

<<大话数据结构>>

图书基本信息

书名：<<大话数据结构>>

13位ISBN编号：9787302255659

10位ISBN编号：7302255652

出版时间：2011-6

出版时间：清华大学出版社

作者：程杰

页数：468

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;大话数据结构&gt;&gt;

## 前言

前言本书起因大家好！

我是《大话设计模式》（2008年初出版）的作者，三年来，承蒙广大读者的厚爱，《大话设计模式》取得了较大的成功。

仅在当当网，截止本文写作时，就已经有1073条评论，705次5星评价，位居五星图书榜计算机/网络类的累计总榜第二名。

此书已经成为国内原创计算机类图书最畅销的书籍之一。

对于这样一个自己喜欢做、可以做得好，而且已经得到了市场广泛认可，为很多朋友提供帮助的事情，我没有理由不去继续做下去。

这就是我准备再写书的原因。

我曾做过调查，数据结构的学习者大多都有这样的感慨：数据结构很重要，一定要学好，但数据结构比较抽象，有些算法理解起来很困难，学得很累。

可我更希望传达这样的信息：数据结构非常有趣，很多算法是智慧的结晶，学习它是去感受计算机编程技术的魅力，在理解掌握它的同时，整个过程都是一种愉悦的精神感受，而非枯燥乏味的一门课程。

因此我决定写作一本关于数据结构有趣的书。

不过现实总比理想来得更“现实”。

要想把书写好，谈何容易，我需要突破很多困难……嗜！

不管如何，现在您看到了本书，那就说明我已经克服了困难战胜了自己。

希望您喜欢上这本书。

本书定位本书的定位就是一本适合读者自学数据结构的书籍，它有区别于教材，希望给大家另一种阅读体验。

通常讲解数据结构的图书都是以教材的方式呈现。

在写作前，我购买或在图书馆借阅了十几本非常好的数据结构相关教材用来为写作本书做准备。

但经过认真阅读后，我发现，它们大多不是一本好的“自学读物”。

我没有轻视这些好书的意思，不过教材和自学读物，所面向的读者是完全不同的。

好的教材应该是提纲挈领、重点突出，一定要留出思考的空间，否则就没必要再听老师上课了。

很多内容的讲解是由老师在课堂完成，教材中有练习、课后习题、思考题等，这些大多可以通过老师来解答。

比如我们中学时的语文、数学课本，很薄的一本书通常要用一学期、甚至一年的时间来学习，这就是因为它们都是教材而不是自学读物。

如果是小说，可能一两天就读完了。

好的自学读物的目标是让初学者“独自”全盘掌握知识，需要强调“独自”一词，这就说明读者在阅读时，是完全依靠自己的力量来向未知发出挑战。

因此书中内容，要么不写，写了就应该写透。

如果读者在阅读时总是疑惑重重，那么这本书就有很大的问题了。

我也就是在基于这样的认识，决心将《大话数据结构》真正写成一本关于数据结构和算法的自学读物来展开写作的。

本书特色1. 趣味引导大部分的编程类图书，在内容上基本都是直奔主题。

但是尼采曾说过：“人们无法理解他没有经历过的事情。

”换句话说，我们只接受过去早已理解的事物相关的信息。

这是一种比较学习过程，在这个过程中，大脑寻找每条信息之间的联系。

所以教育专家普遍认为，吸引学生的注意力，比较好的办法是用他们比较熟知的知识开始。

因此在本书中，我会用一个故事、一个趣味题目、一部电影的介绍等形式来作为每一章甚至很多小节的开头，选择的内容也多多少少与要讲的主题内容相关。

这并不是多余，而是有意为之。

## &lt;&lt;大话数据结构&gt;&gt;

事实上，这样的形式在我的前一本书中已经得到了普遍认可。

2. 图文并茂西方有句谚语，“A picture is worth a thousand words.（一图值千言）”。

用上千个字描述不明白的东西，很可能一张图就能解释清楚。

我非常认可这个观点，所以本书虽没有达到每一页都有图，但基本做到了绝大部分讲解都有相关图示，关键算法更是通过多图逐步分解剖析。

尽管这带来了写作上的难度，但却可以达到较好的效果。

毕竟，读者通过本书开始学习数据结构时，要从一无所知或略知一二到完全理解，甚至掌握应用，是需要一个比较艰苦的过程，用大量的图示可以减少这个过程的长度。

3. 代码详解我在写作中尽量摒弃了传统数据结构教材的“重理论思想而轻代码讲解”的作法。

在准备数据结构写作时我发现，很多教材对数据结构理论和算法设计思想讲得比较好，可一到实际代码时，有的把代码贴出来加少量注释，有的直接用伪代码形式。

这对于上课的学生还好，毕竟有老师在课堂中去详解代码编写原理，可是对于初学数据结构和算法的自学者而言，如果书中不去解释代码某些细节为什么那样编写的原因，甚至代码根本不可能在某个编译器中运行通过，其挫折感是很强烈的。

比如即使理解了图结构中的最短路径求解原理，也可能无法写出最短路径的算法。

我把代码在运行过程中变量的变化融入到整个算法设计思想的讲解中，配合相应的示意图，会帮助大家更加容易理解算法的实质。

这种讲解模式在本书的第6、7、8、9章的很多复杂算法中有具体体现，越是复杂的代码越是讲解细致。

这算是本书的一个特色，希望对读者有帮助。

4. 形式新颖我把本书的内容虚构成了一个老师上课的场景，所有内容都通过这位老师表达出来，书中的文字非常口语化，这样做的目的是为了更加直观地让读者感觉，自己是在学习，是在上课。

有人可能会说，现在的课堂大都是让人昏昏欲睡，把读者带入上课场景，不是更加让读者犯困吗？

我觉得如果你的学习经历中听过一些优秀老师的课，你就不会下这样的结论。

好的老师讲课，是可以做到引人入胜的。

有人可能会问，我为什么不用《大话设计模式》中的对话形式，而采用讲课形式呢？

这是对数据结构这门学问的特点考虑的。

设计模式主要都是思想体现，通常会仁者见仁、智者见智，用对话展开比较容易；而数据结构中更多的是定义、术语、经典算法等，这些公认的知识，可讨论的地方并不多，更多的是需要把它讲清楚。

让两个人在一起讨论某个设计模式的优缺点，会非常合适，而讨论数据结构定义的好坏，就没有太大意义了，不如让一个老师告诉学生数据结构的定义好在哪里更符合实际。

因此用传统的讲课形式会好一些。

另外，本书没有习题，有思考的题目也一定会给出某种答案。

但本书每个复杂知识点的末尾，都会提供另一本书的进一步阅读建议。

这也是基于它是一本自学读物的原则。

读者阅读本书可能是任何时间任何地方，如果书中存在没有解答的习题，碰到了困难是没法及时找到老师来帮助的，因此本书尽量避免让读者有这样的困惑存在。

如果需要练习的同学，我觉得还是应该考虑再去买本习题集来学习。

学习数据结构和算法，做题和上机写代码非常有必要，从这个角度也说明，阅读完本书其实也只是完成入门而已。

本书既然是以老师上课的形式来进行，那就免不了要融入一名教师除了授业解惑以外，还要传达一些个人价值观的体现。

书中很多细微处，如对某位科学家的尊敬、对某个算法的推崇、对勤奋励志故事的讲述等都在表达着一个老师向学生传递真、善、美的意愿。

我始终认为，读者拿到的虽然只是一本没有表情、不会说话的书，但其实也是在隔空与另一个朋友交流。

人与人的交流不可能只是就事论事，一定会有情感的沟通，这种情感如果能产生共鸣、达成互信，就

## &lt;&lt;大话数据结构&gt;&gt;

会让事情（比如学习数据结构与算法这件事）本身更容易理解和接受。

本书内容本书主要是按照教育部关于计算机专业数据结构课程大纲的要求略微增减来组织内容的。主要包括：数据结构介绍，算法推导大O阶的方法，线性表结构的介绍，顺序结构与链式结构差异，栈与队列的应用，串的朴素模式匹配、KMP模式匹配算法，树结构的介绍，二叉树前中后序遍历，线索二叉树，赫夫曼树及应用，图结构的介绍，图的深度、广度遍历，最小生成树两种算法，最短路径两种算法，拓扑排序与关键路径算法，查找应用的相关介绍，折半查找、插值查找、斐波那契查找等静态查找，稠密索引、分块索引、倒排索引等索引技术，二叉排序树、平衡二叉树等动态查找，B树、B+树技术，散列表技术，排序应用的相关介绍，冒泡、选择、插入等简单排序，希尔、堆、归并、快速等改进排序，各位排序算法的对比等。

本书读者数据结构是计算机软件相关专业的基础课程，几乎可以说，要想从事编程工作，无论你是否是科班出身，都不可以绕过这部分知识。

因此，适合阅读本书的读者非常广泛，包括在读的本专科、中专职高技校等计算机专业学生、想转行做开发的非专业人员、欲考计算机研究生的应届或在职人员，以及工作后需要补学或温习数据结构和算法的程序员等各类读者。

本书对读者的技术背景要求比较低，只要是学过一门高级编程语言，例如C、C++、Java、C#、VB等就可以开始阅读本书。

不过由于当中涉及到比较复杂的算法知识，需要读者有一定的数学修养和逻辑思维能力，否则可能书籍的后半部分阅读起来会比较吃力。

本书研读方法事实上，任何有难度的知识和技巧，都不是那么容易被掌握的。

我尽管已经朝着通俗易懂的方向努力，可有些数据结构，特别是经典算法，是几代科学家的智慧结晶，因此要掌握它们还是需要读者的全力投入。

美国畅销书《如何阅读一本书》中提到“阅读可以是一件主动的事，阅读越主动，效果越好。

拿同样的书给背景相近的两个人阅读，一个人却比另一个人从书中得到了更多，这是因为，首先在于这人的主动，其次，在于他在阅读中的每一种活动都参与了更多的技巧。

这两件事是息息相关的。

阅读是一个复杂的活动，就跟写作一样，包含了大量不同的活动。

要达成良好的阅读，这些活动都是不可或缺的。

一个人越能良好运作这些活动，阅读的效果也就越好。

”我当然希望读者在阅读本书后收获巨大，但这显然是一厢情愿。

要想获得更多，您可能也需要付出类似我写作一样的力气来阅读，例如摘抄文字、眉批心得、稿纸演算、代码输入电脑，以及您自己在编程工作中的运用等。

这些相应活动的执行，将会使您得到巨大的收获。

作为作者，建议本书的研读方法为：1. 复习C语言的基础知识。

如果你掌握的是别的语言也不要紧，适当了解一些C语言和你掌握的编程语言的语法差异还是有必要的。

甚至将本书代码改造成另一种语言本身就是一种非常好的学习方法。

2. 阅读第一遍时，建议从头至尾进行。

如果你对前面的知识有足够了解，当然可以跳过直接阅读后面的章节。

不过若要学习一门完整的知识并形成体系。

通读本书，还是最好的学习方法。

3. 阅读时，摘抄是非常好的习惯。

“最淡的墨水也胜于最强的记忆！”

”有不少读者会认为摘抄了将来也不会再去看，有什么必要，但其实在写字的过程就是大脑学习的过程，写字在减缓你阅读的速度，从而让你更好地消化阅读的内容。

相信大家都能理解，“囫囵吞枣”和“慢慢品味”的差异，学习同样如此。

4. 阅读每一章时，特别是在阅读算法的推导过程时，一定要在电脑中运行代码（本书源码的下载地址可以到<http://cj723.cnblogs.com>中的《大话数据结构相关主题》中找到），了解代码的运行过程。

## &lt;&lt;大话数据结构&gt;&gt;

本书的很多算法都做到了逐行讲解，但单纯阅读可能真的很难达到理解的程度（这是纸质书无法克服的缺陷），需要你通过开发工具调试，并设置断点和逐行执行，并参照书中的讲解，观察变量的变化情况来理解算法的编写原理。

5. 阅读完每一章时，一定要在理解基础上记忆一些关键东西。

最佳的效果就是你可以不看书也做到一点不错地默写出相关算法。

6. 阅读完每一章时，一定要适当练习。

本书没有提供练习题，但市场上相关的数据结构习题集比比皆是，可以选择尝试。

另外互联网上也可以获得足够的习题来给你练习。

练习的目的是为了检测自己是否真的完全理解了书中的内容。

事实上很多时候，阅读中的人们只是自我感觉理解，而并非真正的明白。

7. 学习不可能一蹴而就，数据结构和算法如果通过一本书就可以掌握，那本身就是笑话。

本书附录提供了本书写作时的参考书目，基本都是最优秀的数据结构或相关的中文书籍各有侧重，建议大家可以适当地阅读。

8. 在之后的编程学习和工作中，尽量把已经学到的数据结构和算法知识运用到现实开发中。

遗忘时翻阅本书回顾相关内容，最终达到精通数据结构和相关算法的境界。

编程语言说明本书是用C语言编写，基于C90 (ISO C) 的标准。

读者可以选择任何一款基于C90标准的C语言开发工具或更高版本的开发工具来学习本书中的代码。

本人一直习惯于用Visual Studio 2008作为开发工具，因此在写作此书时，也是用此工具的Visual C++来编译调试代码，一切都相安无事，但写作完成后，考虑到不同读者应用开发工具的习惯不同，最终在编辑的建议下，决定提供一份可在C90标准的C语言开发环境中编译通过的代码，结果发现错误百出。

例如C90标准的注释要求是“/\* 注释文字 \*/”而不允许是“//注释文字”：要求变量声明必须要在函数的最前面，只能是“int i; for(i=0;i

## &lt;&lt;大话数据结构&gt;&gt;

## 内容概要

本书为超级畅销书《大话设计模式》作者程杰潜心三年推出的扛鼎之作！以一个计算机教师教学为场景，讲解数据结构和相关算法的知识。通篇以一种趣味方式来叙述，大量引用了各种各样的生活知识来类比，并充分运用图形语言来体现抽象内容，对数据结构所涉及的一些经典算法做到逐行分析、多算法比较。与市场上的同类数据结构图书相比，本书内容趣味易读，算法讲解细致深刻，是一本非常适合自学的读物。

本书以一个计算机教师教学为场景，讲解数据结构和相关算法的知识。通篇以一种趣味方式来叙述，大量引用了各种各样的生活知识来类比，并充分运用图形语言来体现抽象内容，对数据结构所涉及的一些经典算法做到逐行分析、多算法比较。与市场上的同类数据结构图书相比，本书内容趣味易读，算法讲解细致深刻，是一本非常适合自学的读物。

本书主要内容包含：数据结构介绍、算法推导大O阶的方法；顺序结构与链式结构差异、栈与队列的应用；串的朴素模式匹配、KMP模式匹配算法；二叉树前中后序遍历、赫夫曼树及应用；图的深度、广度遍历；最小生成树两种算法、最短路径两种算法；拓扑排序与关键路径算法；折半查找、插值查找、斐波那契查找等静态查找；稠密索引、分块索引、倒排索引等索引技术；二叉排序树、平衡二叉树等动态查找；B树、B+树技术，散列表技术；冒泡、选择、插入等简单排序；希尔、堆、归并、快速等改进排序.....

本书适合学过一门编程语言的各类读者，包括在读的大中专计算机专业学生、想转行做开发的非专业人员、欲考计算机研究生的应届或在职人员，以及工作后需要补学或温习数据结构和算法的程序员等

## <<大话数据结构>>

### 作者简介

一个被读者誉为很适合写IT技术书的家伙。

《大话设计模式》作者。

此书07年末出版至今已经简体版印刷9次、繁体版印刷6次，取得了较好的成绩，开创了一种适合国人阅读的趣味讲解IT知识的风格模式。

其本人参与过政府、证券、游戏、交通等多种行业的软件开发及项目管理工作，也曾做过软件培训的教师。

因曾有过两年半高中数学教学的独特经历，使得其书作当中处处以初学者视角考虑和分析问题，他成为了当前很受欢迎的IT技术图书作者之一。

## &lt;&lt;大话数据结构&gt;&gt;

## 书籍目录

## 第1章 数据结构绪论

## 1.1 开场白

## 1.2 你数据结构怎么学的？

## 1.3 数据结构起源

## 1.4 基本概念和术语

## 1.4.1 数据

## 1.4.2 数据元素

## 1.4.3 数据项

## 1.4.4 数据对象

## 1.4.5 数据结构

## 1.5 逻辑结构与物理结构

## 1.5.1 逻辑结构

## 1.5.2 物理结构

## 1.6 抽象数据类型

## 1.6.1 数据类型

## 1.6.2 抽象数据类型

## 1.7 总结回顾

## 1.8 结尾语

## 第2章 算法

## 2.1 开场白

## 2.2 数据结构与算法关系

## 2.3 两种算法的比较

## 2.4 算法定义

## 2.5 算法的特性

## 2.5.1 输入输出

## 2.5.2 有穷性

## 2.5.3 确定性

## 2.5.4 可行性

## 2.6 算法设计的要求

## 2.6.1 正确性

## 2.6.2 可读性

## 2.6.3 健壮性

## 2.6.4 时间效率高和存储量低

## 2.7 算法效率的度量方法

## 2.7.1 事后统计方法

## 2.7.2 事前分析估算方法

## 2.8 函数的渐近增长

## 2.9 算法时间复杂度

## 2.9.1 算法时间复杂度定义

## 2.9.2 推导大O阶方法

## 2.9.3 常数阶

## 2.9.4 线性阶

## 2.9.5 对数阶

## 2.9.6 平方阶

## &lt;&lt;大话数据结构&gt;&gt;

- 2.10 常见的时间复杂度
- 2.11 最坏情况与平均情况
- 2.12 算法空间复杂度
- 2.13 总结回顾
- 2.14 结尾语

## 第3章 线性表

- 3.1 开场白
- 3.2 线性表的定义
- 3.3 线性表的抽象数据类型
- 3.4 线性表的顺序存储结构
  - 3.4.1 顺序存储定义
  - 3.4.2 顺序存储方式
  - 3.4.3 数据长度与线性表长度区别
  - 3.4.4 地址计算方法
- 3.5 顺序存储结构的插入与删除
  - 3.5.1 获得元素操作
  - 3.5.2 插入操作
  - 3.5.3 删除操作
  - 3.5.4 线性表顺序存储结构的优缺点
- 3.6 线性表的链式存储结构
  - 3.6.1 顺序存储结构不足的解决办法
  - 3.6.2 线性表链式存储结构定义
  - 3.6.3 头指针与头结点的异同
  - 3.6.4 线性表链式存储结构代码描述
- 3.7 单链表的读取
- 3.8 单链表的插入与删除
  - 3.8.1 单链表的插入
  - 3.8.2 单链表的删除
- 3.9 单链表的整表创建
- 3.10 单链表的整表删除
- 3.11 单链表结构与顺序存储结构优缺点
- 3.12 静态链表
  - 3.12.1 静态链表的插入操作
  - 3.12.2 静态链表的删除操作
  - 3.12.3 静态链表优缺点
- 3.13 循环链表
- 3.14 双向链表
- 3.15 总结回顾
- 3.16 结尾语

## 第4章 栈与队列

- 4.1 开场白
- 4.2 栈的定义
  - 4.2.1 栈的定义
  - 4.2.2 进栈出栈变化形式
- 4.3 栈的抽象数据类型
- 4.4 栈的顺序存储结构及实现
  - 4.4.1 栈的顺序存储结构

## &lt;&lt;大话数据结构&gt;&gt;

- 4.4.2 栈的顺序存储结构进栈操作
  - 4.4.3 栈的顺序存储结构出栈操作
  - 4.5 两栈共享空间
  - 4.6 栈的链式存储结构及实现
    - 4.6.1 栈的链式存储结构
    - 4.6.2 栈的链式存储结构进栈操作
    - 4.6.3 栈的链式存储结构出栈操作
  - 4.7 栈的作用
  - 4.8 栈的应用——递归
    - 4.8.1 斐波那契数列实现
    - 4.8.2 递归定义
  - 4.9 栈的应用——四则运算表达式求值
    - 4.9.1 后缀（逆波兰）表示法定义
    - 4.9.2 后缀表达式计算结果
    - 4.9.3 中缀表达式转后缀表达式
  - 4.10 队列的定义
  - 4.11 队列的抽象数据类型
  - 4.12 循环队列
    - 4.12.1 队列顺序存储的不足
    - 4.12.2 循环队列定义
  - 4.13 队列的链式存储结构及实现
    - 4.13.1 队列链式存储结构入队操作
    - 4.13.2 队列链式存储结构出队操作
  - 4.14 总结回顾
  - 4.15 结尾语
- 第5章 串
- 5.1 开场白
  - 5.2 串的定义
  - 5.3 串的比较
  - 5.4 串的抽象数据类型
  - 5.5 串的存储结构
    - 5.5.1 串的顺序存储结构
    - 5.5.2 串的链式存储结构
  - 5.6 朴素的模式匹配算法
  - 5.7 KMP模式匹配算法
    - 5.7.1 KMP模式匹配算法原理
    - 5.7.2 next数组值推导
    - 5.7.3 KMP模式匹配算法实现
    - 5.7.4 KMP模式匹配算法改进
    - 5.7.5 nextval数组值推导
  - 5.8 总结回顾
  - 5.9 结尾语
- 第6章 树
- 6.1 开场白
  - 6.2 树的定义
    - 6.2.1 结点分类
    - 6.2.2 结点间关系

## &lt;&lt;大话数据结构&gt;&gt;

- 6.2.3 树的其他相关概念
- 6.3 树的抽象数据类型
- 6.4 树的存储结构
  - 6.4.1 双亲表示法
  - 6.4.2 孩子表示法
  - 6.4.3 孩子兄弟表示法
- 6.5 二叉树的定义
  - 6.5.1 二叉树特点
  - 6.5.2 特殊二叉树
- 6.6 二叉树的性质
  - 6.6.1 二叉树性质1
  - 6.6.2 二叉树性质2
  - 6.6.3 二叉树性质3
  - 6.6.4 二叉树性质4
  - 6.6.5 二叉树性质5
- 6.7 二叉树的存储结构
  - 6.7.1 二叉树顺序存储结构
  - 6.7.2 二叉链表
- 6.8 遍历二叉树
  - 6.8.1 二叉树遍历原理
  - 6.8.2 二叉树遍历方法
  - 6.8.3 前序遍历算法
  - 6.8.4 中序遍历算法
  - 6.8.5 后序遍历算法
  - 6.8.6 推导遍历结果
- 6.9 二叉树的建立
- 6.10 线索二叉树
  - 6.10.1 线索二叉树原理
  - 6.10.2 线索二叉树结构实现
- 6.11 树、森林与二叉树的转换
  - 6.11.1 树转换为二叉树
  - 6.11.2 森林转换为二叉树
  - 6.11.3 二叉树转换为树
  - 6.11.4 二叉树转换为森林
  - 6.11.5 树与森林的遍历
- 6.12 赫夫曼树及其应用
  - 6.12.1 赫夫曼树
  - 6.12.2 赫夫曼树定义与原理
  - 6.12.3 赫夫曼编码
- 6.13 总结回顾
- 6.14 结尾语
- 第7章 图
  - 7.1 开场白
  - 7.2 图的定义
    - 7.2.1 各种图定义
    - 7.2.2 图的顶点与边间关系
    - 7.2.3 连通图相关术语

## &lt;&lt;大话数据结构&gt;&gt;

- 7.2.4 图的定义与术语总结
- 7.3 图的抽象数据类型
- 7.4 图的存储结构
  - 7.4.1 邻接矩阵
  - 7.4.2 邻接表
  - 7.4.3 十字链表
  - 7.4.4 邻接多重表
  - 7.4.5 边集数组
- 7.5 图的遍历
  - 7.5.1 深度优先遍历
  - 7.5.2 广度优先遍历
- 7.6 最小生成树
  - 7.6.1 普里姆 (Prim) 算法
  - 7.6.2 克鲁斯卡尔 (Kruskal) 算法
- 7.7 最短路径
  - 7.7.1 迪杰斯特拉 (Dijkstra) 算法
  - 7.7.2 弗洛伊德 (Floyd) 算法
- 7.8 拓扑排序
  - 7.8.1 拓扑排序介绍
  - 7.8.2 拓扑排序算法
- 7.9 关键路径
  - 7.9.1 关键路径算法原理
  - 7.9.2 关键路径算法
- 7.10 总结回顾
- 7.11 结尾语
- 第8章 查找
  - 8.1 开场白
  - 8.2 查找概论
  - 8.3 顺序表查找
    - 8.3.1 顺序表查找算法
    - 8.3.2 顺序表查找优化
  - 8.4 有序表查找
    - 8.4.1 折半查找
    - 8.4.2 插值查找
    - 8.4.3 斐波那契查找
  - 8.5 线性索引查找
    - 8.5.1 稠密索引
    - 8.5.2 分块索引
    - 8.5.3 倒排索引
  - 8.6 二叉排序树
    - 8.6.1 二叉排序树查找操作
    - 8.6.2 二叉排序树插入操作
    - 8.6.3 二叉排序树删除操作
    - 8.6.4 二叉排序树总结
  - 8.7 平衡二叉树 (AVL树)
    - 8.7.1 平衡二叉树实现原理
    - 8.7.2 平衡二叉树实现算法

## &lt;&lt;大话数据结构&gt;&gt;

- 8.8 多路查找树 (B树)
    - 8.8.1 2-3树
    - 8.8.2 2-3-4树
    - 8.8.3 B树
    - 8.8.4 B+树
  - 8.9 散列表查找 (哈希表) 概述
    - 8.9.1 散列表查找定义
    - 8.9.2 散列表查找步骤
  - 8.10 散列函数的构造方法
    - 8.10.1 直接定址法
    - 8.10.2 数字分析法
    - 8.10.3 平方取中法
    - 8.10.4 折叠法
    - 8.10.5 除留余数法
    - 8.10.6 随机数法
  - 8.11 处理散列冲突的方法
    - 8.11.1 开放定址法
    - 8.11.2 再散列函数法
    - 8.11.3 链地址法
    - 8.11.4 公共溢出区法
  - 8.12 散列表查找实现
    - 8.12.1 散列表查找算法实现
    - 8.12.2 散列表查找性能分析
  - 8.13 总结回顾
  - 8.14 结尾语
- 第9章 排序
- 9.1 开场白
  - 9.2 排序的基本概念与分类
    - 9.2.1 排序的稳定性
    - 9.2.2 内排序与外排序
    - 9.2.3 排序用到的结构与函数
  - 9.3 冒泡排序
    - 9.3.1 最简单排序实现
    - 9.3.2 冒泡排序算法
    - 9.3.3 冒泡排序优化
    - 9.3.4 冒泡排序复杂度分析
  - 9.4 简单选择排序
    - 9.4.1 简单选择排序算法
    - 9.4.2 简单选择排序复杂度分析
  - 9.5 直接插入排序
    - 9.5.1 直接插入排序算法
    - 9.5.2 直接插入排序复杂度分析
  - 9.6 希尔排序
    - 9.6.1 希尔排序原理
    - 9.6.2 希尔排序算法
    - 9.6.3 希尔排序复杂度分析
  - 9.7 堆排序

## <<大话数据结构>>

9.7.1 堆排序算法

9.7.2 堆排序复杂度分析

9.8 归并排序

9.8.1 归并排序算法

9.8.2 归并排序复杂度分析

9.8.3 非递归实现归并排序

9.9 快速排序

9.9.1 快速排序算法

9.9.2 快速排序复杂度分析

9.9.3 快速排序优化

1. 优化选取枢轴

2. 优化不必要的交换

3. 优化小数组时的排序方案

4. 优化递归操作

9.10 总结回顾

9.11 结尾语

附录 参考文献

## <<大话数据结构>>

### 编辑推荐

《大话数据结构》为超级畅销书《大话设计模式》作者程杰潜心三年推出的扛鼎之作！以一个计算机教师教学为场景，讲解数据结构和相关算法的知识。通篇以一种趣味方式来叙述，大量引用了各种各样的生活知识来类比，并充分运用图形语言来体现抽象内容，对数据结构所涉及的一些经典算法做到逐行分析、多算法比较。与市场上的同类数据结构图书相比，本书内容趣味易读，算法讲解细致深刻，是一本非常适合自学的读物。

## <<大话数据结构>>

### 名人推荐

超级畅销书《大话设计模式》作者的新作！

用户群更为广泛，写作风格一如既往，技术沉淀更加深厚，势必掀起全民数据结构的热潮！

<<大话数据结构>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>