

最短路径

Dijkstra

树

Bellman-Ford

图

深度优先

啊哈磊
著

栈

广度优先

堆

快速排序

并查集

啊哈！去中科院玩单片机
呦吼！在微软亚洲研究院写爬虫

啊哈！算法

Aha! Algorithms

哒哒！写一本开开心心的算法书
你一定能看懂的算法书！

TURING
图灵原创

啊哈磊
著

啊哈!算法

Aha! Algorithms

人民邮电出版社
北京

图书在版编目 (C I P) 数据

啊哈! 算法 / 啊哈磊著. — 北京: 人民邮电出版社, 2014. 6

(图灵原创)

ISBN 978-7-115-35459-4

I. ①啊… II. ①啊… III. ①电子计算机—算法理论
IV. ①TP301.6

中国版本图书馆CIP数据核字(2014)第094155号

内 容 提 要

这是一本充满智慧和趣味的算法入门书。没有枯燥的描述, 没有难懂的公式, 一切以实际应用为出发点, 通过幽默的语言配以可爱的插图来讲解算法。你更像是去阅读一个个轻松的小故事或是在玩一把趣味解谜游戏, 在轻松愉悦中便掌握算法精髓, 感受算法之美。

本书中涉及的数据结构有栈、队列、链表、树、并查集、堆和图等; 涉及的算法有排序、枚举、深度和广度优先搜索、图的遍历, 当然还有图论中不可以缺少的四种最短路径算法、两种最小生成树算法、割点与割边算法、二分图的最大匹配算法等。

-
- ◆ 著 啊哈磊
 - 策划编辑 陈 冰
 - 责任编辑 傅志红
 - 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>
三河市海波印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 16
字数: 300千字 2014年6月第1版
印数: 1—5 000册 2014年6月河北第1次印刷
-

定价: 45.00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: 京崇工商广字第 0021 号

编辑的话

作为本书的策划编辑，我很荣幸。

《啊哈！算法》是我读过的有趣且是我唯一能看懂的一本算法书。

我最初是因为啊哈磊写的另外一本书《啊哈！C》而认识啊哈磊的。啊哈磊还有个网站，也叫啊哈磊，这个啊哈磊网站中有一个论坛，叫啊哈论坛。论坛建立短短一年半时间，就聚集了15000多个啊哈小伙伴，都是萌物。我对他的写作风格很欣赏，那是一种因热爱和探究而产生的纯粹的快乐，因此，当啊哈磊率领着他的一大波萌物开开心心地攻城略地，浩浩荡荡地兵临城下，跟我说他想写一本通俗易懂的算法书，不知是否能出版时，我的回答是：“必须出版！”

这本书出版意向的达成就是这么简单。

但创作的过程一点也不轻松。因为任何一本拿得出手的书的创作都是作者大量时间和精力付出的结果，是毅力的累积。

几个月之后，我拿到了这本书的初稿。我高高兴兴地开始读。这部分写得通俗易懂，我看得津津有味。但读了一些之后，我发现我高兴不起来了，我遇到了困难，有些篇章写得太简略了，只是把算法的基本思路说了一下，然后就直接给出了以该算法实现的某个示例的完整代码。

这样不行，看不懂啊。原理很简单，但实现起来时，看代码就感觉对应不起来了。或许比我聪明的人能看懂，但我希望像我这种在算法方面毫无造诣的普通选手读起来也不吃力，于是我让啊哈磊完善它。我是这么交代的——你得写得让我能看懂才行。这要求非常地简单，但也非常地暗黑。

经过比我想得要长的时间，啊哈磊给了我第二版。

我继续阅读，很多之前看不懂的地方现在能看懂了，或者至少我认为我看懂了（请允许我使用这种让人生气的措辞），但还有少部分欠点劲儿。啊哈磊向我投来困惑又略带鄙视的目光，我用坚定又痴痴呆呆的目光把他的目光给顶了回去。

于是啊哈磊继续埋头苦干。

终于，我完全可以看懂的版本诞生了。

对于一本技术书，一个编辑可能犯下的最有价值的“错误”就是试图去完全读懂它。

在最后，我还要特别强调一点，这本书不仅写得通俗易懂，而且还在一个非常重要的方面超越了其他技术书，那就是这本书中还配了可爱的漫画，萌萌的画风，生动的场景，与文字浑然一体。

本书策划编辑：陈冰

序

我想写一本通俗易懂的算法书很久了，因为对于多数人而言，“算法”给他的第一印象就是很难懂，其实我也是这样。还记得我第一次学习图论的“割点割边”算法时，看过不下于四五本书，其中不乏一些算法经典书籍，还百度了一堆材料，才勉强将其看懂并实现成代码。其实这个算法并不难，核心代码不超过 20 行，但是很多算法书都是草草叙述，不同的书籍给出的参考代码也是五花八门，有的甚至都不稀罕给你代码，这大大增加了学习的难度。我是花了整整一个晚上才搞定的，当然这其中不排除智商因素。第二印象就是算法是枯燥无趣的，并且好像没什么作用。其实在我们的日常生活之中到处都可见到算法的影子，只不过它通常隐匿在事物的背后，不太容易被发现。但是它每天都在默默地为我们服务着。在本书中我将带你一步步揭开算法的奥秘，带它走近你的身边。

由于算法的内容确实是太多了，要想全部写清楚恐怕几本书都不够，本书将介绍一些最常用的算法。此外算法的实现通常需要依附一些数据结构，因此在必要的时候对于需要用到的数据结构我也会进行讲解。本书中涉及的数据结构有栈、队列、树、并查集、堆和图等；算法有各种排序、枚举、深度和广度优先搜索、图上的遍历，当然还有图论中不可以缺少的四种最短路径算法、两种最小生成树算法、割点与割边算法、二分图的最大匹配算法等。

尽管我不敢保证我写的算法你一定可以看懂（但凭着一股强大的自信，我认为初中以上文化程度的应该没问题^_^），但我会以一个故事或者一个你在生活中可能遇到的问题开始对一个算法进行讲解，并尽量用通俗易懂的语言配合有趣的插图让你在阅读本书的时候更像是在品读一篇篇轻松的短篇小说或是在玩一把趣味解谜游戏，在轻松愉悦中掌握算法精髓，感受算法之美。

致 谢

本书能得以面世，首先要感谢图灵的陈冰先生。感谢你主动联系我，给予我信心去完成本书的全部，并且提出了很多宝贵的建议。更加令我吃惊的是你竟然能读懂本书的全部算法（包括每一行代码），还发现了很多隐藏得很深的错误，真是一位非常棒的图书出版人。

在书稿创作的过程中，有幸和很多优秀的学生共同学习和探讨，是他们为本书的创作提供了灵感，感谢他们的倾听、交流和建议。他们是武汉二中的吕凯风同学、武汉外国语学校李嘉浩、熊子健、陈雨禾、郭明达和李丁等同学。

本书之所以变得这么有趣，还必须要感谢我的美女插画师郑佳茜，你灵感涌现的插图功不可没。

感谢我的好友张知严，无私地帮助我搭建了“添柴”编程在线学习系统（tianchai.org），为本书读者提供了更好的学习交流的平台。

感谢我的学生胡梦清，感谢你排除万难来参加你人生中的最后一场 NOIP 竞赛。是你用行动、青春路上追求梦想的精神，告诉我们 18 岁就应该可爱、执着、不畏惧，敢于朝着梦想前行。

特别感谢我的未婚妻 Snowin，是你放弃了近一年来所有的周末和节假日，陪我在书桌旁、咖啡厅里、旅途中……共同完成了本书的每一个字、每一幅图、每一段代码。

最后要感谢我的父母，你们把我拉扯大太不容易了，我爱你们！

啊哈磊

2014年5月6日

欢迎加入

图灵社区 ituring.com.cn

——最前沿的IT类电子书发售平台

电子出版的时代已经来临。在许多出版界同行还在犹豫彷徨的时候，图灵社区已经采取实际行动拥抱这个出版业巨变。作为国内第一家发售电子图书的IT类出版商，图灵社区目前为读者提供两种DRM-free的阅读体验：在线阅读和PDF。

相比纸质书，电子书具有许多明显的优势。它不仅发布快，更新容易，而且尽可能采用了彩色图片（即使有的书纸质版是黑白印刷的）。读者还可以方便地进行搜索、剪贴、复制和打印。

图灵社区进一步把传统出版流程与电子书出版业务紧密结合，目前已实现作译者网上交稿、编辑网上审稿、按章发布的电子出版模式。这种新的出版模式，我们称之为“敏捷出版”，它可以让读者以较快的速度了解到国外最新技术图书的内容，弥补以往翻译版技术书“出版即过时”的缺憾。同时，敏捷出版使得作、译、编、读的交流更为方便，可以提前消灭书稿中的错误，最大程度地保证图书出版的质量。

优惠提示：现在购买电子书，读者将获赠书款20%的社区银子，可用于兑换纸质样书。

——最方便的开放出版平台

图灵社区向读者开放在线写作功能，协助你实现自出版和开源出版的梦想。利用“合集”功能，你就能联合二三好友共同创作一部技术参考书，以免费或收费的形式提供给读者。（收费形式须经过图灵社区立项评审。）这极大地降低了出版的门槛。只要有写作的意愿，图灵社区就能帮助你实现这个梦想。成熟的书稿，有机会入选出版计划，同时出版纸质书。

图灵社区引进出版的外文图书，都将在立项后马上在社区公布。如果你有意翻译哪本图书，欢迎你来社区申请。只要你通过试译的考验，即可签约成为图灵的译者。当然，要想成功地完成一本书的翻译工作，是需要有坚强的毅力的。

——最直接的读者交流平台

在图灵社区，你可以十分方便地写文章、提交勘误、发表评论，以各种方式与作译者、编辑人员和其他读者进行交流互动。提交勘误还能够获赠社区银子。

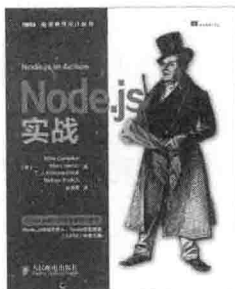
你可以积极参与社区经常开展的访谈、乐译、评选等多种活动，赢取积分和银子，积累个人声望。



2010年5月,美股“闪电崩盘”,万亿美元市值蒸发。2011年4月,亚马逊书店一本研究遗传学、晦涩难懂的旧书《苍蝇的成长》(*The Making of a Fly*),售价居然在几天内飙升至2400万美元。谜底揭开后,人们发现……

《算法帝国》是《纽约时报》畅销书作者 Christopher Steiner 的又一力作,通过一个又一个引人入胜的故事,向读者介绍了算法掌控世界的真实情况,揭示了“机器人革命”是如何悄然在我们身边发生的。

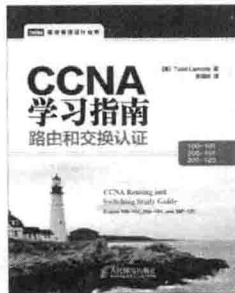
算法帝国
书号: 978-7-115-34900-2
作者: Christopher Steiner
定价: 49.00元



Node.js 实战
书号: 978-7-115-35246-0
作者: Mike Cantelon, Marc Harter, T.J. Holowaychuk, Nathan Rajlich
定价: 69.00元



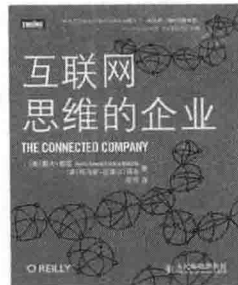
Python 开发实战
书号: 978-7-115-32089-6
作者: BePROUD 股份有限公司
定价: 79.00元



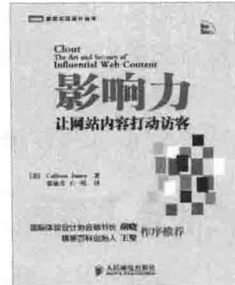
CCNA 学习指南: 路由和交换认证 (100-101, 200-101, 200-120)
书号: 978-7-115-35302-3
作者: Todd Lammle
定价: 129.00元



写给大家看的设计书 (第3版·精装版)
书号: 978-7-115-34338-3
作者: Robin Williams
定价: 69.00元



互联网思维的企业
书号: 978-7-115-34930-9
作者: 戴夫·格雷 托马斯·范德尔·沃尔
定价: 59.00元



影响力: 让网站内容打动访客
书号: 978-7-115-35242-2
作者: Colleen Jones
定价: 49.00元

目 录

第 1 章 一大波数正在靠近——排序	1
第 1 节 最快最简单的排序——桶排序	2
第 2 节 邻居好说话——冒泡排序	7
第 3 节 最常用的排序——快速排序	12
第 4 节 小哼买书	20
第 2 章 栈、队列、链表	25
第 1 节 解密 QQ 号——队列	26
第 2 节 解密回文——栈	32
第 3 节 纸牌游戏——小猫钓鱼	35
第 4 节 链表	44
第 5 节 模拟链表	54
第 3 章 枚举！很暴力	57
第 1 节 坑爹的奥数	58
第 2 节 炸弹人	61
第 3 节 火柴棍等式	67
第 4 节 数的全排列	70
第 4 章 万能的搜索	72
第 1 节 不撞南墙不回头——深度优先搜索	73
第 2 节 解救小哈	81
第 3 节 层层递进——广度优先搜索	88
第 4 节 再解炸弹人	95
第 5 节 宝岛探险	106
第 6 节 水管工游戏	117
第 5 章 图的遍历	128
第 1 节 深度和广度优先究竟是指啥	129
第 2 节 城市地图——图的深度优先遍历	136

第 3 节 最少转机——图的广度优先遍历	142
第 6 章 最短路径	147
第 1 节 只有五行的算法——Floyd-Warshall	148
第 2 节 Dijkstra 算法——通过边实现松弛	155
第 3 节 Bellman-Ford——解决负权边	163
第 4 节 Bellman-Ford 的队列优化	171
第 5 节 最短路径算法对比分析	177
第 7 章 神奇的树	178
第 1 节 开启“树”之旅	179
第 2 节 二叉树	183
第 3 节 堆——神奇的优先队列	185
第 4 节 擒贼先擒王——并查集	200
第 8 章 更多精彩算法	211
第 1 节 镖局运镖——图的最小生成树	212
第 2 节 再谈最小生成树	219
第 3 节 重要城市——图的割点	229
第 4 节 关键道路——图的割边	234
第 5 节 我要做月老——二分图最大匹配	237
第 9 章 还能更好吗——微软亚洲研究院面试	243

第1章

一大波数正在靠近
排序



第 1 节 最快最简单的排序——桶排序

在我们生活的这个世界中到处都是被排序过的东东。站队的时候会按照身高排序，考试的名次需要按照分数排序，网上购物的时候会按照价格排序，电子邮箱中的邮件按照时间排序……总之很多东东都需要排序，可以说排序是无处不在。现在我们举个具体的例子来介绍一下排序算法。



首先出场的是我们的主人公小哼，上面这个可爱的娃就是啦。期末考试完了老师要将同学们的分数按照从高到低排序。小哼的班上只有 5 个同学，这 5 个同学分别考了 5 分、3 分、5 分、2 分和 8 分，哎，考得真是惨不忍睹（满分是 10 分）。接下来将分数进行从大到小排序，排序后是 8 5 5 3 2。你有没有什么好方法编写一段程序，让计算机随机读入 5 个数然后将这 5 个数从大到小输出？请先想一想，至少想 15 分钟再往下看吧（*^__^*）。



我们这里只需借助一个一维数组就可以解决这个问题。请确定你真的仔细想过再往下看哦。

首先我们需要申请一个大小为 11 的数组 `int a[11]`。OK，现在你已经有了 11 个变量，编号从 `a[0]`~`a[10]`。刚开始的时候，我们将 `a[0]`~`a[10]` 都初始化为 0，表示这些分数还都没有人得过。例如 `a[0]` 等于 0 就表示目前还没有人得过 0 分，同理 `a[1]` 等于 0 就表示目前还没有人得过 1 分……`a[10]` 等于 0 就表示目前还没有人得过 10 分。

0	0	0	0	0	0	0	0	0	0	0	0
<code>a[0]</code>	<code>a[1]</code>	<code>a[2]</code>	<code>a[3]</code>	<code>a[4]</code>	<code>a[5]</code>	<code>a[6]</code>	<code>a[7]</code>	<code>a[8]</code>	<code>a[9]</code>	<code>a[10]</code>	

数组下标 0~10 分别表示分数 0~10
不同的分数所对应的单元格则存储着得此分数的人数

下面开始处理每一个人的分数，第一个人的分数是 5 分，我们就将相对应的 `a[5]` 的值在原来的基础上增加 1，即将 `a[5]` 的值从 0 改为 1，表示 5 分出现过了一次。

0	0	0	0	0	1	0	0	0	0	0
<code>a[0]</code>	<code>a[1]</code>	<code>a[2]</code>	<code>a[3]</code>	<code>a[4]</code>	<code>a[5]</code>	<code>a[6]</code>	<code>a[7]</code>	<code>a[8]</code>	<code>a[9]</code>	<code>a[10]</code>

第二个人的分数是 3 分，我们就把相对应的 `a[3]` 的值在原来的基础上增加 1，即将 `a[3]` 的值从 0 改为 1，表示 3 分出现过了一次。

0	0	0	1	0	1	0	0	0	0	0
<code>a[0]</code>	<code>a[1]</code>	<code>a[2]</code>	<code>a[3]</code>	<code>a[4]</code>	<code>a[5]</code>	<code>a[6]</code>	<code>a[7]</code>	<code>a[8]</code>	<code>a[9]</code>	<code>a[10]</code>

注意啦！第三个人的分数也是 5 分，所以 `a[5]` 的值需要在此基础上再增加 1，即将 `a[5]` 的值从 1 改为 2，表示 5 分出现过了两次。

0	0	0	1	0	2	0	0	0	0	0
<code>a[0]</code>	<code>a[1]</code>	<code>a[2]</code>	<code>a[3]</code>	<code>a[4]</code>	<code>a[5]</code>	<code>a[6]</code>	<code>a[7]</code>	<code>a[8]</code>	<code>a[9]</code>	<code>a[10]</code>

按照刚才的方法处理第四个和第五个人的分数。最终结果就是下面这个图啦。

0	0	1	1	0	2	0	0	1	0	0
<code>a[0]</code>	<code>a[1]</code>	<code>a[2]</code>	<code>a[3]</code>	<code>a[4]</code>	<code>a[5]</code>	<code>a[6]</code>	<code>a[7]</code>	<code>a[8]</code>	<code>a[9]</code>	<code>a[10]</code>

你发现没有，`a[0]~a[10]` 中的数值其实就是 0 分到 10 分每个分数出现的次数。接下来，我们只需要将出现过的分数打印出来就可以了，出现几次就打印几次，具体如下。

`a[0]` 为 0，表示“0”没有出现，不打印。

`a[1]` 为 0，表示“1”没有出现，不打印。

`a[2]` 为 1，表示“2”出现过 1 次，打印 2。

`a[3]` 为 1，表示“3”出现过 1 次，打印 3。

`a[4]` 为 0，表示“4”没有出现，不打印。

`a[5]` 为 2，表示“5”出现过 2 次，打印 5 5。

`a[6]` 为 0，表示“6”没有出现，不打印。

`a[7]` 为 0，表示“7”没有出现，不打印。

`a[8]` 为 1，表示“8”出现过 1 次，打印 8。

`a[9]` 为 0，表示“9”没有出现，不打印。

`a[10]` 为 0，表示“10”没有出现，不打印。

最终屏幕输出“23558”，完整的代码如下。

```
#include <stdio.h>
int main()
{
    int a[11],i,j,t;
    for(i=0;i<=10;i++)
        a[i]=0; //初始化为0

    for(i=1;i<=5;i++) //循环读入5个数
    {
```



```

scanf("%d",&t); //把每一个数读到变量t中
a[t]++; //进行计数
}

for(i=0;i<=10;i++) //依次判断a[0]~a[10]
    for(j=1;j<=a[i];j++) //出现了几次就打印几次
        printf("%d ",i);

getchar();getchar();
//这里的getchar();用来暂停程序,以便查看程序输出的内容
//也可以用system("pause");等来代替
return 0;
}

```

输入数据为:

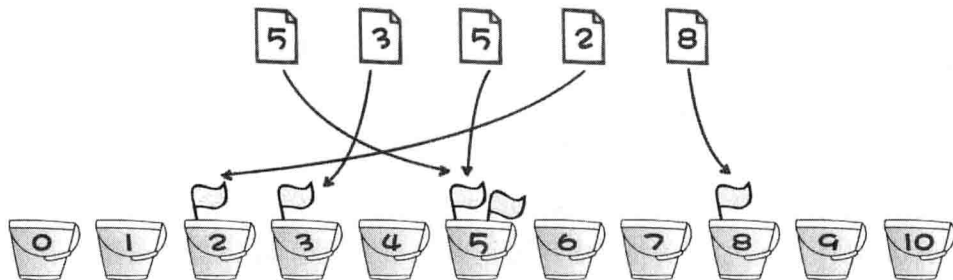
```
5 3 5 2 8
```

仔细观察的同学会发现,刚才实现的是从小到大排序。但是我们要求是从大到小排序,这该怎么办呢?还是先自己想一想再往下看哦。

其实很简单。只需要将 `for(i=0;i<=10;i++)` 改为 `for(i=10;i>=0;i--)` 就 OK 啦,快去试一试吧。

这种排序方法我们暂且叫它“桶排序”。因为其实真正的桶排序要比这个复杂一些,以后再详细讨论,目前此算法已经能够满足我们的需求了。

这个算法就好比有 11 个桶,编号从 0~10。每出现一个数,就在对应编号的桶中放一个小旗子,最后只要数数每个桶中有几个小旗子就 OK 了。例如 2 号桶中有 1 个小旗子,表示 2 出现了一次;3 号桶中有 1 个小旗子,表示 3 出现了一次;5 号桶中有 2 个小旗子,表示 5 出现了两次;8 号桶中有 1 个小旗子,表示 8 出现了一次。



现在你可以尝试一下输入 n 个 0~1000 之间的整数,将它们从大到小排序。提醒一下,

如果需要对数据范围在 0~1000 的整数进行排序，我们需要 1001 个桶，来表示 0~1000 之间每一个数出现的次数，这一点一定要注意。另外，此处的每一个桶的作用其实就是“标记”每个数出现的次数，因此我喜欢将之前的数组 `a` 换个更贴切的名字 `book`（`book` 这个单词有记录、标记的意思），代码实现如下。

```
#include <stdio.h>

int main()
{
    int book[1001],i,j,t,n;
    for(i=0;i<=1000;i++)
        book[i]=0;
    scanf("%d",&n); //输入一个数n，表示接下来有n个数
    for(i=1;i<=n;i++) //循环读入n个数，并进行桶排序
    {
        scanf("%d",&t); //把每一个数读到变量t中
        book[t]++; //进行计数，对编号为t的桶放一个小旗子
    }
    for(i=1000;i>=0;i--) //依次判断编号1000~0的桶
        for(j=1;j<=book[i];j++) //出现了几次就将桶的编号打印几次
            printf("%d ",i);

    getchar();getchar();
    return 0;
}
```

可以输入以下数据进行验证。

```
10
8 100 50 22 15 6 1 1000 999 0
```

运行结果是：

```
1000 999 100 50 22 15 8 6 1 0
```

最后来说下时间复杂度的问题。代码中第 6 行的循环一共循环了 m 次（ m 为桶的个数），第 9 行的代码循环了 n 次（ n 为待排序数的个数），第 14 行和第 15 行一共循环了 $m+n$ 次。所以整个排序算法一共执行了 $m+n+m+n$ 次。我们用大写字母 O 来表示时间复杂度，因此该