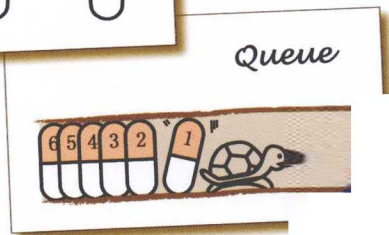
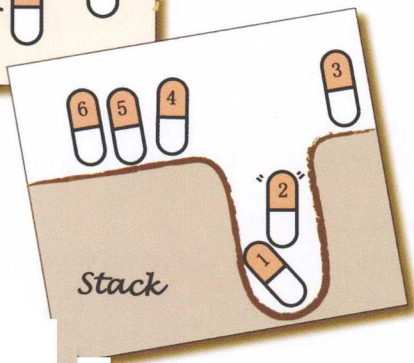
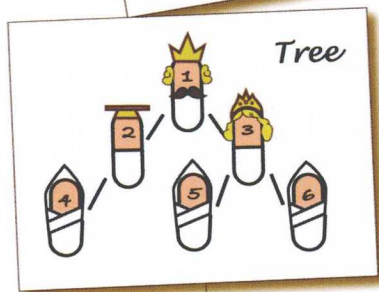
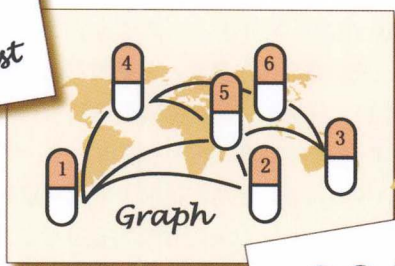
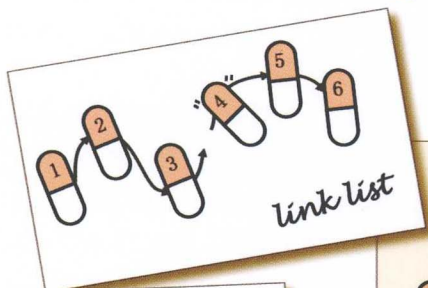


# Play “

数据结构好枯燥，算法很复杂，这是真的吗？本书尝试用轻松的文字和形象的图示重新组织这部分知识的讲解过程，大话让数据结构和算法的学习变得有趣了。不信？翻开读读看。”

# WITH DATA STRUCTURE



# 大话数据结构

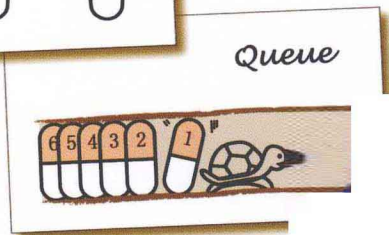
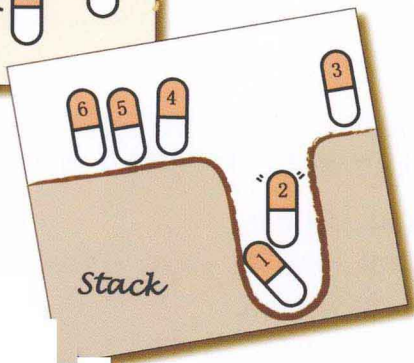
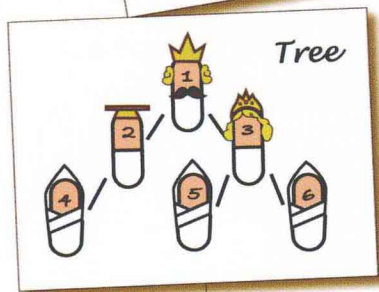
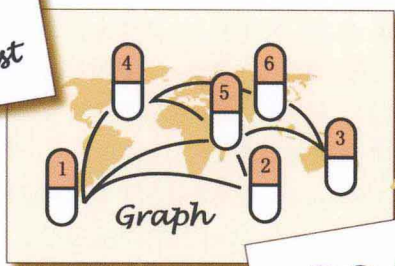
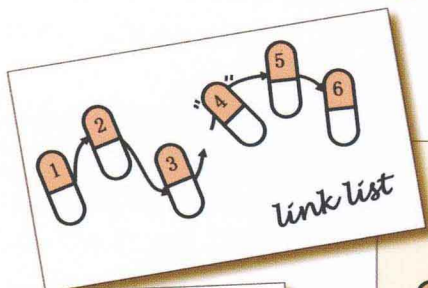
程杰 著

清华大学出版社  
北京

# Play “

数据结构好枯燥，算法很复杂，这是真的吗？本书尝试用轻松的文字和形象的图示重新组织这部分知识的讲解过程，大话让数据结构和算法的学习变得有趣了。不信？翻开读读看。”

# WITH DATA STRUCTURE



# 大话数据结构

程杰 著

清华大学出版社  
北京

## 内 容 简 介

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。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

### 图书在版编目(CIP)数据

大话数据结构 / 程杰 著. —北京：清华大学出版社，2011.6

ISBN 978-7-302-25565-9

I. ①大… II. ①程… III. ①数据结构 IV. ①TP311.12

中国版本图书馆 CIP 数据核字 (2011) 第 084835 号

责任编辑：栾大成

责任校对：徐俊伟

责任印制：杨 艳

出版发行：清华大学出版社

地 址：北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编：100084

社 总 机：010-62770175

邮 购：010-62786544

投稿与读者服务：010-62795954, [jsjic@tup.tsinghua.edu.cn](mailto:jsjic@tup.tsinghua.edu.cn)

质 量 反 馈：010-62772015, [zhiliang@tup.tsinghua.edu.cn](mailto:zhiliang@tup.tsinghua.edu.cn)

印 刷 者：北京鑫丰华彩印有限公司

装 订 者：北京国马印刷厂

经 销：全国新华书店

开 本：188×260 印 张：29 插 页：1 字 数：555 千字

版 次：2011 年 6 月第 1 版 印 次：2011 年 8 月第 2 次印刷

印 数：5001~10000

定 价：59.00 元

---

产品编号：042342-02

# 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 次评论，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……嗨！不管怎样，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。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：

- 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。
- 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。
- 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。
- 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。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。



- 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。
- 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，而非真正的明白。
- 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，建议大家可以适当阅读。
- 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<n;i++)……”，而不允许如 “for (int i=0;i<n;i++)” 这样的方式：再比如 C++ 中函数的参数可以传递如 “void CreateBiTree(BiTree &T)” 的地址变量，但在 C 语言中，只能传递如 “void CreateBiTree(BiTree \*T)” 的指针变量。因此当你看到书中的有些代码到处都是 “\*” 时，就用不着奇怪了。

出于为了让代码可以在低端编译环境通过的考虑，牺牲一些代码的简捷性和优雅性也是无可奈何和必要的。最终我将书中全部代码都改成 C90 标准的代码。

C 语言初学者可能会因为刚接触编程语言，特别是对指针的理解不深，而担心阅读困难。我个人感觉，单纯学习指针是很难理解它的真正用途和好处，而通过学习数

据结构，特别是像链式存储结构在各种结构算法中的运用，反而可以让读者进一步的理解指针的优越之处。从这个角度说，数据结构的学习可以反过来加强读者对 C 语言，特别是指针概念的理解。

## 编程语言差异

C 语言是一门古老的高级语言，它的应用范围非常广泛，因此我选择它作为本书的算法展示语言。如果读者之前学过它，那么阅读本书就不存在语言障碍。懂得 C++ 语言的读者，同样也不会有任何语言上的问题。

掌握 Java、C#、VB 等面向对象语言的读者，当面对书中大量的 C 语言式的结构 (struct) 声明和针对结构的参数传递的代码时，可以理解为是类的定义和由类生成对象的传递。尽管的确存在差异，但并不影响整体对数据结构知识和算法原理的理解。

我个人感觉，哪怕是对 C 语言不熟悉，也不妨利用学习数据结构的机会，学习一下 C 语言的编程方法，这对于将来应用其他高级语言也是有很大帮助的。

## 不是一个人在战斗

首先要感谢我的妻子李秀芳对我写作本书期间的全力支持，我辞职写作，没有她精神上的理解鼓励和生活上的悉心照顾，是不可能走出这一步并顺利完成书稿的。我们的儿子程晟涵如今已经三周岁，我是在他每日的欢声笑语和哭哭啼啼中进行每一章节的构思和写作，希望他可以茁壮成长。我的父母已经年迈，他们为我的全职创作也甚为担心和忧虑，这里也要说一声抱歉。

写作过程中，本人购买和借阅了与数据结构相关的大量书籍，详细书目见附录。没有前辈的贡献，就没有本书的出版，也希望本书能成为这些书籍的前期读物。在此向这些图书作者表示衷心的感谢。

仅有作者是不可能完成图书的出版的，本人要非常感谢清华大学出版社的朋友们，他们是本书的最初读者，也是协助本人将此书由毛糙变精良的最有力帮手。本书的封面设计程瑜、插图设计周翔，都是在反反复复的修改中完成创作的。写作中还得到了周筠、卢鹤翔、张伸、胡文佳、Milo、陈钢、刘超、刘唯一、杨绣国、戚婷婷、雷顺、杨诗盈、高宇翔、林健的友情帮助，他们都在本人的创作中提出了宝贵建议。

在此向所有帮助与支持我的朋友道一声：谢谢！

程杰

## 编者的话

2007年，一本特立独行的IT技术图书《大话设计模式》横空出世，开创了一种新技术图书风格。当年以及后来的数年间，横扫各大排行，目前销售已经超过5万册，几乎是纯粹的店销销量，在最近10年的IT图书市场中，这是个了不起的数据。

作者程杰并没有满足这个成绩，耗时3年潜心创作了另外一本同样是程序员基础的著作——《大话数据结构》。

说实话，作为策划，我并没有过多干预作者的创作过程，仅仅是在细节上提出我的一些看法。我的工作更多的是与作者来回推敲成稿后的一些说法或者技术点上的问题，另外还反复探讨了书中的类比案例的可理解性。最终细节敲定还是在上海一家咖啡馆里完成的，我和作者在那里折腾了一下午，终于达成共识。

数据结构在某种程度上和设计模式类似，都是前辈的武功套路。不同的是，设计模式是近几十年的卓越程序员的智慧结晶，而数据结构是几百上千年的无数科学家、数学家的智慧沉淀，更加具有深厚的背景。

大家知道，程序是利用计算机高速运算能力来协助我们处理一些需要海量运算得出结果的问题，应用程序花哨的界面和有效的用户体验，归根到底都需要在后台看不见的地方进行运算，得出我们需要的结果——无论是在气象预报还是“极品飞车”。

一台计算机的CPU运算能力是固定的，只会机械地接受程序的指令，所以，算法的优劣就决定了程序设计水平的高低。举个简单的例子，数据库性能优化这个工作，收费是按照小时来计算的，水平高的每小时可以达到30万美金，为什么会值这么多钱？有价值吗？这其实就是算法的力量，使用优秀的算法可以为大型企业节省海量的硬件投入同时带来巨大的效率提升——比如之前需要100台小型机，优化之后只需要10台就够了；之前查询一个数据需要一分钟出来结果，优化之后1秒钟就够了……这些对于企业来说，节省的成本可就远远不止投入的几十上百万的优化费用了。

国内外优秀的程序员很多毕业于数学专业，也在一定程度上说明了这个问题。国内的程序开发现状跟国外略有不同，大家都在关注界面和用户体验，在算法上往往要求不高。这其实是国内软件行业与国外软件行业的最大差距所在。

我们的程序员因为在受教育的过程中（大都是在大学），由于种种原因，数据结构和算法的基本功通常要差一些，等从业以后想再补课又缺乏好的教材，或者说适合自学的教材。数据结构不是说没有优秀教材，比如《数据结构》《算法导论》这样的经典著作我们绝对不能说不好，但是作为自学，实在是有点难啃。

《大话数据结构》延续了作者一贯的轻松调侃的风格，采用了师生对话的方式，展开讨论，其中穿插了大量“庸俗”的类比案例，帮助大家迅速“开窍”。

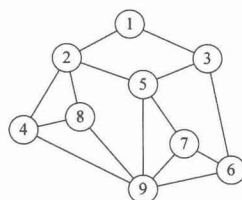
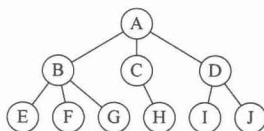
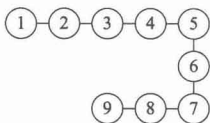
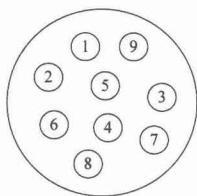
在我的一篇博文([kobeluan.blog.51cto.com/237742/212175](http://kobeluan.blog.51cto.com/237742/212175))中谈到国内原创技术书整体层次偏低，不是因为国内没有高手，最关键的原因有两条：第一，国内版税太低，与高手作者的收入差别太大，导致很多人没有时间和心气来写书；第二，国内知识产权保护不力，出版社无法用提高定价的方式来为读者和作者提供更好的服务。

类似程杰这样的作者，真诚地将自己的感悟奉献出来，与作者的用心相比，作为策划编辑付出的劳动就不值得一提了。这里真心希望读者可以从书中找到需要的东西，也希望国内更多高人涌现出来，为大家创作更适合中国人阅读的优秀技术图书。

清华大学出版社 栾大成

# 目 录

第 1 章 数据结构绪论 ..... 1



1.1 开场白 ..... 2

如果你交给某人一个程序，你将折磨他一整天；如果你教某人如何编写程序，你将折磨他一辈子。

1.2 你数据结构怎么学的？ ..... 3

他完成开发并测试通过后，得意地提交了代码。项目经理看完代码后拍着桌子对他说：“你数据结构是怎么学的？”

1.3 数据结构起源 ..... 4

1.4 基本概念和术语 ..... 5

正所谓“巧妇难为无米之炊”，再强大的计算机，也要有“米”下锅才可以干活，否则就是一堆破铜烂铁。这个“米”就是数据。

1.4.1 数据 ..... 5	1.4.4 数据对象 ..... 6
1.4.2 数据元素 ..... 5	1.4.5 数据结构 ..... 6
1.4.3 数据项 ..... 6	

1.5 逻辑结构与物理结构 ..... 7

1.5.1 逻辑结构 ..... 7	1.5.2 物理结构 ..... 9
--------------------	--------------------

1.6 抽象数据类型 ..... 11

大家都需要房子住，但显然没钱考虑大房子是没有意义的。于是商品房就出现了各种各样的户型，有几百平米的别墅，也有仅两平米的胶囊公寓……

1.6.1 数据类型 ..... 11	1.6.2 抽象数据类型 ..... 12
---------------------	-----------------------

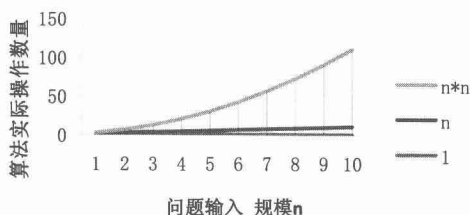
1.7 总结回顾 ..... 14

1.8 结尾语 ..... 15

最终的结果一定是，你对着别人很牛的说“数据结构——就那么回事。”

第 2 章 算法 ..... 17

不同算法的操作数量对比



2.1 开场白 ..... 18

2.2 数据结构与算法关系 ..... 18

计算机界的前辈们，是一帮很牛很牛的人，他们使得很多看似没法解决或者很难解决的问题，变得如此美妙和神奇。

2.3 两种算法的比较 ..... 19

高斯在上小学的一天，老师要求每个学生都计算  $1+2+\dots+100$  的结果，谁先算出来谁先回家.....

2.4 算法定义 ..... 20

现实世界中的算法千变万化，没有通用算法可以解决所有问题。甚至一个小问题，某个解决此类问题很优秀的算法却未必就适合它。

2.5 算法的特性 ..... 21

2.5.1 输入输出 ..... 21	2.5.3 确定性 ..... 21
2.5.2 有穷性 ..... 21	2.5.4 可行性 ..... 21

2.6 算法设计的要求 ..... 22

求 100 个人的高考成绩平均分与求全省所有考生的成绩平均分在占用时间和内存存储上有非常大的差异，我们自然追求高效率 and 低存储的算法来解决问题。

2.6.1 正确性 ..... 22	2.6.3 健壮性 ..... 23
2.6.2 可读性 ..... 23	2.6.4 时间效率高和存储量低 ..... 23

2.7 算法效率的度量方法 ..... 24

    随着  $n$  值越来越大，它们在时间效率上的差异也就越来越大。好比有些人每天都在学习，而另一些人，打游戏、睡睡大觉，毕业后前者名企争着要，后者求职处处无门。

    2.7.1 事后统计方法 ..... 24      2.7.2 事前分析估算方法 ..... 25

2.8 函数的渐近增长 ..... 27

2.9 算法时间复杂度 ..... 29

    理解大  $O$  推导不算难，难的其实是对数列的一些相关运算，这考察的更多的是数学知识和能力。

    2.9.1 算法时间复杂度定义 ..... 29      2.9.4 线性阶 ..... 31

    2.9.2 推导大  $O$  阶方法 ..... 30      2.9.5 对数阶 ..... 32

    2.9.3 常数阶 ..... 30      2.9.6 平方阶 ..... 32

2.10 常见的时间复杂度 ..... 35

    有些时候，告诉你某些东西不可以去尝试，也是一种知识的传递。总不能非要去被毒蛇咬一口才知道蛇不可以去招惹吧。

2.11 最坏情况与平均情况 ..... 35

2.12 算法空间复杂度 ..... 36

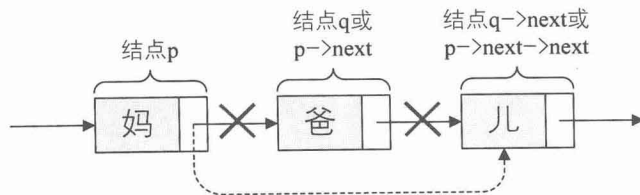
    事先建立一个有 2050 大的数组，然后把所有年份按下标数字对应，如果是闰年，此数组项的值就是 1，如果不是就是 0。这样，所谓的判断某一年是否是闰年就变成了查找这个数组的某一项的值是多少的问题。

2.13 总结回顾 ..... 37

2.14 结束语 ..... 38

    愚公移山固然可敬，但发明炸药和推土机，可能更加实在和聪明。

第 3 章 线性表 ..... 41



3.1 开场白.....42

门外家长都挤在大门口与门里的小孩子的井然有序，形成了鲜明对比。哎，有时大人的所作所为，其实还不如孩子。

3.2 线性表的定义.....42

3.3 线性表的抽象数据类型.....45

有时我们想知道某个小朋友（比如麦兜）是否是班级的同学，老师会告诉我说，没有，麦兜是在春田花花幼儿园里。这种查找某个元素是否存在的操作很常用。

3.4 线性表的顺序存储结构.....47

他每次一吃完早饭就冲着去了图书馆，挑一个好地儿，把他书包里的书，一本一本的按座位放好，长长一排，九个座硬是被他占了。

- 3.4.1 顺序存储定义.....47
- 3.4.2 顺序存储方式.....47
- 3.4.3 数据长度与线性表长度区别.....48
- 3.4.4 地址计算方法.....49

3.5 顺序存储结构的插入与删除.....50

春运时去买火车票，大家都排队排着好好的，这时来了一个美女：“可否让我排在你前面？”这可不得了，后面的人像蠕虫一样，全部都得退后一步。

- 3.5.1 获得元素操作.....50
- 3.5.2 插入操作.....51
- 3.5.3 删除操作.....52
- 3.5.4 线性表顺序存储结构的优缺点.....54

3.6 线性表的链式存储结构.....55

反正也是要让相邻元素间留有足够余地，那干脆所有元素都不要考虑相邻位置了，哪有空位就到哪里。而只是让每个元素知道它下一个元素的位置在哪里。

- 3.6.1 顺序存储结构不足的解决办法.....55
- 3.6.2 线性表链式存储结构定义.....56
- 3.6.3 头指针与头结点的异同.....58
- 3.6.4 线性表链式存储结构代码描述.....58

3.7 单链表的读取.....60

3.8 单链表的插入与删除.....61

本来是爸爸左牵着妈妈的手，右牵着宝宝的手在马路边散步。突然迎面走来一美女，爸爸失神般地望着，此情景被妈妈逮个正着，于是扯开父子俩，拉起宝宝的左手就快步朝前走去。

- 3.8.1 单链表的插入.....61
- 3.8.2 单链表的删除.....64

3.9 单链表的整表创建.....66

3.10 单链表的整表删除 ..... 69

3.11 单链表结构与顺序存储结构优缺点 ..... 70

3.12 静态链表 ..... 71

    对于一些语言，如 Basic、Fortran 等早期的编程高级语言，由于没有指针，这链表结构，按照前面我们的讲法，它就没法实现了。怎么办呢？

    3.12.1 静态链表的插入操作 ..... 73      3.12.3 静态链表优缺点 ..... 77

    3.12.2 静态链表的删除操作 ..... 75

3.13 循环链表 ..... 78

    这个轮回的思想很有意思。它强调了不管你今生是穷是富，如果持续行善积德，下辈子就会好过，反之就会遭到报应。

3.14 双向链表 ..... 81

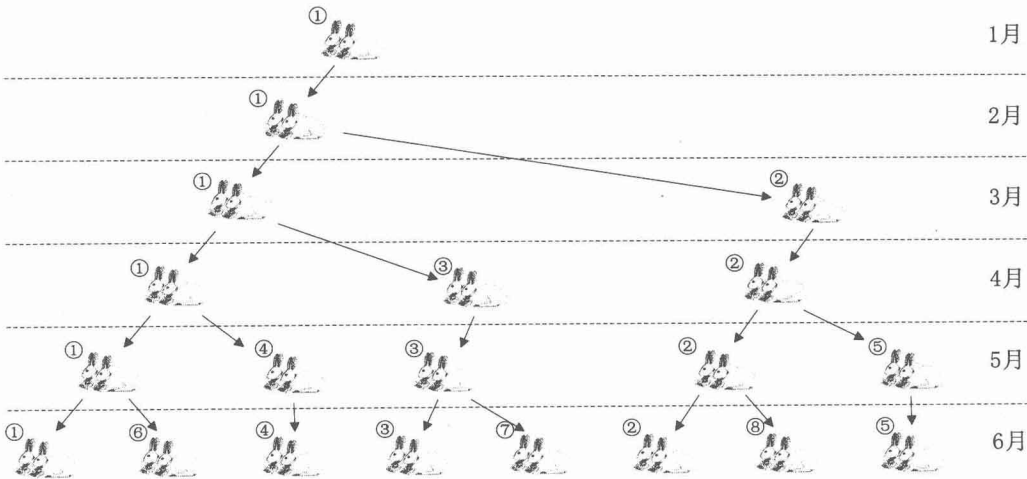
    就像每个人的人生一样，欲收获就得付代价。双向链表既然是比单链表多了如可以反向遍历查找等的数据结构，那么也就需要付出一些小的代价。

3.15 总结回顾 ..... 84

3.16 结尾语 ..... 85

如果你觉得上学读书是受罪，假设你可以活到 80 岁，其实你最多也就吃了 20 年苦。用人生四分之一的时间来换取其余时间的幸福生活，这点苦不算啥。

第 4 章 栈与队列 ..... 87





4.1	开场白 .....	88
	想想看,在你准备用枪的时候,突然这手枪明明有子弹却打不出来,这不是要命吗。	
4.2	栈的定义 .....	89
	类似的很多软件,比如 Word、Photoshop 等,都有撤消(undo)的操作,也是用栈这种思想方式来实现的。	
4.2.1	栈的定义 .....	89
4.2.2	进栈出栈变化形式 .....	90
4.3	栈的抽象数据类型 .....	91
4.4	栈的顺序存储结构及实现 .....	92
4.4.1	栈的顺序存储结构 .....	92
4.4.2	栈的顺序存储结构—— 进栈操作 .....	93
4.4.3	栈的顺序存储结构——出栈操作 .....	94
4.5	两栈共享空间 .....	94
	两个大学室友毕业同时到北京工作,他们都希望租房时能找到独自住的一室户或一室一厅,可找来找去发现,实在是承受不起。	
4.6	栈的链式存储结构及实现 .....	97
4.6.1	栈的链式存储结构 .....	97
4.6.2	栈的链式存储结构—— 进栈操作 .....	98
4.6.3	栈的链式存储结构——出栈操作 .....	99
4.7	栈的作用 .....	100
4.8	栈的应用——递归 .....	100
	当你往镜子前面一站,镜子里面就有一个你的像。但你试过两面镜子一起照吗?如果 A、B 两面镜子相互面对面放着,你往中间一站,嘿,两面镜子里都有你的千百个“化身”。	
4.8.1	斐波那契数列实现 .....	101
4.8.2	递归定义 .....	103
4.9	栈的应用——四则运算表达式求值 .....	104
4.9.1	后缀(逆波兰)表示法定义 .....	104
4.9.2	后缀表达式计算结果 .....	106
4.9.3	中缀表达式转后缀表达式 .....	108
4.10	队列的定义 .....	111