<<程序员实用算法>>

图书基本信息

书名：<<程序员实用算法>>

13位ISBN编号：9787111272960

10位ISBN编号：711127296X

出版时间：2009-9

出版时间：机械工业出版社

作者：Andrew Binstock,John Rex

页数：437

译者：陈宗斌

<<程序员实用算法>>

前言

数据结构与算法是计算机专业的核心课程，是计算机软件开发和应用人员必备的专业基础。
今天的大多数关于算法的图书都是大学教科书，或者是令人厌倦的相同算法集合改头换面后的作品。
本书是给出所有算法的完整代码实现的第一本书，这些算法对于开发人员在其日常工作中是有用的。

本书介绍了关于算法的基础知识、基本数据结构、散列、查找、排序、树、日期和时间、任意精度的算术运算、数据压缩以及数据完整性和验证等内容。
本书的目的是为在应用程序中使用的算法提供一个实用的纲要。
与关于算法的大多数著作不同的是，本书不是一本教材：书中没有提供实现细节，把它作为练习留给读者完成；也没有利用较小的代码段对算法进行高度理论化的讨论，以说明如何进行实现。
相反，本书完全用c语言实现了各种算法，并且讨论了如何在各种应用程序中最佳地使用它们。

本书只要求读者具有c语言的初级知识以及不超出基本代数之外的数学知识。
源代码是符合ANSI标准的，并且对它们进行了测试，它们都可以运行在UNIX下以及Borland、Microsoft和Watcom的编译器上。

本书非常适合于高等院校计算机专业的学生阅读，对于从事计算机软件开发的人员，也将从本书中受益匪浅。

参加本书翻译的人员有：陈宗斌、张景友、易小丽、陈婷、管学岗、王新彦、金惠敏、张海峰、徐晔、戴锋。

由于时间紧迫，加之译者水平有限，错误在所难免，恳请广大读者批评指正。

<<程序员实用算法>>

内容概要

《程序员实用算法》重点关注的是实用、立即可用的代码，并且广泛讨论了可移植性和特定于实现的细节。

《程序员实用算法》作者介绍了一些有用但很少被讨论的算法，它们可用于语音查找、日期和时间例程（直到公元1年）、B树和索引文件、数据压缩、任意精度的算术、校验和与数据验证，并且还最全面地介绍了查找例程、排序算法和数据结构。

《程序员实用算法》结构清晰，示例丰富，可作为广大程序员的参考用书。

作者简介

Andrew Binstock，是《UNIX Review》的主编和《C Gazette》的创刊编辑。他是《HP LaserJet Programming》（Addison-Wesley，1991）的第一作者。

John Rex，是一位计算机顾问，专攻C和C++。他是《C Gazette》的前任技术编辑，并且为许多杂志撰写文章。

<<程序员实用算法>>

书籍目录

<<程序员实用算法>>

章节摘录

第1章　绪论　　1.1　评估算法　　除了最直观的应用之外，算法是所有程序的核心和灵魂。
算法一般被设计用于以最小的代价高效地解决特定的问题。
算法的价值一般取决于两方面因素：如何恰当地解决问题以及如何高效地实现解决方案。
这些是算法分析的定性和定量方面。
对于许多算法，质量不是一个问题。
例如，对于排序算法，必须保证每次都对所有元素正确地进行了排序。
一旦出错，就必须丢弃它并且严格说来不能将其视为一种算法。
在其他领域，不能基于这种简单的通过／失败测试来度量质量。
例如，在第4章中介绍的Soundex算法允许检索听起来相同的单词或名字。
与排序算法不同，可以调整Soundex算法，以寻找接近的匹配或者相当宽泛的匹配；这取决于实现算法的方式和开发人员的需求。
在这种情况下，质量是可度量的并且是算法的重要方面，并且指导我们认真选择不同的解决方案。
算法设计的定量方面尝试确定算法所需的资源。
一般来说，最重要的度量标准是时间：即算法运行得有多快?偶尔，计算机资源（比如可用的内存）也是一个重要因素。
度量性能　　与基准的性能不同，算法的性能很少依据时间来加以说明。
在论及排序例程时，你几乎从未听到它完成排序要花费8．62秒这样的说明。
这有一个很好的理由：这种计时难以复制，并且通常依赖于正在处理的数据的具体特征。
算法不依赖于计时，而是依赖于一个直观的方程，以显示输入的大小与性能之间的关系。
用于显示这种关系的传统方法是使用符号D，称为大O表示法（big.oh notation）。
其工作方式是：假定你有一个算法，它简单地通读一个文本文件，从中查找单词flea。
一种合理的方法是寻找字母f的每个实例（参见第4章）。
当找到一个f，该算法就测试4个字母的序列，看看它是不是单词flea。
在这个示例中，显然执行时间直接与文本文件的大小成正比。
如果给定的文件包含IV个字符，那么我们就说该算法的执行时间的界限是O（N）。
你会注意到这种表述没有考虑到可能影响性能的其他因素——例如，字母f在文本中出现的频繁程度。
在查找字符串时（比如fleas rarely wear collars），字符串的长度以及其中相似字符串（比如fleasrarely weal"colors）出现的频率也会影响性能。
不过，严格来讲，这些因素是要处理的数据的函数，而不是算法的函数。
因此，在大O表示法中，它们不会出现在公式中。
该表示法只是简单地说明数据规模（一般用IV表示，偶尔也用n表示）与算法的典型性能之间的关系。
……

<<程序员实用算法>>

版权说明