

算法技术手册

作者：海涅曼(George T.Heineman)

算法技术手册

Algorithms in a Nutshell

[美]海涅曼 (Heineman, G. T.)

[美]波利切 (Pollice, G.)

[美]塞克欧 (Selkow, S.) 著

杨晨 等译

ISBN: 978-7-111-28674-5

本书纸版由机械工业出版社于2009年出版，电子版由华章分社（北京华章图文信息有限公司）全球范围内制作与发行。

版权所有，侵权必究

客服热线：+ 86-10-68995265

客服信箱：service@bbbvip.com

官方网址：www.hzmedia.com.cn

新浪微博 @研发书局

腾讯微博 @yanfabook

目 录

[O'Reilly Media, Inc. 介绍](#)

[译者序](#)

[前言](#)

[原则：使用实际代码，而不是伪代码](#)

[原则：将算法和将要解决的问题分开](#)

[原则：仅仅讲述足够的数学](#)

[原则：用经验来支持数学分析](#)

[目标读者](#)

[本书组织方式](#)

[本书体例](#)

[代码使用说明](#)

[联系我们](#)

[致谢](#)

[参考文献](#)

[第一部分](#)

[第1章 算法真的很重要](#)

[理解问题](#)

[如果需要，尽可能用实践检验](#)

[解决问题的算法](#)

[花絮](#)

[故事的寓意](#)

[参考文献](#)

[第2章 算法的数学原理](#)

[问题样本的规模](#)

[函数的增长率](#)

[最好最坏和平均情况下的性能分析](#)

[最坏情况](#)

[平均情况](#)

[最好情况](#)

[性能指标](#)

[讨论0：常数级算法的性能](#)

[讨论1：对数级算法的性能](#)

[讨论2：次线性的算法的性能，时间复杂度为 \$O\(n\)\$](#)

[讨论3：线性算法的性能](#)

[讨论4： \$n \log n\$ 算法的性能](#)

[讨论5a：二次方的算法性能](#)

[讨论5b：性能不明显的计算](#)

[混合操作](#)

[基准测试](#)

[最后一点](#)

[参考文献](#)

[第3章 模式和领域](#)

[模式：一种交流语言](#)

[算法模式的格式](#)

[算法模式的格式](#)

[伪代码模式的格式](#)

[设计格式](#)

[基于经验的评价格式](#)

[领域和算法](#)

[浮点计算](#)

[舍入的错误](#)

[值之间的比较](#)

[特殊的量](#)

[性能](#)

[手动内存分配](#)

[选择一门编程语言](#)

[参考文献](#)

[第二部分](#)

[第4章 排序算法](#)

[概述](#)
[术语](#)
[表述](#)
[可比较的元素](#)
[稳定排序](#)
[分析技术](#)
[通用输入](#)
[插入排序](#)
[使用环境](#)
[驱动因素](#)
[解决方案](#)
[结论](#)
[分析](#)
[中值排序](#)
[使用环境](#)
[驱动因素](#)
[解决方案](#)
[结论](#)
[分析](#)
[快速排序](#)
[使用环境](#)
[解决方案](#)
[结论](#)
[分析](#)
[变种](#)
[选择排序](#)
[堆排序](#)
[使用环境](#)
[驱动因素](#)
[解决方案](#)
[分析](#)
[变种](#)
[计数排序](#)
[使用环境](#)
[驱动因素](#)
[解决方案](#)
[分析](#)
[桶排序](#)
[使用环境](#)
[驱动因素](#)
[解决方案](#)
[分析](#)
[变种](#)
[选择排序算法的标准](#)
[综合分析基准测试结果](#)
[双浮点数的基准测试结果](#)
[参考文献](#)
[第5章 查找](#)
[概述](#)
[顺序查找](#)
[输入/输出](#)
[使用环境](#)
[驱动因素](#)
[解决方案](#)
[结论](#)
[分析](#)
[变种](#)
[二分查找](#)
[输入/输出](#)
[使用环境](#)

[驱动因素](#)
[解决方案](#)
[结论](#)
[分析](#)
[变种](#)
[基于散列的查找](#)

[输入/输出](#)
[使用环境](#)
[驱动因素](#)
[解决方案](#)
[结果](#)
[分析](#)
[变种](#)

[二叉查找树](#)
[输入/输出](#)
[使用环境](#)
[驱动因素](#)
[解决方案](#)
[结论](#)
[分析](#)
[变种](#)

[参考文献](#)
[第6章 图算法](#)

[概述](#)
[图](#)
[存储问题](#)
[图分析](#)
[数据结构设计](#)
[问题](#)

[深度优先搜索](#)
[输入/输出](#)
[使用环境](#)
[解决方案](#)
[分析](#)

[广度优先搜索](#)
[输入/输出](#)
[使用环境](#)
[解决方案](#)
[分析](#)

[单源最短路径](#)
[输入/输出](#)
[解决方案](#)

[结论](#)
[分析](#)
[变种](#)
[比较](#)

[所有点对最短路径](#)

[输入/输出](#)
[解决方案](#)
[分析](#)

[最小生成树算法](#)
[解决方案](#)

[结论](#)
[分析](#)
[变种](#)

[参考文献](#)
[第7章 人工智能中的寻路](#)

[概述](#)
[博弈树](#)
[搜索树](#)

[关键思想](#)

[假设](#)

[深度优先搜索](#)

[输入/输出](#)

[使用环境](#)

[解决方案](#)

[结论](#)

[分析](#)

[广度优先搜索](#)

[输入/输出](#)

[使用环境](#)

[解决方案](#)

[结论](#)

[分析](#)

[A*搜索](#)

[输入/输出](#)

[使用环境](#)

[解决方案](#)

[结论](#)

[驱动因素](#)

[分析](#)

[变种](#)

[相关算法](#)

[比较](#)

[Minimax](#)

[输入/输出](#)

[使用环境](#)

[解决方案](#)

[结论](#)

[分析](#)

[变种](#)

[NegMax](#)

[输入/输出](#)

[使用环境](#)

[解决方案](#)

[结论](#)

[分析](#)

[AlphaBeta](#)

[输入/输出](#)

[解决方案](#)

[结论](#)

[分析](#)

[参考文献](#)

[第8章 网络流算法](#)

[概述](#)

[网络流](#)

[最大流](#)

[输入/输出](#)

[解决方案](#)

[结论](#)

[分析](#)

[优化](#)

[相关算法](#)

[二部图匹配](#)

[输入/输出](#)

[解决方案](#)

[分析](#)

[在增广路上的深入思考](#)

[最小开销流](#)

[转运问题](#)

[解决方案](#)
[运输问题](#)
[解决方案](#)
[任务分配问题](#)
[解决方案](#)
[线性编程](#)
[参考文献](#)
[第9章 计算几何](#)
[概述](#)
[经典问题](#)
[假设](#)
[计算几何经典问题](#)
[凸包扫描](#)
[输入/输出](#)
[使用环境](#)
[驱动因素](#)
[解决方案](#)
[结论](#)
[分析](#)
[变种](#)
[相关算法](#)
[线段扫描](#)
[输入/输出](#)
[使用环境](#)
[驱动因素](#)
[解决方案](#)
[结论](#)
[分析](#)
[变种](#)
[最近点查询](#)
[输入/输出](#)
[使用环境](#)
[驱动因素](#)
[解决方案](#)
[结论](#)
[分析](#)
[变种](#)
[范围查询](#)
[输入/输出](#)
[使用环境](#)
[驱动因素](#)
[解决方案](#)
[分析](#)
[参考文献](#)
[第三部分](#)
[第10章 最后的招数](#)
[另类算法](#)
[近似算法](#)
[离线算法](#)
[并行算法](#)
[随机算法](#)
[估算集合的大小](#)
[估算搜索树的大小](#)
[结果可能出错但可以衰减错误率的算法](#)
[检测数据库间的不一致](#)
[零知识证明](#)
[参考文献](#)
[第11章 尾声](#)
[概述](#)
[原则：了解数据](#)

[原则：将问题分解至更小的问题](#)

[原则：选择正确的数据结构](#)

[原则：空间换时间](#)

[原则：如果没有显而易见的解法，使用搜索](#)

[原则：如果没有显而易见的解法，将问题归约为另一个有解的问题](#)

[原则：编写算法难，测试算法更难](#)

[第四部分](#)

[附录 基准测试](#)

[统计基础](#)

[硬件](#)

[例子](#)

[Java基准测试解决方案](#)

[Linux基准测试解决方案](#)

[Scheme基准测试解决方案](#)

[报告](#)

[精度](#)

[作者简介](#)

[封面介绍](#)

O'Reilly Media,Inc.介绍

为了满足读者对网络和软件技术知识的迫切需求，世界著名计算机图书出版机构O'Reilly Media,Inc.授权机械工业出版社，翻译出版一批该公司久负盛名的英文经典技术专著。

O'Reilly Media,Inc.是世界上在UNIX、X、Internet和其他开放系统图书领域具有领导地位的出版公司，同时也是联机出版的先锋。

从最畅销的The Whole Internet User's Guide & Catalog（被纽约公共图书馆评为20世纪最重要的50本书之一）到GNN（最早的Internet门户和商业网站），再到WebSite（第一个桌面PC的Web服务器软件），O'Reilly Media,Inc.一直处于Internet发展的最前沿。

许多书店的反馈表明，O'Reilly Media,Inc.是最稳定的计算机图书出版商——每一本书都一版再版。与大多数计算机图书出版商相比，O'Reilly Media,Inc.具有深厚的计算机专业背景，这使得O'Reilly Media,Inc.形成了一个非常不同于其他出版商的出版方针。O'Reilly Media,Inc.所有的编辑人员以前都是程序员，或者是顶尖级的技术专家。O'Reilly Media,Inc.还有许多固定的作者群体——他们本身是相关领域的技术专家、咨询专家，而现在编写著作，O'Reilly Media,Inc.依靠他们及时地推出图书。因为O'Reilly Media,Inc.紧密地与计算机业界联系着，所以O'Reilly Media,Inc.知道市场上真正需要什么图书。

译者序

算法，神秘而晦涩的词汇。算法，是计算机科学中最重要同时也是最基础的一环。从开始学习计算机，我们就深知，算法是整个计算机科学的核心。然而直至我们工作数年后，能够真正学好算法的人，却依旧是凤毛麟角。这并不是计算机教育的错，也不是计算机从业人员的错，更不是算法的错。长久以来，算法就像古老的咒语，算法背后高深的数学知识更让人望而生畏。其实，我们始终没有找到一条从理论走向实践的路。

在这里，我们很高兴能向大家介绍本书。它正是能够带领你学好算法的一本不可多得的好书。

本书的三位作者是伍斯特理工学院的教授，其中George T.Heineman毕业于达特茅斯学院和哥伦比亚大学，曾经获得过GE、IBM和AT&T的研究奖金，在软件工程方面有独到的研究。而Gary Pollice曾经供职于Rational Software、Sun等多家巨头，有着丰富的工业界经验，知道如何将学术和工业结合起来。Stanley M.Selkow毕业于卡内基梅隆大学和宾夕法尼亚大学，擅长图论和算法设计。本书由这三位伍斯特理工学院计算机理论专家合著，向我们展示了工业界和学术界对算法的不同看法以及如何高效地将理论和实践相结合。本书搭建了一条真正属于开发者的路。

本书的读者主要面向本科生以及程序设计人员，同样也适用于产品和项目管理人员。由于译者的知识和经验有限，翻译中难免有疏漏或错误，敬请广大读者谅解并批评指正。

杨晨 李明

2009年7月

前言

就像《黑客帝国》里面的Trinity所说的：

“Neo，是这个问题驱使着我们，是这个问题带你来到这儿。”

你知道这个问题，我也是。

作为本书的作者，我们将回答引领你到此的问题：

我能够使用某个算法解决我的问题吗？如果可以，那么怎么实现呢？

你也许并不需要理解一个算法为什么是正确的。如果你需要，那么请看看其他的资料，例如1180页的算法圣经——《算法导论》，作者是Thomas H.Cormen等（2001）。在那本书中你会了解到推论、定理以及证明；你也会从一些练习题和逐步递进的样例中看到算法是如何执行的。也许你会惊奇地发现，在算法导论中你找不到任何的代码，仅仅是一些伪代码的片段，伪代码是无数的算法教科书用来阐述算法的高级描述手段。在课堂上，这些教科书是非常重要的，但是在实际软件开发中，它们却起不到应有的作用，因为这些书假定伪代码都能够直接变成实际代码。

我们希望经验丰富的程序员在寻找问题的解决方案时，能够频繁参考本书。作为一名程序员，你每天要解决的问题都能在这里找到解决方案。在软件中，算法是决定成败的关键因素，在这里你能够了解到哪些决定能够改善关键算法的性能，也能够找到适合你的需求和解决方案的实际代码。

所有的算法都有实现，并且都使用测试工具经过仔细测试，以确保其正确性。而且，它们有足够的代码文档，能在这本书的代码库附录中找到它们。我们严格地遵照一系列的原则来设计算法、实现算法，以及编写这本书。如果这些原则对你很有意义，那么这本书也会同样有用。

原则：使用实际代码，而不是伪代码

为了计算最大网络流，一个实践者应该做些什么才能将图P-1的Ford-Fulkerson算法描述转换成实际代码呢？

□

图 P-1 教科书中常见的伪代码

图中的算法描述来自于维基百科（http://en.wikipedia.org/wiki/Ford_Fulkerson），这个描述与《算法导论》上的伪代码极其相似。最好还是不要期望一个软件的开发能够根据这个Ford-Fulkerson算法的描述开发出实际的代码。翻到第8章，对比一下我们的代码。我们只使用有注释的，并且是精心设计过的代码。在你自己写的代码或者软件系统中使用我们提供的现成代码，或者这些代码的逻辑吧。

一些算法教科书确实有完整的C或者Java代码。但是这些教科书的目的是教初学者编程语言，或者是解释如何实现抽象数据类型。而且代码都只是在页面的狭窄边栏，作者通常都会忽略注释和错误处理，或者使用在实际应用中不会用到的快捷方法。我们相信程序员能够从有注释的，并且是精心设计过的代码中学到更多的东西，这就是我们为什么做如此多的工作来开发算法的实际解决方案。

欢迎访问：电子书学习和下载网站 (<https://www.shgis.cn>)

文档名称：《算法技术手册》海涅曼(George T.Heineman) 著.epub

请登录 <https://shgis.cn/post/322.html> 下载完整文档。

手机端请扫码查看：

