Broadview®
www.broadview.com.cn

让你轻松学会、爱不释手的Python算法书，
有故事，有思路，有深度！

你也能看得懂的Python算法书

王硕 董文馨 张舒行 张洁◎编著

很资深： 凝聚4位作者10年编程经验，带你领悟算法的精髓
很有趣： 全书采用生动风趣的语言，让算法不再难学
很实战： 全书包括36个实例，9大类算法，很有实战价值
很图解： 对每种算法，都给出了图解说明，一目了然

Python

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

你也能看得懂的Python算法书

王硕 董文馨 张舒行 张洁◎编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

编程的核心是算法，学习算法不仅能教会你解决问题的方法，而且还能为你今后的发展提供一种可能。

本书面向算法初学者，首先介绍当下流程的编程语言 Python，详细讲解 Python 语言中的变量和循序、分支、循环三大结构，以及列表和函数的使用，为之后学习算法打好基础。然后以通俗易懂的语言讲解双指针、哈希、深度优先、广度优先、回溯、贪心、动态规划和最短路径等经典算法。

本书适合有一定编程基础的算法爱好者阅读。

图书在版编目（CIP）数据

你也能看得懂的 Python 算法书/王硕等编著.—北京：电子工业出版社，2018.11
ISBN 978-7-121-35255-3

Ⅰ. ①你… Ⅱ. ①王… 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 240979 号

策划编辑：张月萍
责任编辑：刘　舫
印　　刷：三河市君旺印务有限公司
装　　订：三河市君旺印务有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×980　1/16　　印张：16.25　　字数：352 千字
版　　次：2018 年 11 月第 1 版
印　　次：2018 年 11 月第 1 次印刷
印　　数：3500 册　　定价：59.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

前言

为什么要写这本书

算法是编程的核心，就像一台计算机的 CPU，算法的好坏决定了一个系统的效率高低。

许多人认为学习编程就是学习最新的编程语言、技术和框架，其实计算机算法更重要。计算机语言和技术日新月异，但万变不离其宗的是算法。修炼好算法这门“内功”，再辅以新技术这些“招式”，才能独霸“武林”。这也是为什么像 Google 和 Facebook 这类大公司在面试中主要会考查算法问题的原因。

目前图书市场上关于算法的图书不少，经典的如《算法导论》。但是大多数算法图书太学术、太复杂，对于初学者来说门槛太高。学习算法本身就不是一件容易的事情，再加上复杂的场景和数学理论，会让算法的学习曲线更陡。因此本书的作者们就萌生了写一本让大家都能看懂的算法书的想法，以生动的语言把算法的思想过程写出来，让学习算法不再那么枯燥。

在本书的创作过程中，王硕老师迎来了人生的第一个宝宝小朗朗。于是几位作者就有了一个心愿，希望青少年朋友也学会算法，因此，在描述算法问题时，尽量用简单、通俗的形式表述，使青少年朋友也能看懂，也希望如此这般能够增加读者的学习兴趣。既然是学习算法，免不了要写代码，对于编程语言的选择，我们选择了 Python 这门简单易懂的语言作为本书的编程语言。对于初学编程的人来说，Python 可以缩短学习编程语言的时间和降低学习编程的难度。

学习算法不易，且行且珍惜。

本书有何特色

1. 以孩子的口吻，生动形象地讲解算法，提高趣味性

为了便于读者理解本书内容，提高学习效率，大部分问题都以孩子的口吻来讲解，以解决小朗朗的实际问题作为出发点引出问题，增加了趣味性和可读性。

2. 涵盖核心算法知识点

本书涵盖双指针问题、哈希、深度优先遍历、广度优先遍历、回溯、贪心、动态规划、最短路径问题、分治等 9 大算法，帮助读者全面掌握核心算法的知识点。

3. 以 Python 语言作为载体，降低学习难度

抛弃其他复杂的编程语言，本书采用简单的编程语言 Python 作为算法的载体，并在第 1 章介绍了 Python 语言的语法。

4. 选择经典算法的经典问题，有较高的通用性

本书在简单介绍 Python 编程语言以后，选择了 9 大经典算法，重点讲解算法原理，并选择经典问题进行有针对性的练习。

5. 提供完善的技术支持和售后服务

本书提供了专门的邮箱供读者咨询：317977682@qq.com。读者在阅读本书的过程中有任何疑问都可以通过该邮箱获得帮助。

本书内容及知识体系

第 1 章　编程基础

掌握一门编程语言是学习算法的基础。学习编程语言是为了与计算机“交流”，只有正确的格式才能被计算机成功识别。学习编程语言的起点就是了解这门语言的语法。本书使用 Python 语言进行算法讲解，本章主要讲解 Python 3 的编程语法。

第 2 章　双指针问题

“指针”是编程语言中的一个对象，它存储着一个内存空间的地址，计算机可以通过这个地址找到变量的值。也就是说，这个特定的地址指向这个特定的值。指针最大的优点在于它可以有效利用零碎的内存空间。

第 3 章　哈希算法

哈希算法又称散列函数算法，是一种查找算法，简单来说，就是把一些复杂的数据，通过某种函数映射关系，映射成更加易于查找的方式。但是这种映射关系有可能会发生多个关键字映射到同一地址的现象，我们称之为冲突。在这种特殊情况下，需要对关键字进行第二次或更多次的处理，在其他的大多数情况下，哈希算法可以实现在常数时间内存储和查找这些关键字。

第 4 章　深度优先遍历算法

深度优先遍历算法是经典的图论算法，它的搜索逻辑就和它的名字一样，只要有可能，就尽量深入搜索，直到找到答案，或者尝试了所有可能后确定没有解。

第 5 章　广度优先遍历算法

广度优先遍历算法与深度优先遍历算法类似，也是查询的方法之一，它从某个状态出发查询可以到达的所有状态。但不同于深度优先遍历，广度优先遍历算法总是先去查询距离初始状态最近的状态。

第 6 章　回溯算法

回溯算法可以被看作走迷宫，因为我们不知道出口在哪里，所以只能不断地深入，尝试不同的路线。一旦找到了出口便可以回溯到起点，辨清路线。

第 7 章　贪心算法

贪心算法就是遵循某种既定原则，不断地选取当前条件下最优的选择来构造每个子步骤的解决方案，直到获得问题最终的解。即在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，它所做的仅是在某种意义上的局部最优解。

第 8 章　动态规划算法

动态规划算法将待求解问题拆分成一系列相互交叠的子问题，通过递推关系定义各子问题的求解策略，并随时记录子问题的解，最终获得原始问题的解，避免了对交叠子问题的重复求解。

第 9 章　最短路径问题

把地点看成节点，把道路看成边，整个地图就可看成一个加权图。计算加权图中两点间的最短路径是编程的一个重要问题，这一章我们会用以下几个算法解决这个问题：迪可斯特朗算法、Floyd 算法、A*算法。

第 10 章　分治算法

分治算法的核心思想是把一个规模很大的问题化简为多个规模较小的问题，这些子问题虽然独立而不同，但是问题的本质是一致的，从而达到分而治之的目的。

适合阅读本书的读者

- 需要全面学习算法的人员
- 需要学习 Python 的程序员
- 对编程算法感兴趣的人员
- 希望提高算法水平的程序员
- 专业培训机构的学员

阅读本书的建议

- 没有 Python 基础的读者，建议从第 1 章开始顺次阅读并演练每个实例。
- 有一定 Python 基础的读者，可以根据实际情况有重点地选择阅读各个模块和项目案例。
- 对于每个模块和案例，先自己思考一下实现的思路，然后再阅读，学习效果会更好。这样理解起来也会更加容易、更加深刻。

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **提交勘误**：您对书中内容的修改意见可在*提交勘误*处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动**：在页面下方*读者评论*处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：http://www.broadview.com.cn/35255

目录

第 1 章
编程基础

掌握一门编程语言是学习算法的基础。学习编程语言是为了与计算机“交流”，只有正确的格式才能被计算机成功识别。学习编程语言的起点就是了解这门语言的语法。本书使用 Python 语言进行算法讲解，本章主要讲解 Python 3 的编程语法，涉及的主要内容有如下几项。

- 变量：变量的类型和操作。
- 三大结构：循序、分支和循环。
- 数组：数组的概念和使用。
- 函数：函数的定义与调用。

1.1 变量

变量在计算机语言中，可以用于存储不同的计算结果和不同类型的数据，或表示一个数值。变量是可变的，也就是说，它可以被重复赋值。

在认识变量之前，我们先来了解输入和输出，这样才能对程序输入数据或者看到运行后的结果。

1.1.1 输出和输入

我们以非常经典的一项任务入手，在屏幕上输出一行文字：Hello World!

在 Python 3 中，实现这个任务只需要一行代码（Python 中“#”后方的文字为注释，不影响程序运行，详细信息见 1.1.5 节最后）。

```
01    print("Hello World!")                    #注意：编写程序时一定要用英文输入法！
```

运行程序，输出结果为：

```
    Hello World!
```

在这一行程序中，print()是用于输出的函数，“Hello World!”是输出的数据。其中双引号代表数据是一个字符串。

我们再来看如何输入数据。如果想输入一个数字，把它除以 2 然后打印出来，该怎么做呢？下面的程序就实现了这个功能。

```
01    num = int(input("Enter a number:"))     #输入一个数字，用变量 num 存储
02    num = num/2                             #把 num 除以 2
03    print(num)                              #输出 num 的值
```

运行程序，程序输出如下：

```
Enter a number:
```

此时可以输入任意整数。我们输入 6，程序输出 3 后结束。

在程序的第一行中，新建了变量 num 来存储输入的数据。因为这个变量是用于存储一个数的，所以把它命名为 num，这样比较直观易懂。只要遵守变量的命名法则（参见本小节末尾），那么可以随意为变量取名。

在第一行中，input()函数用于输入数据，括号内的字符串是提示语，就是在输入数据之前输出的那一段字符串。如果括号内没有内容（input()），则不输出任何文字，用户直接输入数据即可。而 input()函数外面套着的 int()函数则是用于把数据转换为 int 类型的函数。Input()函数返回的值默认为字符串类型（string），字符串不可以进行数学运算。所以，我们使用 int()函数先把字符串转换为整型（int），再进行赋值、运算并输出。数据类型的知识会在下一小节介绍。

程序中的“=”用于给变量赋值，不是相等的意思。“=”左边必须是一个可以被赋值的变量，“=”右边可以是一个常量、变量或函数的返回值等。赋值语句运行结束后，“=”左边的变量的值更新为“=”右边的值。与其他编程语言不同，由于 Python 中变量的类型不固定，所以被赋值的变量的数据类型同样可以被更新，如下面的程序所示。

```
variable = 0                    #给变量赋值 0，数据类型为整型（int）
variable = "abcd"               #再给变量赋值"abcd"，数据类型为字符串（string）
print(variable)                 #输出变量
```

运行这段程序，输出结果为 abcd。

在这一小节中我们了解了输入和输出，也对变量的概念有了初步认识。为了避免程序出错，以下是变量命名的几项法则：

- 变量名中只能出现字母、数字和下画线，且第一个字符不能是数字。比如，可以将变量命名为 name_6，但不能将变量命名为 6_name。
- 不要把在 Python 中有特殊用途的单词作为变量名，如函数名。比如，目前我们接触过的 print 函数，如果把 print 作为一个变量名称，运行程序时又用到 print 函数，编译器就会报错。

  下面就是一个例子：

```
print = 0
print(print)                    #print 函数无法被调用，print 被系统认为是变量名称了
```

- 一个变量名对应一个变量，不会有两个变量有相同的变量名。变量的名字最好和变量的功能相对应，比如一个存储平均数的变量，将它命名为 average 比命名为 abcd 要直观，这样不容易记错变量名称而导致程序出错。

1.1.2　简单变量类型

Python 语言中有 5 种标准变量类型：数字、字符串、列表、元组和字典，其中数字和字符串是比较简单的，本小节主要介绍这两种变量类型。

首先，我们来仔细了解一下变量的概念。每个变量都有一个独特的变量名，我们通过这个独特的变量名来调用这个变量的值。就像图 1.1 中的气球和牌子一样，人够不着气球，但是可以通过扯动牌子来拿到气球。每个牌子都对应一个单独的气球。牌子就相当于变量名，而气球就相当于变量存储的值。

现在我们知道了变量的概念，再来分类型介绍变量。

数字类型的变量是最常用的一种。数字类型还可以再细分，其中主要的两种类型为整型（int）和浮点型（float）。整型变量只能存储整数，浮点型变量则可以存储小数。