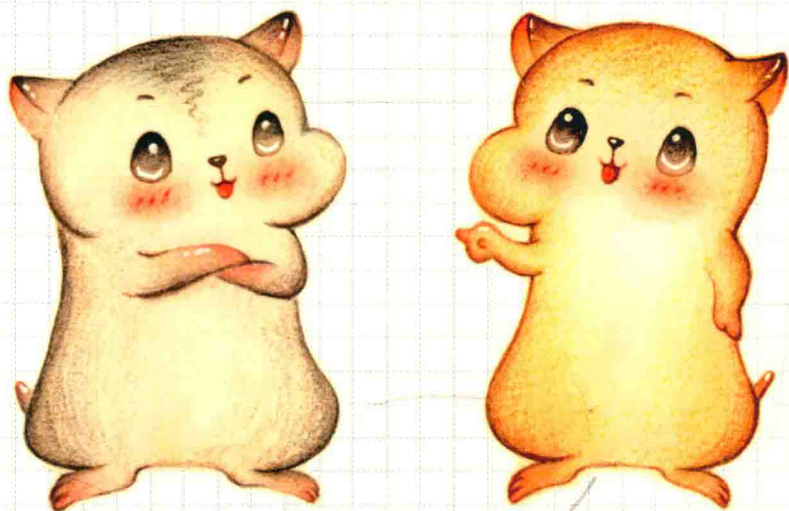


全网阅读量近1000万次的算法故事
一群可爱的小仓鼠，带你轻松入门算法和数据结构！



漫画算法

小灰的算法之旅♡♡

魏梦舒 (@程序员小灰) 著

漫画算法

小灰的算法之旅^{♡♡}

魏梦舒 (@程序员小灰) 著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书通过虚拟的主人公小灰的心路历程，用漫画的形式讲述了算法和数据结构的基础知识、复杂多变的算法面试题目及算法的实际应用场景。

第1章介绍了算法和数据结构的相关概念，告诉大家算法是什么，数据结构又是什么，它们有哪些用途，如何分析时间复杂度，如何分析空间复杂度。

第2章介绍了最基本的数据结构，包括数组、链表、栈、队列、哈希表的概念和读写操作。

第3章介绍了树和二叉树的概念、二叉树的各种遍历方式、二叉树的特殊形式——二叉堆和优先队列的应用。

第4章介绍了几种典型的排序算法，包括冒泡排序、快速排序、堆排序、计数排序、桶排序。

第5章介绍了10余道职场上流行的算法面试题及详细的解题思路。例如怎样判断链表有环、怎样计算大整数相加等。

第6章介绍了算法在职场上的一些应用，例如使用LRU算法来淘汰冷数据，使用Bitmap算法来统计用户特征等。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目(CIP)数据

漫画算法：小灰的算法之旅 / 魏梦舒著. — 北京：电子工业出版社，2019.5
ISBN 978-7-121-36197-5

I. ①漫… II. ①魏… III. ①算法分析②数据结构 IV. ①TP301.6②TP311.12

中国版本图书馆CIP数据核字(2019)第057355号

策划编辑：张月萍

责任编辑：牛勇

印刷：北京富诚彩色印刷有限公司

装订：北京富诚彩色印刷有限公司

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：720×1000 1/16

印张：17.5

字数：404千字

版次：2019年5月第1版

印次：2019年6月第2次印刷

印数：20001~28000册 定价：79.00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。



魏梦舒 (@程序员小灰)

微信公众号“程序员小灰”的作者，具有多年软件行业从业经验，先后在京东金融、摩拜科技从事研发工作，对算法有一定的兴趣和经验。



关注微信公众号“程序员小灰”，
回复“漫画算法”，
即可获得本书的程序代码。

推荐序

P REFACE

初识小灰是因为在他的微信公众号看到一篇讲动态规划的文章，当时觉得挺意外，没想到还能有人用漫画来解释动态规划算法。

所谓算法，其实是个很宽泛的概念。有理解起来难度超大，烧脑到要“爆炸”的；也有简单直接，一目了然的；更多的却是，虽然看起来复杂，但只要方法得当，搞清原理，掌握起来还是很容易的那种算法。

可是很多人被“算法”二字“狰狞”的外表吓住了，久久不敢接触它。好不容易斗胆翻翻算法书，结果看到的不是大篇大篇的代码，就是乱七八糟的符号。这都是什么呀？算了，看来是学不会算法了，放弃吧……

但凡书籍文章，最难读的，肯定是公式符号；而最好读的，无外乎图像、对话等。本书作者以可爱的小灰和大黄两个漫画形象为主人公，把对算法的描述过程嵌入到它们的对话之中，并辅之以图形等直观方式来表达数据结构和操作步骤——这种表达形式带着天然的亲和力，完全没有计算机背景的读者读来也不觉得生硬。

小灰所做的事情，就是给算法这颗“炮弹”包上了“糖衣”，让算法的威力潜藏于内，外表不再吓人，反而变得萌萌哒，Q弹可爱，清新怡人。

先干为敬，让我们一起吞了这颗包着“炸药”的“糖丸”吧！

李焯，微软高级软件工程师

许多程序员对算法望而生畏，认为算法是一门高深莫测的学问。

以前我曾经面试过一个求职者，起初考查他的技术功底和项目经验，他都回答得不错。接下来我对他说：“OK，那我考查一下你的算法水平吧。”

题目还没说出口，该求职者立马摆摆手说：“不要不要，我算法不行的！”

我还是有些不甘心，接着说道：“我只考查最基础的，你说说冒泡排序的基本思路吧！”

他仍旧说：“我不知道，我算法一点都不会……”

算法真的那么难，真的那么无趣吗？

恰恰相反，算法是编程领域中最有意思的一块内容，也不像许多人想象的那样难以驾驭。

许多人把算法比作程序员的“内功”，但笔者觉得这个比喻并不是很恰当。内功实实在在，没有任何巧妙可言，而算法天马行空，千变万化，就像金庸笔下令狐冲的一套独孤九剑。

学习算法，我们不需要死记硬背那些冗长复杂的背景知识、底层原理、指令语法……需要做的是领悟算法思想、理解算法对内存空间和性能的影响，以及开动脑筋去寻求解决问题的最佳方案。相比编程领域的其他技术，算法更纯粹，更接近数学，也更具有趣味性。

我一直希望写出一些东西，让更多的IT同行能够领略到算法的魅力，可是用什么方式来写呢？

2016年9月，一次突如其来的灵感让我创造了一个初出茅庐的菜鸟程序员形象，这个菜鸟程序员名叫小灰。

程序员小灰的故事活跃在同名的微信公众号上，该公众号用漫画的形式诉说着小灰一次又一次的面试经历，倔强的小灰屡战屡败，屡败屡战。小灰是我刚刚入行时的真实写照，相信许多程序员也能从中看到自己的影子。

终于，在朋友们的支持和鼓励下，程序员小灰的故事从微信公众号搬到了纸质图书上。能让更多同行看到小灰的故事，我感到十分欣慰。

本书特色

这本书通过漫画的形式，讲述了小灰学习算法和数据结构知识的心路历程。书中许多内容源于本人的微信公众号，但是比公众号上所展现的内容更加系统、全面，也更加严谨。

本书的前4章是对算法基础知识的讲解，没有算法和数据结构基础的读者可以从头开始进行系统学习。

对于有一定基础的读者，也可以选择从第5章面试题的讲解开始阅读，每一道面试题都是相对独立的，并不需要严格地按顺序学习。同时，也推荐大家适当看看前面的内容，巩固一下自己的算法知识体系。

这不是一本编程入门书。在编程方面完全零基础的读者，建议至少先了解一门编程语言。

这也不是一本局限于某个编程语言的书籍，虽然书中的代码示例都是用Java来实现的，但算法思想是相通的。在实现代码时，书中尽可能规避了Java语言的特殊语法和工具类，相信熟悉其他语言的开发者也不难看懂。

勘误和支持

除书中所提供的代码示例以外，大家也可以关注微信公众号“程序员小灰”，在后台回复“漫画算法”，获得全书完整的、可运行的代码。为了保证代码的简洁，在部分代码实现中省略了烦琐的参数判空和验证逻辑。

由于作者水平有限，书中难免会出现一些错误，恳请广大读者批评指正。读者如果在阅读过程中产生疑问或发现Bug，欢迎随时到微信公众号的后台留言。“程序员小灰”微信公众号二维码如下。



致谢

感谢微信公众号“程序员小灰”的读者。你们的鼓励和支持，给了我坚持创作的动力。

感谢成都道然科技有限责任公司的姚新军老师。有了他的肯定、支持和指导意见，才有了这本书的正式出版。

感谢朴提、单耳和康慧三位插画师所画的精彩插画，是你们让小灰的形象更丰满、更可爱。感谢为本书审稿的杨道谈先生，感谢为本书写序的李焯老师，感谢在百忙之中阅读书稿并写书评的专家们，他们是刘欣、张洪亮、安晓辉、李艳鹏、翟永超等。

特别感谢我的父母，是他们把我带进了数学的大门。在我上小学的时候，是他们的坚持，才让我有机会学习奥数，参加数学竞赛，并对数学和逻辑产生了兴趣。在这本书的写作过程中，又是他们辛苦努力屏蔽生活琐事对我的干扰，让我能够全身心地投入到本书的写作当中。

谨以此书献给我的家人，我的读者，以及热爱编程的朋友们！

魏梦舒，微信公众号“程序员小灰”的作者

第1章 算法概述 / 1

- 1.1 算法和数据结构 / 1
 - 1.1.1 小灰和大黄 / 1
 - 1.1.2 什么是算法 / 3
 - 1.1.3 什么是数据结构 / 7
- 1.2 时间复杂度 / 8
 - 1.2.1 算法的好与坏 / 8
 - 1.2.2 基本操作执行次数 / 10
 - 1.2.3 渐进时间复杂度 / 12
 - 1.2.4 时间复杂度的巨大差异 / 15
- 1.3 空间复杂度 / 16
 - 1.3.1 什么是空间复杂度 / 16
 - 1.3.2 空间复杂度的计算 / 19
 - 1.3.3 时间与空间的取舍 / 21
- 1.4 小结 / 22

第2章 数据结构基础 / 23

- 2.1 什么是数组 / 23
 - 2.1.1 初识数组 / 23
 - 2.1.2 数组的基本操作 / 26
 - 2.1.3 数组的优势和劣势 / 32
- 2.2 什么是链表 / 33
 - 2.2.1 “正规军”和“地下党” / 33
 - 2.2.2 链表的基本操作 / 35
 - 2.2.3 数组VS链表 / 41

- 2.3 栈和队列 / 42
 - 2.3.1 物理结构和逻辑结构 / 42
 - 2.3.2 什么是栈 / 43
 - 2.3.3 栈的基本操作 / 44
 - 2.3.4 什么是队列 / 45
 - 2.3.5 队列的基本操作 / 46
 - 2.3.6 栈和队列的应用 / 50
- 2.4 神奇的散列表 / 51
 - 2.4.1 为什么需要散列表 / 51
 - 2.4.2 哈希函数 / 54
 - 2.4.3 散列表的读写操作 / 55
- 2.5 小结 / 59

第3章 树 / 61

- 3.1 树和二叉树 / 61
 - 3.1.1 什么是树 / 61
 - 3.1.2 什么是二叉树 / 64
 - 3.1.3 二叉树的应用 / 67
- 3.2 二叉树的遍历 / 71
 - 3.2.1 为什么要研究遍历 / 71
 - 3.2.2 深度优先遍历 / 73
 - 3.2.3 广度优先遍历 / 84
- 3.3 什么是二叉堆 / 88
 - 3.3.1 初识二叉堆 / 88
 - 3.3.2 二叉堆的自我调整 / 90
 - 3.3.3 二叉堆的代码实现 / 95
- 3.4 什么是优先队列 / 98
 - 3.4.1 优先队列的特点 / 98
 - 3.4.2 优先队列的实现 / 99
- 3.5 小结 / 103

第4章 排序算法 / 105

- 4.1 引言 / 105
- 4.2 什么是冒泡排序 / 107
 - 4.2.1 初识冒泡排序 / 107
 - 4.2.2 冒泡排序的优化 / 110
 - 4.2.3 鸡尾酒排序 / 114
- 4.3 什么是快速排序 / 118
 - 4.3.1 初识快速排序 / 118
 - 4.3.2 基准元素的选择 / 120
 - 4.3.3 元素的交换 / 122
 - 4.3.4 单边循环法 / 125
 - 4.3.5 非递归实现 / 128
- 4.4 什么是堆排序 / 131
 - 4.4.1 传说中的堆排序 / 131
 - 4.4.2 堆排序的代码实现 / 134
- 4.5 计数排序和桶排序 / 137
 - 4.5.1 线性时间的排序 / 137
 - 4.5.2 初识计数排序 / 138
 - 4.5.3 计数排序的优化 / 140
 - 4.5.4 什么是桶排序 / 145
- 4.6 小结 / 148

第5章 面试中的算法 / 150

- 5.1 踌躇满志的小灰 / 150
- 5.2 如何判断链表有环 / 151
 - 5.2.1 一场与链表相关的面试 / 151
 - 5.2.2 解题思路 / 155
 - 5.2.3 问题扩展 / 158
- 5.3 最小栈的实现 / 161
 - 5.3.1 一场关于栈的面试 / 161
 - 5.3.2 解题思路 / 163

- 5.4 如何求出最大公约数 / 166
 - 5.4.1 一场求最大公约数的面试 / 166
 - 5.4.2 解题思路 / 168
- 5.5 如何判断一个数是否为2的整数次幂 / 173
 - 5.5.1 一场很“2”的面试 / 173
 - 5.5.2 解题思路 / 175
- 5.6 无序数组排序后的最大相邻差 / 177
 - 5.6.1 一道奇葩的面试题 / 177
 - 5.6.2 解题思路 / 179
- 5.7 如何用栈实现队列 / 184
 - 5.7.1 又是一道关于栈的面试题 / 184
 - 5.7.2 解题思路 / 186
- 5.8 寻找全排列的下一个数 / 191
 - 5.8.1 一道关于数字的题目 / 191
 - 5.8.2 解题思路 / 193
- 5.9 删去 k 个数字后的最小值 / 196
 - 5.9.1 又是一道关于数字的题目 / 196
 - 5.9.2 解题思路 / 198
- 5.10 如何实现大整数相加 / 205
 - 5.10.1 加法,你会不会 / 205
 - 5.10.2 解题思路 / 206
- 5.11 如何求解金矿问题 / 211
 - 5.11.1 一个关于财富自由的问题 / 211
 - 5.11.2 解题思路 / 213
- 5.12 寻找缺失的整数 / 223
 - 5.12.1 “五行”缺一个整数 / 223
 - 5.12.2 问题扩展 / 225

第6章 算法的实际应用 / 230

- 6.1 小灰上班的第1天 / 230
- 6.2 Bitmap的巧用 / 232

漫画算法：小灰的算法之旅

- 6.2.1 一个关于用户标签的需求 / 232
- 6.2.2 用算法解决问题 / 234
- 6.3 LRU算法的应用 / 241
 - 6.3.1 一个关于用户信息的需求 / 241
 - 6.3.2 用算法解决问题 / 243
- 6.4 什么是A星寻路算法 / 249
 - 6.4.1 一个关于迷宫寻路的需求 / 249
 - 6.4.2 用算法解决问题 / 251
- 6.5 如何实现红包算法 / 262
 - 6.5.1 一个关于钱的需求 / 262
 - 6.5.2 用算法解决问题 / 264
- 6.6 算法之路无止境 / 268

第1章

算法概述

1.1 算法和数据结构

1.1.1 小灰和大黄

在大四临近毕业时，计算机专业的同学大都收到了满意的offer，可是小灰却还在着急上火。虽然他这几天面试了很多家IT公司，可每次都被面试官“虐”得很惨很惨。



就在心灰意冷之际，小灰忽然想到，他们系里有一位学霸名叫大黄，大黄不但技术很强，而且很乐意帮助同学。于是，小灰赶紧去找大黄，希望能够得到一些指点。



小灰，听说你昨天又去面试了，结果怎么样？





唉，还不是被面试官给“虐”了？面试官说我算法和数据结构基础太差……



对程序员来说，算法和数据结构是很重要的基础知识，一定要好好掌握啊！



谁说不是啊？可我当初所学的都还给老师了……大黄，能不能给我补补算法和数据结构有关的知识？我请你吃大餐！



好吧，好吧，我们就从最基础的知识来讲解，你可要认真听哦。



1.1.2 什么是算法

算法，对应的英文单词是algorithm，这是一个很古老的概念，最早来自数学领域。有一个关于算法的小故事，估计大家都有耳闻。

在很久很久以前，曾经有一个顽皮又聪明的“熊孩子”，天天在课堂上调皮捣蛋。终于有一天，老师忍无可忍，对“熊孩子”说：

臭小子，你又调皮啊！今天罚你算加法，算出 $1+2+3+4+5+6+7\cdots$
一直加到10000的结果，算不完不许回家！



嘿嘿，我算就是了。

老师以为，“熊孩子”会按部就班地一步一步计算，就像下面这样。

$$\begin{aligned} 1 + 2 &= 3 \\ 3 + 3 &= 6 \\ 6 + 4 &= 10 \\ 10 + 5 &= 15 \\ &\cdots \end{aligned}$$

这还不得算到明天天亮？够这小子受的！老师心里幸灾乐祸地想着。谁知仅仅几分钟后……



老师，我算完了！结果是50 005 000，对不对？

这、这、这……你小子怎么算得这么快？我读书多，你骗不了我的！



看着老师惊讶的表情，“熊孩子”微微一笑，讲出了他的计算方法。首先把从1到10 000这10 000个数字两两分组相加，如下。

$$\begin{aligned} 1 + 10\,000 &= 10\,001 \\ 2 + 9999 &= 10\,001 \\ 3 + 9998 &= 10\,001 \\ 4 + 9997 &= 10\,001 \\ &\cdots \end{aligned}$$

一共有多少组这样结果相同的和呢？有 $10\,000 \div 2$ 即5000组。所以1到10 000相加的总和可以这样来计算：

$$(1+10\,000) \times 10\,000 \div 2 = 50\,005\,000$$

这个“熊孩子”就是后来著名的犹太数学家约翰·卡尔·弗里德里希·高斯，而他所采用的这种等差数列求和的方法，被称为高斯算法。（上文的故事情节与史实略有出入。）



这是数学领域中算法的一个简单示例。在数学领域里，算法是用于解决某一类问题的公式和思想。

而本书所涉及的算法，是计算机科学领域的算法，它的本质是一系列程序指令，用于解决特定的运算和逻辑问题。

从宏观上来看，数学领域的算法和计算机领域的算法有很多相通之处。

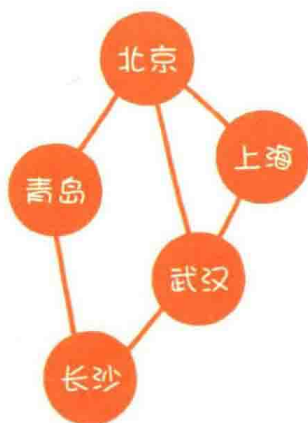
算法有简单的，也有复杂的。

简单的算法，诸如给出一组整数，找出其中最大的数。



Max=?

复杂的算法，诸如在多种物品里选择装入背包的物品，使背包里的物品总价值最大，或找出从一个城市到另一个城市的最短路线。



算法有高效的，也有拙劣的。

刚才所讲的从1加到10000的故事中，高斯所用的算法显然是更加高效的算法，它利用等差数列的规律，四两拨千斤，省时省力地求出了最终结果。

而老师心中所想的算法，按部就班地一个数一个数进行累加，则是一种低效、笨拙的算法。虽然这种算法也能得到最终结果，但是其计算过程要低效得多。

在计算机领域，我们同样会遇到各种高效和拙劣的算法。衡量算法好坏的重要标准有两个。

- 时间复杂度
- 空间复杂度

具体的概念会在本章进行详细讲解。

算法的应用领域多种多样。

算法可以应用在很多不同的领域中，其应用场景更是多种多样，例如下面这些。

1. 运算

有人或许会觉得，不就是数学运算吗？这还不简单？

其实还真不简单。

例如求出两个数的最大公约数，要做到效率的极致，的确需要动一番脑筋。

再如计算两个超大整数的和，按照正常方式来计算肯定会导致变量溢出。这又该如何求解呢？

$$\begin{array}{r}
 4\ 2\ 6\ 7\ 0\ 9\ 7\ 5\ 2\ 3\ 1\ 8 \\
 +\ 9\ 5\ 4\ 8\ 1\ 2\ 5\ 3\ 1\ 2\ 9 \\
 \hline
 5\ 2\ 2\ 1\ 9\ 1\ 0\ 0\ 5\ 4\ 4\ 7
 \end{array}$$

2. 查找

当你使用谷歌、百度搜索某一个关键词，或在数据库中执行某一条SQL语句时，你有没有思考过数据和信息是如何被查出来的呢？

