas 和 Bjørn Olstad，他们是我在算法方面的导师。然后是 Apress 出版社的相关负责团队以及我那声名显赫的技术编辑 Alex。感谢 Nils Grimsmo、Jon Marius Venstad、Ole Edsberg、Rolv Seehuus、Jorg Rødsjø 等人在本书上所投入的精力与提供的帮助。我还要感谢我的父母 Kjersti Lie、Tor M. Hetland 以及妹妹 Anne Lie-Hetland 对我的关怀与支持。还有 Axel 叔叔，谢谢您帮忙检查我的法语。最后要特别感谢 PSF(Python Software Foundation)组织，他们准许我部分复制 Python 标准库的源码；还有 Randall Munroe，谢谢您准许我引用那些精彩的 XKCD 漫画作品。

前 言

这本书结合了我的三大爱好：算法、Python 编程及诠释事物。对我来说，这三项都是美学问题——找出让事情尽善尽美的方法。这就需要我们首先去发现相关事物的精华所在，然后精雕细琢，使其发光发亮，或至少比原先要闪亮一些。当然，由于某些材料表面的杂质太多，加工的结果可能会有些不尽人意。但幸运的是，本书中所涉及的内容大多都是现成品，因为我所写的都是一些久负盛名的算法及其论证，采用的也是最受欢迎的编程语言之一。至于在诠释事物方面，我一直在努力试着让事情变得尽可能显而易见一些。但即便如此，我也肯定还有许多失败的地方。如果读者有任何对本书的改进建议，我都非常乐意听取。没准这其中的某些意见会成就本书将来的修订版呢！但就目前来说，我还希望读者能享受这本书，用你们的洞察力去重新发现一些玩法，并且去实际运行它们。如果可能的话，用它制造出一些“可怕”的东西也行。总之，想怎么玩就怎么玩吧！

目 录

第 1 章 引言1	2.4 请提防黑盒子..... 36
1.1 这是一本什么样的书.....2	2.4.1 隐性平方级操作..... 37
1.1.1 本书将主要涉及以下内容.....3	2.4.2 浮点运算的麻烦..... 38
1.1.2 本书还将简单或部分涉及 以下内容.....3	2.5 本章小结..... 41
1.1.3 本书不会涉足以下领域.....3	2.6 如果您感兴趣..... 42
1.2 为什么要读这本书.....3	2.7 练习题..... 42
1.3 一些准备工作.....5	2.8 参考资料..... 43
1.4 本书主要内容.....5	第 3 章 计数初步 46
1.5 本章小结.....7	3.1 求和式的含义..... 46
1.6 如果您感兴趣.....7	3.1.1 更多希腊字母..... 47
1.7 练习题.....7	3.1.2 求和式的运用..... 47
1.8 参考资料.....8	3.2 两种赛制的故事..... 48
第 2 章 基础知识9	3.2.1 握手问题..... 48
2.1 计算领域中一些核心理念.....9	3.2.2 龟兔赛跑..... 50
2.2 渐近记法.....11	3.3 子集与排列组合..... 54
2.2.1 我看不懂这些希腊文.....12	3.4 递归与递归式..... 56
2.2.2 交通规则.....14	3.4.1 手动推导..... 57
2.2.3 让我们拿渐近性问题 练练吧.....16	3.4.2 几个重要例子..... 59
2.2.4 三种重要情况.....19	3.4.3 猜测与检验..... 62
2.2.5 实证式算法评估.....20	3.4.4 主定理：一刀切式的 解决方案..... 65
2.3 图与树的实现.....24	3.5 这一切究竟是什么呢..... 67
2.3.1 邻接列表及其类似结构.....26	3.6 本章小结..... 69
2.3.2 邻接矩阵.....29	3.7 如果您感兴趣..... 69
2.3.3 树的实现.....32	3.8 练习题..... 70
2.3.4 多种表示法.....35	3.9 参考资料..... 71

第4章 归纳、递归及归简72	6.3 折半搜索..... 129
4.1 哦, 这其实很简单.....73	6.3.1 搜索树的遍历及其剪枝..... 132
4.2 一而再, 再而三.....74	6.3.2 选取算法..... 135
4.3 魔镜, 魔镜.....77	6.4 折半排序..... 137
4.4 基于归纳法(与递归法)的设计.....82	排序操作究竟可以有多快..... 140
4.4.1 寻找最大排列.....82	6.5 三个额外实例..... 140
4.4.2 明星问题.....86	6.5.1 最近点对问题..... 141
4.4.3 拓扑排序问题.....88	6.5.2 凸包问题..... 142
4.5 更强的假设条件.....92	6.5.3 最大切片问题..... 144
4.6 不变式与正确性.....94	6.6 树的平衡与再平衡..... 145
4.7 松弛法与逐步完善.....94	6.7 本章小结..... 151
4.8 归简法+换位法=困难度证明.....95	6.8 如果您感兴趣..... 152
4.9 一些解决问题的建议.....97	6.9 练习题..... 152
4.10 本章小结.....98	6.10 参考资料..... 153
4.11 如果您感兴趣.....98	第7章 贪心有理吗? 请证明 154
4.12 练习题.....99	7.1 步步为营, 万无一失..... 154
4.13 参考资料.....101	7.2 背包问题..... 158
第5章 遍历: 算法学中的万能钥匙102	7.2.1 分数背包问题..... 158
5.1 公园漫步.....109	7.2.2 整数背包问题..... 159
5.1.1 不允许出现环路.....109	7.3 哈夫曼算法..... 159
5.1.2 停止循环遍历的方式..... 110	7.3.1 具体算法..... 160
5.2 继续深入..... 111	7.3.2 首次贪心选择..... 162
5.3 无限迷宫与最短(不加权)	7.3.3 走完剩余部分..... 163
路径问题..... 115	7.3.4 最优化归并..... 164
5.4 强连通分量.....120	7.4 最小生成树问题..... 165
5.5 本章小结.....123	7.4.1 最短边问题..... 166
5.6 如果您感兴趣.....123	7.4.2 其余部分的相关情况..... 167
5.7 练习题.....124	7.4.3 Kruskal 算法..... 168
5.8 参考资料.....125	7.4.4 Prim 算法..... 170
第6章 分解、合并、解决126	7.5 贪心不是问题, 问题是
6.1 树状问题, 即平衡问题.....126	何时贪心..... 173
6.2 经典分治算法.....129	7.5.1 坚持做到最好..... 173
	7.5.2 尽量做到完美..... 174

7.5.3 做好安全措施	175	10.3 最大流问题	236
7.6 本章小结	177	10.4 最小切割集问题	240
7.7 如果您感兴趣	178	10.5 最小成本的流及赋值问题	241
7.8 练习题	178	10.6 一些应用	243
7.9 参考资料	179	10.7 本章小结	247
第 8 章 复杂依赖及其记忆体化	180	10.8 如果您感兴趣	247
8.1 不要重复自己	181	10.9 练习题	248
8.2 有向无环图中的最短路径问题	187	10.10 参考资料	249
8.3 最长递增子序列问题	190	第 11 章 困难问题及其（有限）稀释	250
8.4 序列比对问题	193	11.1 重提归简	251
8.5 背包问题的反击	196	11.2 不待在肯萨斯州了	254
8.6 序列的二元分割	199	11.3 但目前，我们还是得回 到肯萨斯州	255
8.7 本章小结	202	11.4 我们应从何处开始？ 前往何处呢	259
8.8 如果您感兴趣	203	11.5 怪物乐园	264
8.9 练习题	203	11.5.1 背包的返回	264
8.10 参考资料	204	11.5.2 分团与着色	266
第 9 章 Dijkstra 及其朋友们		11.5.3 路径与环路	269
从 A 到 B 的旅程	206	11.6 当困难来临的时候，聪明人 就开始犯错	272
9.1 扩展知识	207	11.7 尽力寻找解决方案	275
9.2 松弛可“疯狂”	208	11.8 这些故事告诉我们	277
9.3 找到隐藏的 DAG 图	213	11.9 本章小结	279
9.4 多对多问题	216	11.10 如果您感兴趣	279
9.5 “牵强”的子问题	218	11.11 练习题	280
9.6 中途相遇	220	11.12 参考资料	281
9.7 把握未来走向	223	附录 A 猛踩油门！令 Python 加速	282
9.8 本章小结	226	附录 B 一些著名问题与算法	286
9.9 如果您感兴趣	227	问题部分	286
9.10 练习题	228	算法与数据结构部分	289
9.11 参考资料	228	附录 C 图论基础	295
第 10 章 匹配、切割及流量	229		
10.1 二分图匹配	230		
10.2 不相交的路径	233		

附录 D 习题提示	301	第 6 章	308
第 1 章.....	301	第 7 章	310
第 2 章.....	301	第 8 章	312
第 3 章.....	303	第 9 章	313
第 4 章.....	304	第 10 章	314
第 5 章.....	307	第 11 章	315

引 言

1. 提出问题。
2. 思考真正困难所在。
3. 提出解决方案。

——摘自《The Feynman Algorithm》，Murray Gell-Mann 著

让我们先来考虑一下下面这个问题：我们想要访遍瑞典境内所有的城市、小镇和村庄，然后再返回出发地点。显然，这段旅程肯定要耗费掉不少时间（毕竟我们要访问 24 978 个地方），因而我们希望能最小化该旅行路线。也就是说，我们既要能按计划逐个参观这些地方，又要尽可能地走出一条最短路线来。作为一个程序员，我们当然不屑于用手工方式来设计该路线，显然会更倾向于用写代码的方式来完成相关计划。然而，这似乎是一件不可能完成的任务。的确，想要写出一个针对少量城镇的简单程序并不难，但如果想要进一步用它来解决实际问题，相关的改进就会变得极其困难。这是为什么呢？

其实，早在 2004 年就已经有一个五人的研究团队^①发现了这个被称为瑞典之旅的问题，之后陆续有多个别的研究团队也都试图解决过这个问题，但是都失败了。该五人团队采用了一款带有智能优化技术的、能模拟交易技巧的软件，并将其运行在一组由 96 台机器（Xeon 2.6GHz）组成的工作站集群上。结果，该软件从 2003 年 3 月一直运行到 2004 年 5 月。最终不得不在打印出该问题的最佳解决方案之前被中止运行。因为综合各方面的因素，他们估计程序所需要的总 CPU 时间竟然长达 85 年！

下面再来思考一个类似的问题：我们想要在中国西部的喀什市到东海岸的宁波市之间找出一条最短路线^②。目前，中国境内有 3 583 715 千米长的公路和 77 834 千米长的铁路，之间有着数以百万计的路口可供我们考虑，其中可供选择的路线千千万万，难以估算。这个问题似乎与前一个问题密切相关，都是希望通过 GPS 导航和在线地图服务找出最短路线的规划问题，并且不能有太明显的滞后现象。也就是说，只要向您最爱的地图服务程序输入这两个城市，您就应该能在短时间内得到它们之间的最短路线。这又应该怎么做呢？

关于这些问题，读者将来都可以在本书中找到相关进一步的讨论。例如，第一个问题其实叫

① 这五人分别是 David Applegate、Robert Bixby、Vašek Chvátal、William Cook 与 Keld Helsgaun。

② 这里假设飞机不是我们的选项。

作旅行商问题 (traveling salesman, 或推销员问题 (salesrep)), 这将是本书第 11 章所要涵盖的内容。而所谓的最短路径问题 (shortest path), 则是第 9 章中所要介绍的内容。但我们更希望读者能从中学会如何深入分析问题的困难所在, 并且掌握那些已被承认的、高效的知名解决方案。更为重要的是, 我们将在本书中传授一系列常用的算法处理技术, 及一些与计算机运算相关的问题, 以便帮助读者熟练掌握书中所介绍的技术和算法, 以及所演示的针对困难问题而提出的最接近期望值的解决方案。在这一章中, 我们将对本书的主要内容做简单介绍——我们可以期望什么, 我们应该期望什么。另外, 我们还会列举本书各章节所要介绍的具体内容, 以便读者可以直接跳到自己想阅读的内容附近。

1.1 这是一本怎么样的书

简而言之, 这是一本为 Python 程序员解决算法问题的书。正如书上所说, 它的内容涵盖相关的面向对象模式和一些常见问题的处理方式——也就是相关的解决方案。对于一个算法设计者, 我们需要的不是简单地实现或执行一些现有算法的能力。相反, 我们期望能拿出一个新算法——一个能解决一般性问题的、前人没有提过的全新解决方案。在本书中, 我们要学习的就是此类解决方案的设计原则。

但这又不是一本传统意义上的算法书。毕竟, 大部分这类题材的权威书籍 (例如 Knuth 那部经典著作, 或是由 Cormen 等人合著的那本标准教科书^①) 都属于理论研究型的, 显得有些过于严肃, 尽管其中也不乏一些侧重于可读性的作品 (例如 Kleinberg 与 Tardos 合写的书^②就是其中之一)。当然, 我们在这里并不是要取代这些优秀的作品, 而只是希望能在此基础上做一些补充。我希望利用自己多年的算法教学经验, 尽可能清楚地为读者诠释算法的工作方式, 以及一些需要共同遵守的基本原则。对于一个程序员来说, 这种程度的诠释可能就已经足够了, 读者需要有更多的机会去理解为什么相关的算法是正确的、如何将这些算法运用到他们所面对的新问题中去。但这就需要去阅读一些更形而上的、百科全书式的教科书。我希望这本书能为读者打下一定的基础, 这将有助于他们理解相关的定理以及其相应的证明。

■ **请注意:** 这本书和其他算法书之间的一个区别是, 我采用了谈话式的风格。虽然我希望这至少能吸引一些我的读者, 但这可能不是您喜欢的风格。我们对此深感抱歉, 但现在您已经至少被提醒了。

除此之外, 市面上还有另一种算法类书籍。它们通常以“(数据结构与) 算法 (blank 版)”为题, 这里的 blank 通常为作者本人所使用的编程语言。有不少这样的书 (而且似乎都是以 blank=Java 的情况为主), 但其关注点多集中在与基本数据结构有关的东西上, 以至于忽略了某些更为实质性的内容。如果说这是某一门数据结构基础课程的教科书, 或许还可以理解, 但对于 Python 程序员来说, 学习单向或双向链表并不是一件能让所有人兴致勃勃的事情 (尽管我们在下

① 译者注: 指《The Art of Computer Programming》、《Introduction to Algorithms》这两部经典巨作。

② 译者注: 即《Algorithm Design》。

一章中还是会提到一些)。即便是哈希这样重要的技术,我们也可以通过 Python 中的字典类型免费获得相应的哈希表,完全不需要再去考虑重新实现它们。恰恰相反,我将注意力集中在更高级的一些算法。但这样一来,许多重要的概念在 Python 语言本身或标准库对相关算法(如查找、搜索、哈希等)的“黑盒”化实现中被淡化了。为此,我们在文中加入了一些特定的“黑盒子”专栏,以做补充。

当然,本书与那些“Java/C/C++/C#”算法流派还有一个显著的区别,即这里的 blank 为 Python。这使得本书更接近那些与语言无关的算法书(例如 Knuth^①、Cormon 等人以及 Kleinberg 与 Tardos 的作品),这些书常常使用伪代码来说明问题。这实际上是一种侧重于可读性的伪编程语言,因而它不具备执行能力。而可读性正好是 Python 最显著的特点之一,因此它或多或少可以被视为是一种具有执行能力的伪代码。就算我们从没用过 Python,也能看懂绝大部分 Python 程序。总之,这书中代码示例都是高度可读的——我们不需要成为 Python 方面专家,也能轻松读懂这些示例(尽管有时候还是需要读者去查阅一些与内置函数有关的资料)。当然,您也可以把这些示例当作伪代码来理解。综上所述……

1.1.1 本书将主要涉及以下内容

- 算法分析,主要侧重于渐近运行时间分析。
- 算法设计的基本原则。
- 如何用 Python 描述那些常用的数据结构。
- 如何用 Python 实现那些知名算法。

1.1.2 本书还将简单或部分涉及以下内容

- Python 中直接可用的算法,它们通常是语言本身或其标准库的一部分。
- 纯思想性及形而上的内容(尽管本书会对它们做一些证明,并提供相关的证明示例)。

1.1.3 本书不会涉足以下领域

- 与数值计算或数学理论有关的算法(只有第 2 章中涉及了一些浮点运算)。
- 并行算法与多核编程。

正如大家所知,“用 Python 实现”只是整个拼图的一部分。我们所希望的是,读者能掌握其中的设计原则与理论基础,以便能设计出属于自己的算法与数据结构。

1.2 为什么要读这本书

当我们在工作中使用算法时,通常都是希望能更有效地解决问题、使程序运行得更快,并且

^① 众所周知,Knuth 还专门设计了一套属于他自己的抽象计算编码。