

Mastering Algorithms with Python

Python

算法详解

张玲玲◎编著

网站论坛+QQ群
提供作者服务，
形成互帮互助的朋友圈

网站论坛：

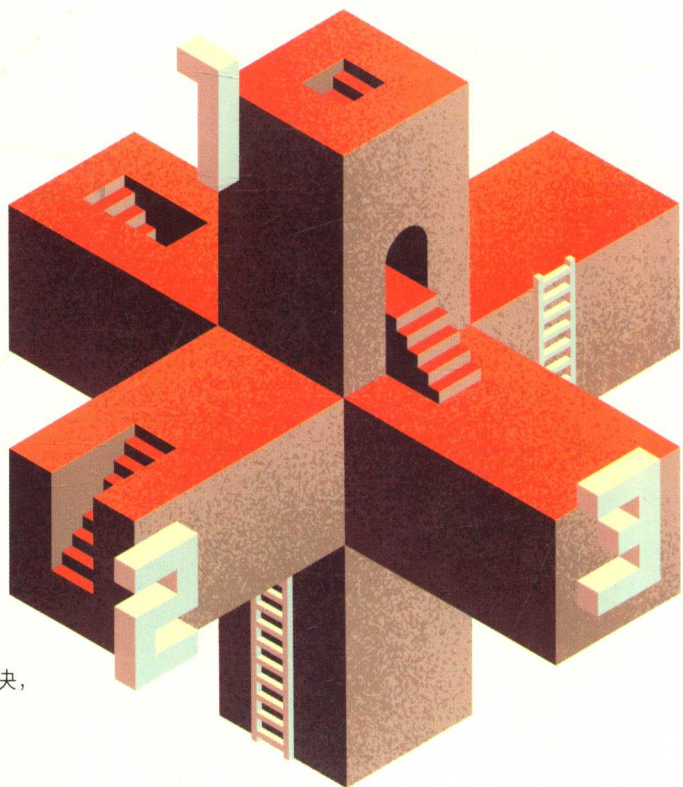
作者答疑 / 学习辅导 / PPT 资源下载

QQ群：

读者学习讨论 / 与作者直接交流

涵盖**数据结构**，**线性表**、**队列**和**栈**，**树**，**图**，
查找算法，**内部排序**算法，**经典算法问题**的解决，
图像问题的解决等内容

195 个实例，提供更多的**实践演练**机会

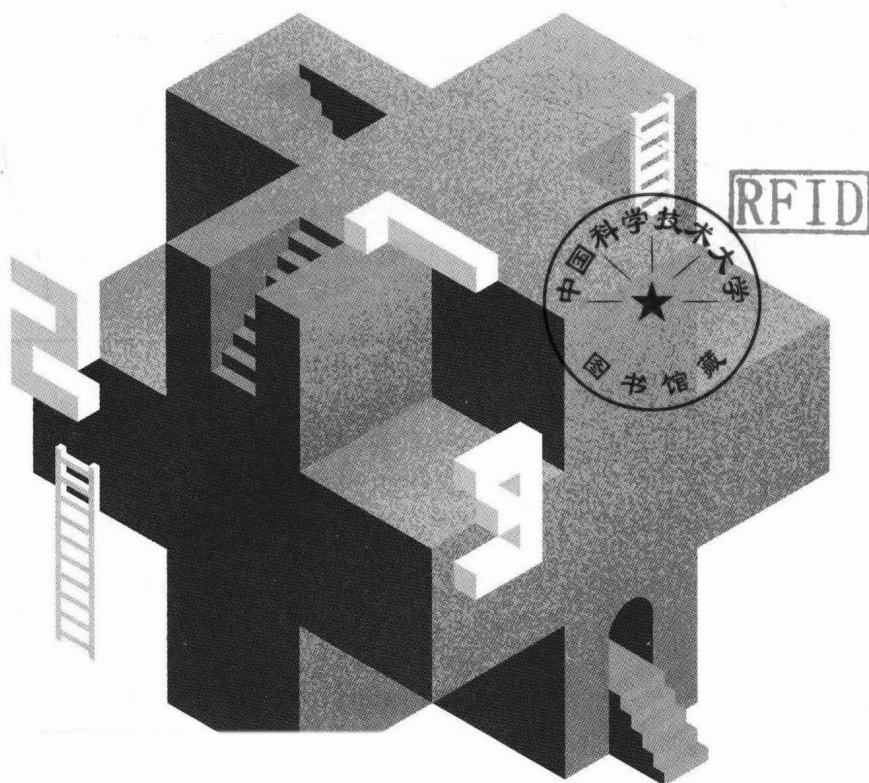


Mastering Algorithms with Python

Python

算法详解

张玲玲◎编著



人民邮电出版社

北京

图书在版编目 (C I P) 数据

Python算法详解 / 张玲玲编著. — 北京 : 人民邮电出版社, 2019. 6
ISBN 978-7-115-50338-1

I. ①P… II. ①张… III. ①软件工具—程序设计
IV. ①TP311.561

中国版本图书馆CIP数据核字(2018)第278867号

内 容 提 要

本书循序渐进、由浅入深地讲解 Python 算法的核心技术, 并通过具体实例的实现过程演练各个知识点的具体使用流程。全书共 13 章, 包括算法, 数据结构, 常用的算法思想、线性表、队列和栈, 树, 图, 查找算法, 内部排序算法, 经典的数据结构问题, 数学问题的解决, 经典算法问题的解决, 图像问题的解决, 游戏和算法等内容。

本书不但适合研究和学习算法的初学者, 也适合有一定算法基础的读者, 还可以作为大中专院校相关专业师生的学习用书和培训学校的教材。

-
- ◆ 编 著 张玲玲
责任编辑 张 涛
责任印制 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>
三河市祥达印刷包装有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 22.75
字数: 606 千字
印数: 1—2 400 册
- 2019 年 6 月第 1 版
2019 年 6 月河北第 1 次印刷
-

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
广告经营许可证: 京东工商广登字 20170147 号

前 言

从程序设计语言实践的角度而言，算法是从事信息技术领域工作的专业人员必须学习的一门基础理论课程。无论你采用哪一种程序设计语言编写程序，要使设计的程序能快速而高效地完成预定的任务，算法是一个关键因素。

本书内容

市面上许多与算法相关的图书会介绍大量的理论或者讲解表达算法的核心概念，但是这类书缺乏完整的程序设计范例，因而对于第一次接触算法的初学者来说，将算法运用于实际应用就成了一道跨不过的鸿沟。为了帮助更多人用比较轻松的方式了解各种算法和各种经典数学问题的求解方法，本书包括了枚举算法、分治算法、贪心算法、试探算法、迭代算法；线性表、队列和栈；二叉树、霍夫曼树；图的遍历、图的连通性、寻求最短路径；查找算法；内部排序法、插入排序法、交换类排序法、选择排序法、归并排序法、基数排序法；经典的数据结构问题，如约瑟夫环、大整数运算、顺序表的处理、链表的基本操作、基于列表实现二叉树、实现 AVL 树、使用二维数组生成有向图；经典数学问题的解决，如利用递归算法获取斐波那契数列前 n 项的值、通过多个进程验证哥德巴赫猜想、百钱买百鸡、素数问题、埃及分数式等。

为了让读者学以致用，每讲一个算法，同时都会给出具体的实例和运行的效果图。同时使用 Python 实现算法，以期能将各种算法真正应用在学习者将来的程序设计中。因此，这是一本学习算法的入门书。

本书特色

(1) 以“入门到精通”的写作方法构建内容，让读者入门容易。

为了使读者能够完全看懂本书的内容，本书遵循“从入门到精通”基础类图书的写法，循序渐进地讲解算法的知识。

(2) 破解语言难点，以“技术解惑”贯穿全书，绕过学习中的陷阱。

为了帮助读者学懂算法，每章都会有“技术解惑”模块，让读者知其然又知其所以然。

(3) 书中包含大量典型实例。

书中有大量实例，通过这些实例的练习，读者有更多的实践演练机会。

(4) 通过 QQ 群和网站论坛实现教学互动，形成互帮互学的朋友圈。

本书作者为了方便给读者答疑，特地提供了网站论坛、QQ 群等技术支持，并且随时在线与读者互动。让大家在互学互帮中形成一个良好的学习编程的氛围。

本书的论坛是 toppr 网站（网站后缀名为.net）。

本书的 QQ 群是 292693408。

本书读者对象

- 初学编程的自学者
- 计算机相关专业的教师和学生
- 程序开发人员
- 相关培训机构的教师和学员

致谢

在编写过程中，本书得到了人民邮电出版社编辑的大力支持，正是各位编辑的敬业和高效，才使得本书能够在这么短的时间内出版。另外，十分感谢我的家人给予的巨大支持。本人水平毕竟有限，书中纰漏之处在所难免，诚请读者提出意见或建议。编辑联系邮箱是 zhangtao@ptpress.com.cn。

最后感谢读者购买本书，希望本书能成为读者编程道路上的挚友，祝读者阅读快乐！

作者

资源与支持

本书由异步社区出品，社区（<https://www.epubit.com/>）为您提供相关资源和后续服务。

配套资源

本书配套的资源是书中所有实例的源代码。

要获得以上配套资源，请在异步社区本书页面中单击 **配套资源**，跳转到下载界面，按提示进行操作即可。注意，为保证购书读者的权益，该操作会给出相关提示，要求输入提取码进行验证。

如果您是教师，希望获得教学配套资源，请在社区本书页面中直接联系本书的责任编辑。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，单击“提交勘误”，输入勘误信息，单击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。

The screenshot shows a web form for submitting勘误 (勘误). At the top, there are three tabs: '详细信息' (Detailed Information), '写书评' (Write a Review), and '提交勘误' (Submit勘误), with the latter being the active tab. Below the tabs, there are three input fields: '页码:' (Page Number), '页内位置 (行数):' (Page Position (Line Number)), and '勘误印次:' (勘误次数). Below these fields is a rich text editor with a toolbar containing icons for bold (B), italic (I), underline (U), strikethrough (ABC), bulleted list (•), numbered list (1-2-3), quote ("), link (S), and unlink (S). At the bottom right of the form, there is a '字数统计' (Character Count) label and a '提交' (Submit) button.

扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。



与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 <https://www.epubit.com>。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术等。



异步社区



微信服务号

目 录

第 1 章 算法概述	1
1.1 算法的基础.....	2
1.1.1 算法的特征.....	2
1.1.2 何为算法.....	2
1.2 计算机中的算法.....	3
1.2.1 认识计算机中的算法.....	3
1.2.2 为什么说算法是程序的灵魂.....	4
1.3 计算机中表示算法的方法.....	4
1.3.1 用流程图表示算法.....	4
1.3.2 用 N-S 流程图表示算法.....	6
1.3.3 用计算机语言表示算法.....	6
1.4 学习建议.....	6
第 2 章 数据结构	8
2.1 使用列表.....	9
2.1.1 列表的基本用法.....	9
2.1.2 删除列表中的重复元素并保持顺序不变.....	10
2.1.3 找出列表中出现次数最多的元素.....	11
2.1.4 排序类定义的实例.....	11
2.1.5 使用列表推导式.....	12
2.1.6 命名切片.....	13
2.2 使用元组.....	14
2.2.1 创建并访问元组.....	14
2.2.2 修改元组.....	15
2.2.3 删除元组.....	15
2.2.4 使用内置方法操作元组.....	15
2.2.5 将序列分解为单独的变量.....	16
2.2.6 将序列分解为单独的变量.....	17
2.2.7 实现优先级队列.....	17
2.3 使用字典.....	19
2.3.1 创建并访问字典.....	19
2.3.2 添加、修改、删除字典中的元素.....	19
2.3.3 映射多个值.....	20
2.3.4 使用 OrderedDict 创建有序字典.....	21
2.3.5 获取字典中的最大值和最小值.....	22
2.3.6 获取两个字典中相同的键值对.....	23
2.3.7 使用函数 itemgetter() 对字典进行排序.....	24
2.3.8 使用字典推导式.....	25
2.3.9 根据记录进行分组.....	26
2.3.10 转换并换算数据.....	27
2.3.11 将多个映射合并为单个映射.....	28
第 3 章 常用的算法思想	30
3.1 枚举算法思想.....	31
3.1.1 枚举算法基础.....	31
3.1.2 实践演练——24 点游戏.....	31
3.1.3 实践演练——计算平方根.....	32
3.2 递归算法思想.....	32
3.2.1 递归算法基础.....	33
3.2.2 实践演练——解决“斐波那契数列”问题.....	33

3.2.3	实践演练——解决 “汉诺塔”问题	34	第4章 线性表、队列和栈	50
3.2.4	实践演练——解决 “阶乘”问题	36	4.1 线性表操作	51
3.3	分治算法思想	37	4.1.1 线性表的特性	51
3.3.1	分治算法基础	38	4.1.2 顺序表操作	52
3.3.2	实践演练——求顺序表 中的最大值	38	4.1.3 实践演练——实现线性表 顺序存储的插入操作	53
3.3.3	实践演练——判断某个 元素是否在列表中	38	4.1.4 实践演练——实现线性表 顺序存储的删除操作	54
3.3.4	实践演练——找出一组 序列中第 k 小的元素	39	4.1.5 实践演练——顺序表的 插入、检索、删除和反转 操作	54
3.4	贪心算法思想	39	4.2 链表操作	57
3.4.1	贪心算法基础	39	4.2.1 什么是链表	57
3.4.2	实践演练——解决“找零” 问题	40	4.2.2 实践演练——实现完整 链表操作	60
3.4.3	实践演练——解决“汽车 加油”问题	41	4.2.3 实践演练——在链表中 增加比较功能	64
3.5	试探算法思想	42	4.2.4 实践演练——单链表 结构字符串	67
3.5.1	试探算法基础	42	4.3 先进先出的队列	70
3.5.2	实践演练——解决 “八皇后”问题	42	4.3.1 什么是队列	71
3.5.3	实践演练——解决 “迷宫”问题	44	4.3.2 Python语言的队列操作	72
3.6	迭代算法思想	45	4.3.3 实践演练——完整的 顺序队列的操作	72
3.6.1	迭代算法基础	46	4.3.4 实践演练——基于列表 实现的优先队列	73
3.6.2	实践演练——解决“非线性 方程组”问题	46	4.3.5 实践演练——基于堆 实现的优先队列	74
3.7	技术解惑	47	4.4 后进先出的栈	75
3.7.1	衡量算法的标准是什么	47	4.4.1 什么是栈	75
3.7.2	递推和递归有什么差异	48	4.4.2 顺序栈	76
3.7.3	总结分治算法能解决什么 类型的问题	48	4.4.3 链栈	77
3.7.4	分治算法的机理是什么	48	4.4.4 实践演练——实现顺序栈 操作	77
3.7.5	为什么说贪婪算法并不是 解决问题的最优方案	48	4.4.5 实践演练——使用顺序表 方法和单链表方法实现栈	78
3.7.6	回溯算法会影响算法 效率吗	49	4.5 实现堆队列操作	79
3.7.7	递归算法与迭代算法 有什么区别	49	4.5.1 Python中的堆操作	79
			4.5.2 实践演练——实现二叉 堆操作	80

4.6 技术解惑	82	5.5.2 二叉树和链表的效率比较	121
4.6.1 顺序表插入操作的时间复杂度是多少	82	5.5.3 如何输出二叉树中的所有路径	121
4.6.2 顺序表删除操作的时间复杂度是多少	82	第 6 章 图	122
4.6.3 顺序表按值查找操作的时间复杂度是多少	82	6.1 图的起源	123
4.6.4 堆和栈的区别是什么	82	6.2 图的相关概念	124
第 5 章 树	84	6.3 存储结构	127
5.1 树基础	85	6.3.1 使用邻接矩阵表示图	127
5.1.1 什么是树	85	6.3.2 实践演练——将邻接矩阵输出成图	128
5.1.2 树的相关概念	85	6.3.3 使用邻接表表示图	129
5.2 使用列表表示的树	86	6.3.4 实践演练——使用邻接表表示图	130
5.3 二叉树详解	87	6.4 图的遍历	131
5.3.1 二叉树的定义	87	6.4.1 深度优先搜索	131
5.3.2 二叉树的性质	88	6.4.2 广度优先搜索	132
5.3.3 二叉树的存储结构	88	6.4.3 实践演练——实现图的深度优先和广度优先搜索	133
5.3.4 实践演练——使用嵌套列表表示树	90	6.5 图的连通性	135
5.3.5 实践演练——把二叉树的任何子节点当成二叉树	91	6.5.1 无向图的连通分量	135
5.3.6 实践演练——实现二叉搜索树查找操作	93	6.5.2 实践演练——通过二维数组建立无向图	136
5.3.7 实践演练——实现二叉搜索树的删除操作	97	6.5.3 实践演练——根据邻接矩阵绘制无向图	137
5.3.8 遍历二叉树	104	6.5.4 最小生成树	138
5.3.9 线索二叉树	107	6.5.5 实践演练——实现最小生成树和拓扑序列	139
5.4 霍夫曼树	115	6.5.6 关键路径	140
5.4.1 霍夫曼树基础	115	6.5.7 实践演练——递归解决 AOE 网最长关键路径的问题	141
5.4.2 实践演练——使用面向过程方式和面向对象方式实现霍夫曼树	117	6.6 寻求最短路径	143
5.4.3 实践演练——实现霍夫曼树的基本操作	118	6.6.1 求某一顶点到其他各顶点的最短路径	143
5.4.4 总结霍夫曼编码的算法实现	120	6.6.2 任意一对顶点间的最短路径	145
5.5 技术解惑	120	6.6.3 实践演练——使用 Dijkstra 算法计算指定点到其他各顶点的路径	146
5.5.1 树和二叉树的差别是什么	120		

6.6.4	实践演练——使用 Floyd-Warshall 算法计算图的最短路径	147	7.4.4	散列表的查找过程	177
6.6.5	实践演练——使用 Bellman-Ford 算法计算图的最短路径	148	7.4.5	实践演练——使用散列表查找数据	177
6.6.6	实践演练——使用 Dijkstra 算法解决加权的最短路径问题	149	7.5	斐波那契查找法	178
6.7	技术解惑	150	7.5.1	斐波那契查找法介绍	178
6.7.1	几种最短路径算法的比较	150	7.5.2	实践演练——使用斐波那契查找法	179
6.7.2	邻接矩阵与邻接表的对比	152	7.6	高级树表查找算法	180
6.7.3	比较深度优先算法和广度优先算法	152	7.6.1	2-3 查找树介绍	180
第 7 章	查找算法	154	7.6.2	红黑树介绍	181
7.1	几个相关概念	155	7.6.3	实践演练——使用红黑树操作数据	181
7.2	基于线性表的查找法	155	7.6.4	B 树和 B+树	185
7.2.1	顺序查找法	155	7.6.5	实践演练——使用 B 树排序数据	186
7.2.2	实践演练——实现顺序查找算法	156	7.6.6	实践演练——使用 B+树操作数据	188
7.2.3	折半查找法	157	7.7	技术解惑	193
7.2.4	实践演练——使用折半查找法查找数据	158	7.7.1	分析查找算法的性能	193
7.2.5	插值查找法	160	7.7.2	分析散列法的性能	194
7.2.6	实践演练——使用插值查找法查找指定的数据	160	第 8 章	内部排序算法	195
7.2.7	分块查找法	161	8.1	排序基础	196
7.3	基于树的查找法	162	8.1.1	排序的目的和过程	196
7.3.1	二叉排序树	162	8.1.2	内部排序与外部排序	196
7.3.2	实践演练——实现二叉树的完整操作	165	8.1.3	稳定排序与不稳定排序	196
7.3.3	平衡二叉树	167	8.2	插入排序算法	197
7.3.4	实践演练——实现平衡二叉树的基本操作	170	8.2.1	直接插入排序	197
7.4	散列法	174	8.2.2	实践演练——编写直接插入排序算法	198
7.4.1	散列法的基本思想	174	8.2.3	实践演练——使用折半插入排序算法	198
7.4.2	构造散列函数	175	8.2.4	希尔排序	199
7.4.3	处理冲突	176	8.2.5	实践演练——使用希尔排序算法对数据进行排序	199
			8.2.6	实践演练——使用希尔排序处理一个列表	200
			8.3	交换类排序法	201
			8.3.1	冒泡排序(相邻比序法)	201

8.3.2	快速排序	201	8.7.2	希尔排序和插入排序的 速度比较	218
8.3.3	实践演练——实现从大 到小的冒泡排序	202	8.7.3	快速排序的时间耗费	218
8.3.4	实践演练——使用冒泡 排序算法排序	202	8.7.4	堆排序与直接选择排序的 区别	219
8.3.5	实践演练——实现基本的 快速排列	203	8.7.5	归并排序的效率与选择 方法	219
8.4	选择排序法	204	8.7.6	综合比较各种排序方法	219
8.4.1	直接选择排序	204	第 9 章 经典的数据结构问题	221	
8.4.2	树形选择排序	204	9.1	约瑟夫环	222
8.4.3	堆排序	205	9.1.1	问题描述	222
8.4.4	实践演练——实现直接 选择排序	206	9.1.2	算法分析	222
8.4.5	实践演练——演示选择 排序的操作步骤	207	9.1.3	具体实现	222
8.4.6	实践演练——选择排序和 Python 内置函数的效率 对比	208	9.2	大整数运算	224
8.4.7	实践演练——使用堆排序 处理数据	209	9.2.1	模拟大整数乘法的小学 竖式计算过程	224
8.4.8	实践演练——实现 最小堆	210	9.2.2	实现大数相加运算	225
8.5	归并排序	211	9.3	顺序表的修改、查找、统计、 删除、销毁操作	225
8.5.1	归并排序思想	211	9.3.1	算法分析	225
8.5.2	两路归并算法的思路	212	9.3.2	具体实现	226
8.5.3	实现归并排序	212	9.4	实现链表的基本操作	227
8.5.4	实践演练——使用归并 排序处理指定列表	213	9.4.1	算法分析	227
8.5.5	实践演练——使用归并 排序处理两个列表	213	9.4.2	具体实现	227
8.5.6	实践演练——使用两路 归并排序处理一个列表	214	9.5	带有尾节点引用的单链表	229
8.6	基数排序	215	9.5.1	算法分析	229
8.6.1	多关键字排序	215	9.5.2	具体实现	229
8.6.2	链式基数排序	216	9.6	增加新功能的单链表结构 字符串	230
8.6.3	实践演练——使用基数 排序处理随机数	217	9.7	实现堆排序功能	232
8.7	技术解惑	218	9.7.1	算法分析	232
8.7.1	插入排序算法的描述	218	9.7.2	具体实现	233
			9.8	实现队列、链表、顺序表和循环 顺序表	234
			9.8.1	时间复杂度分析	234
			9.8.2	具体实现	234
			9.9	基于列表实现二叉树	236
			9.10	实现二元表达式	237
			9.11	使用多叉树寻找最短路径	239

9.11.1	算法分析	239	10.10.1	用高斯消元法解 方程组	258
9.11.2	具体实现	239	10.10.2	用二分法解非线性 方程	260
9.12	实现 AVL 树	240	10.11	求平方根	261
9.13	使用二维数组生成有向图	244	10.11.1	使用二分法求平方根	261
9.14	使用广度优先和深度优先遍历 二叉树	245	10.11.2	用牛顿迭代法求 平方根	262
第 10 章	数学问题的解决	248	10.12	矩阵运算	264
10.1	解决一个数学问题	249	10.12.1	问题描述	264
10.1.1	问题描述	249	10.12.2	算法分析	264
10.1.2	具体实现	249	10.12.3	具体实现	264
10.2	使用递归算法计算两个数的 乘积	250	10.13	一元多项式运算	265
10.3	利用递归算法获取斐波那契数 列前 n 项的值	250	10.13.1	一元多项式求导	266
10.4	1000 以内的完全数	251	10.13.2	实现多项式的加法、 减法、乘法运算	266
10.4.1	问题描述	251	10.14	百钱买百鸡	267
10.4.2	算法分析	251	10.14.1	问题描述	267
10.4.3	具体实现	252	10.14.2	算法分析	268
10.5	多进程验证哥德巴赫猜想	253	10.14.3	具体实现	268
10.5.1	问题描述	253	10.15	素数问题	268
10.5.2	算法分析	253	10.15.1	求 1000 以内的所有 素数	269
10.5.3	具体实现	253	10.15.2	孪生素数问题	269
10.6	最大公约数和最小公倍数	255	10.15.3	金蝉素数	270
10.6.1	问题描述	255	10.15.4	可逆素数	271
10.6.2	算法分析	255	10.15.5	回文素数	271
10.6.3	具体实现	255	10.15.6	等差素数数列	272
10.7	亲密数	255	10.16	埃及分数式	273
10.7.1	问题描述	255	10.16.1	问题描述	273
10.7.2	算法分析	256	10.16.2	算法分析	274
10.7.3	具体实现	256	10.16.3	具体实现	274
10.8	计算 10000 以内的自守数	256	10.17	对正整数分解质因数	274
10.8.1	问题描述	256	10.17.1	问题描述	274
10.8.2	算法分析	256	10.17.2	算法分析	274
10.8.3	具体实现	257	10.17.3	具体实现	274
10.9	水仙花数	257	第 11 章	经典算法问题的解决	276
10.9.1	问题描述	257	11.1	歌星大奖赛	277
10.9.2	算法分析	257	11.1.1	问题描述	277
10.9.3	具体实现	257			
10.10	方程求解	257			

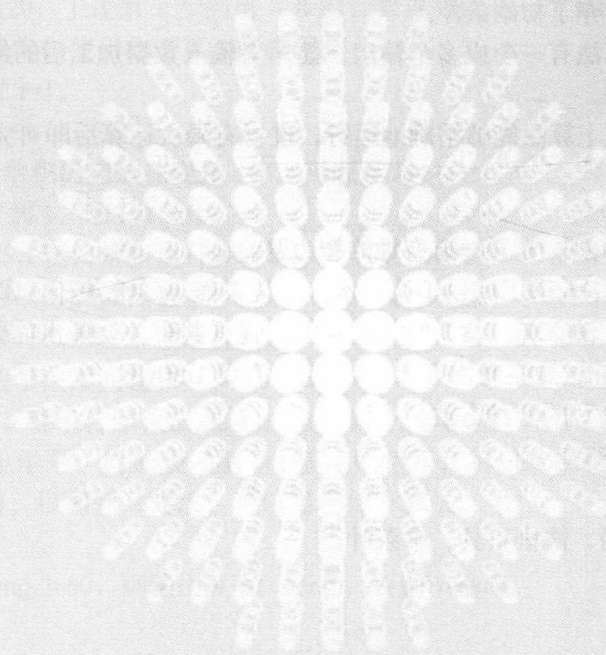
11.1.2 具体实现	277	11.12.2 算法分析	290
11.2 借书方案	278	11.11.3 具体实现	290
11.2.1 问题描述	278	11.12 奇数幻方问题	291
11.2.2 算法分析	278	11.12.1 问题描述	291
11.2.3 具体实现	278	11.12.2 具体实现	291
11.3 捕鱼和分鱼	279	11.13 常胜将军问题	291
11.3.1 问题描述	279	11.13.1 问题描述	291
11.3.2 算法分析	279	11.13.2 算法分析	292
11.3.3 具体实现	279	11.13.3 具体实现	292
11.4 出售金鱼	280	11.14 背包问题	292
11.4.1 问题描述	280	11.14.1 使用动态规划法解决 “背包”问题	292
11.4.2 算法分析	280	11.14.2 使用递归法解决“背包” 问题	293
11.4.3 具体实现	280	11.15 野人与传教士问题	295
11.5 平分七筐鱼	280	11.15.1 问题描述	295
11.5.1 问题描述	280	11.15.2 算法分析	295
11.5.2 算法分析	281	11.15.3 具体实现	295
11.5.3 具体实现	281	11.16 三色旗问题	296
11.6 绳子的长度和井深	281	11.16.1 问题描述	296
11.6.1 问题描述	281	11.16.2 算法分析	296
11.6.2 算法分析	282	11.16.3 具体实现	297
11.6.3 具体实现	282	11.17 猴子分桃	297
11.7 鸡兔同笼	282	11.17.1 问题描述	297
11.7.1 问题描述	282	11.17.2 算法分析	298
11.7.2 算法分析	283	11.17.3 具体实现	298
11.7.3 具体实现	283	11.18 将老师随机分配到办公室	298
11.8 汉诺塔	283	11.18.1 问题描述	298
11.8.1 问题描述	283	11.18.2 具体实现	299
11.8.2 算法分析	284	11.19 龙的世界	300
11.8.3 具体实现	284	11.19.1 问题描述	300
11.9 马踏棋盘	285	11.19.2 具体实现	300
11.9.1 使用递归法	285	11.20 凯撒密码游戏	301
11.9.2 贪婪、递归、迭代三种 算法的对比	287	11.20.1 问题描述	301
11.10 三色球问题	289	11.20.2 算法分析	301
11.10.1 问题描述	289	11.20.3 具体实现	301
11.10.2 算法分析	289	第 12 章 图像问题	303
11.10.3 具体实现	289	12.1 生命游戏	304
11.11 计算年龄	290	12.1.1 问题描述	304
11.11.1 问题描述	290		

12.1.2	算法分析	304	12.5.1	问题描述	317
12.1.3	具体实现	304	12.5.2	算法分析	317
12.2	黑白棋问题	306	12.5.3	具体实现	317
12.2.1	问题描述	306	12.6	绘制 Hangman 游戏	319
12.2.2	算法分析	307	12.6.1	问题描述	319
12.2.3	具体实现	307	12.6.2	算法分析	319
12.3	马踏棋盘(骑士周游问题)	311	12.6.3	具体实现	320
12.3.1	问题描述	311	第 13 章 游戏和算法	324	
12.3.2	算法分析	311	13.1	开发一个俄罗斯方块游戏	325
12.3.3	具体实现	311	13.1.1	规划图形	325
12.4	井字棋问题	313	13.1.2	具体实现	325
12.4.1	问题描述	313	13.2	跑酷游戏	332
12.4.2	算法分析	313	13.3	水果连连看游戏	337
12.4.3	具体实现	313	13.4	AI 智能贪吃蛇游戏	341
12.5	用蒙特卡罗方法验证凯利 公式	317	13.5	AI 智能五子棋游戏	346

第1章

算法概述

算法是程序的灵魂，只有掌握了算法，才能轻松地驾驭程序开发。软件开发工作不是按部就班，而是选择一种最合理的算法去实现项目功能。算法能够引导开发者在面对一个项目功能时用什么思路去实现，有了这个思路后，编程工作只需要遵循这个思路去实现即可。本章将详细讲解计算机算法的基础知识，为读者步入后面的学习打下基础。



1.1 算法的基础

自然界中的很多事物并不是独立存在的，而是和许多其他事物有着千丝万缕的联系。就拿算法和编程来说，两者之间就有着必然的联系。在编程界有一个不成文的原则，要想学好编程，就必须学好算法。要想获悉这一说法的原因，先看下面对两者的定义。

算法是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制。也就是说，能够对符合一定规范的输入，在有限时间内获得所要求的输出。如果一个算法有缺陷，或不适合于某个问题，执行这个算法将不会解决这个问题。不同的算法可能用不同的时间、空间或效率来完成同样的任务。

编程是让计算机为解决某个问题而使用某种程序设计语言编写程序代码，并最终得到结果的过程。为了使计算机能够理解人的意图，人类就必须将需要解决的问题的思路、方法和手段通过计算机能够理解的形式“告诉”计算机，使计算机能够根据人的指令一步一步去工作，完成某种特定的任务。编程的目的是实现人和计算机之间的交流，整个交流过程就是编程。

在上述对编程的定义中，核心内容是思路、方法和手段等，这都需要用算法来实现。由此可见，编程的核心是算法，只要算法确定了，后面的编程工作只是实现算法的一个形式而已。

1.1.1 算法的特征

在1950年，算法(Algorithm)一词经常同欧几里得算法联系在一起。这个算法就是在欧几里得的《几何原本》中所阐述的求两个数的最大公约数的过程，即辗转相除法。从此以后，算法这一叫法一直沿用至今。

随着时间的推移，算法这门学科得到了长足的发展，算法应该具有如下5个重要的特征。

- 有穷性：保证执行有限步骤之后结束。
- 确切性：每一步骤都有确切的定义。
- 输入：每个算法有零个或多个输入，以刻画运算对象的初始情况。所谓零个输入，是指算法本身舍弃了初始条件。
- 输出：每个算法有一个或多个输出，显示对输入数据加工后的结果，没有输出的算法是毫无意义的。
- 可行性：原则上算法能够精确地运行，进行有限次运算后即可完成一种运算。

1.1.2 何为算法

为了理解什么是算法，先看一道有趣的智力题。

“烧水泡茶”有如下5道工序：①烧开水，②洗茶壶，③洗茶杯，④拿茶叶，⑤泡茶。

烧开水、洗茶壶、洗茶杯、拿茶叶是泡茶的前提。其中，烧开水需要15min，洗茶壶需要2min，洗茶杯需要1min，拿茶叶需要1min，泡茶需要1min。

下面是“烧水泡茶”的两种方法。

方法1的步骤如下。

第1步：烧水。

第2步：水烧开后，洗刷茶具，拿茶叶。

第3步：沏茶。