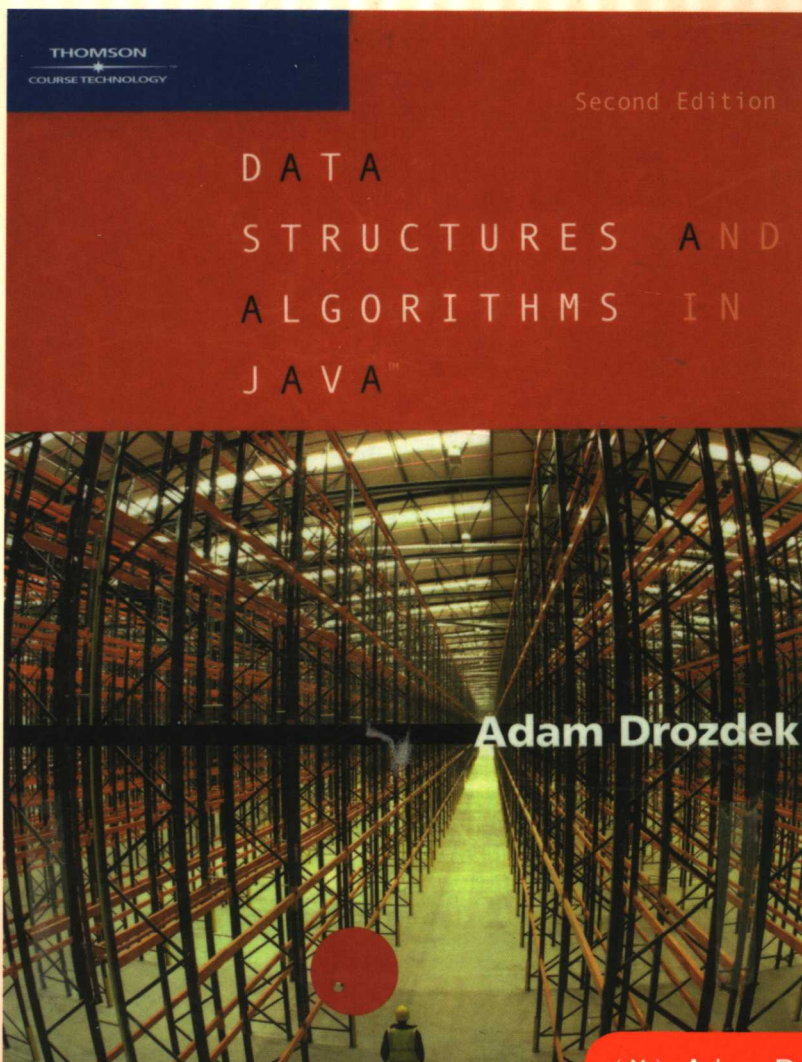


经 典 原 版 书 库

# 数据结构与算法

## Java语言版

(英文版·第2版)



(美) Adam Drozdek 著



机械工业出版社  
China Machine Press

经典原版书库

# 数据结构与算法

## Java语言版

(英文版·第2版)

Data Structures and Algorithms in Java

(Second Edition)

江苏工业学院图书馆  
藏书章

(美) Adam Drozdek 著



机械工业出版社  
China Machine Press

Adam Drozdek: Data Structures and Algorithms in Java, Second Edition (ISBN 0-534-49252-5).

Copyright © 2005 by Course Technology, a division of Thomson Learning, Inc.

Original language published by Thomson Learning (a division of Thomson Learning Asia Pte Ltd). All rights reserved.

China Machine Press is authorized by Thomson Learning to publish and distribute exclusively this English language reprint edition. This edition is authorized for sale in the People's Republic of China only (excluding Hong Kong, Macao SAR and Taiwan). Unauthorized export of this edition is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

本书原版由汤姆森学习出版集团出版。

本书英文影印版由汤姆森学习出版集团授权机械工业出版社独家出版发行。此版本仅限在中华人民共和国境内（不包括中国香港、澳门特别行政区及中国台湾）销售。未经授权的本书出口将被视为违反版权法的行为。未经出版者预先书面许可，不得以任何方式复制或发行本书的任何部分。

981-265-923-4

版权所有，侵权必究。

本书法律顾问 北京市展达律师事务所

本书版权登记号：图字：01-2006-1973

图书在版编目（CIP）数据

数据结构与算法：Java语言版（英文版·第2版）/（美）德罗兹德克（Drozdek, A.）著. - 北京：机械工业出版社，2006.4

（经典原版书库）

书名原文：Data Structures and Algorithms in Java, Second Edition

ISBN 7-111-18826-8

I. 数… II. 德… III. ①数据结构 - 英文 ②Java语言 - 程序设计 - 英文  
IV. ①TP311.12 ②TP312

中国版本图书馆CIP数据核字（2006）第029824号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：迟振春

北京京北制版印刷厂印刷·新华书店北京发行所发行

2006年4月第1版第1次印刷

170mm × 242mm · 48.25印张

定价：79.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

# 出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域中取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章图文信息有限公司较早意识到“出版要为教育服务”。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及度藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔

滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T., Stanford, U.C. Berkeley, C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证，但我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

电子邮件：hzjsj@hzbook.com

联系电话：(010) 68995264

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

# 专家指导委员会

(按姓氏笔画顺序)

尤晋元  
石教英  
张立昂  
邵维忠  
周克定  
郑国梁  
高传善  
裘宗燕

王 珊  
吕 建  
李伟琴  
陆丽娜  
周傲英  
施伯乐  
梅 宏  
戴 葵

冯博琴  
孙玉芳  
李师贤  
陆鑫达  
孟小峰  
钟玉琢  
程 旭

史忠植  
吴世忠  
李建中  
陈向群  
岳丽华  
唐世渭  
程时端

史美林  
吴时霖  
杨冬青  
周伯生  
范 明  
袁崇义  
谢希仁

TO MY WIFE, BOGNA

# Preface

The study of data structures, a fundamental component of a computer science education, serves as the foundation upon which many other computer science fields are built. Some knowledge of data structures is a must for students who wish to do work in design, implementation, testing, or maintenance of virtually any software system. The scope and presentation of material in *Data Structures and Algorithms in Java* provide students with the knowledge necessary to perform such work.

This book highlights three important aspects of data structures. First, a very strong emphasis is placed on the connection between data structures and their algorithms, including analyzing algorithms' complexity. Second, data structures are presented in an object-oriented setting in accordance with the current design and implementation paradigm. In particular, the information-hiding principle to advance encapsulation and decomposition is stressed. Finally, an important component of the book is data structure implementation, which leads to the choice of Java as the programming language.

The Java language, an object-oriented descendant of C and C++, has gained popularity in industry and academia as an excellent programming language due to widespread use of the Internet. Because of its consistent use of object-oriented features and the security of the language, Java is also useful and natural for introducing data structures. Currently, C++ is the primary language of choice for teaching data structures; however, because of the wide use of Java in application programming and the object-oriented characteristics of the language, using Java to teach a data structures and algorithms course, even on the introductory level, is well justified.

This book provides the material for a course that includes the topics listed under CS2 and CS7 of the old ACM curriculum. It also meets the requirements for most of the courses  $C_A$  202,  $C_D$  202, and  $C_F$  204 of the new ACM curriculum.

Most chapters include a case study that illustrates a complete context in which certain algorithms and data structures can be used. These case studies were chosen from different areas of computer science such as interpreters, symbolic computation, and file processing, to indicate the wide range of applications to which topics under discussion may apply.



Brief examples of Java code are included throughout the book to illustrate the practical importance of data structures. However, theoretical analysis is equally important. Thus, presentations of algorithms are integrated with analyses of efficiency.

Great care is taken in the presentation of recursion because even advanced students have problems with it. Experience has shown that recursion can be explained best if the run-time stack is taken into consideration. Changes to the stack are shown when tracing a recursive function not only in the chapter on recursion, but also in other chapters. For example, a surprisingly short method for tree traversal may remain a mystery if work done by the system on the run-time stack is not included in the explanation. Standing aloof from the system and retaining only a purely theoretical perspective when discussing data structures and algorithms are not necessarily helpful. This book also includes comprehensive chapters on data compression and memory management.

The thrust of this book is data structures, and other topics are treated here only as much as necessary to ensure a proper understanding of this subject. Algorithms are discussed from the perspective of data structures, so the reader will not find a comprehensive discussion of different kinds of algorithms and all the facets that a full presentation of algorithms requires. However, as mentioned, recursion is covered in depth. In addition, complexity analysis of algorithms is presented in some detail.

Chapters 1 and 3–8 present a number of different data structures and the algorithms that operate on them. The efficiency of each algorithm is analyzed, and improvements to the algorithm are suggested.

- Chapter 1 presents the basic principles of object-oriented programming, an introduction to dynamic memory allocation and the use of pointers, and a rudimentary introduction to Java.
- Chapter 2 describes some methods used to assess the efficiency of algorithms.
- Chapter 3 contains an introduction to linked lists.
- Chapter 4 presents stacks and queues and their applications.
- Chapter 5 contains a detailed discussion of recursion. Different types of recursion are discussed, and a recursive call is dissected.
- Chapter 6 discusses binary trees, including implementation, traversal, and search. This chapter also includes balanced trees.
- Chapter 7 details more generalized trees such as tries, 2–4 trees, and B-trees.
- Chapter 8 presents graphs.

Chapters 9–12 show different applications of data structures introduced in the previous chapters. They emphasize the data structure aspects of each topic under consideration.

- Chapter 9 analyzes sorting in detail, and several elementary and nonelementary methods are presented.

- Chapter 10 discusses hashing, one of the most important areas in searching. Various techniques are presented with an emphasis on the utilization of data structures.
- Chapter 11 discusses data compression algorithms and data structures.
- Chapter 12 presents various techniques and data structures for memory management.
- Chapter 13 discusses many algorithms for exact and approximate string matching.
- Appendix A discusses in greater detail big-O notation, introduced in Chapter 2.
- Appendix B gives a proof of Cook's theorem and illustrates it with an extended example.

Each chapter contains a discussion of the material illustrated with appropriate diagrams and tables. Except for Chapter 2, all chapters include a case study, which is an extended example using the features discussed in that chapter. All case studies have been tested using the Visual C++ compiler on a PC and the g++ compiler under UNIX except the von Koch snowflake, which runs on a PC under Visual C++. At the end of each chapter is a set of exercises of varying degrees of difficulty. Except for Chapter 2, all chapters also include programming assignments and an up-to-date bibliography of relevant literature.

Chapters 1–6 (excluding Sections 2.9, 3.4, 6.4.3, 6.7, and 6.8) contain the core material that forms the basis of any data structures course. These chapters should be studied in sequence. The remaining six chapters can be read in any order. A one-semester course could include Chapters 1–6, 9, and Sections 10.1 and 10.2. The entire book could also be part of a two-semester sequence.

## TEACHING TOOLS

**Electronic Instructor's Manual.** The Instructor's Manual that accompanies this text-book includes complete solutions to all text exercises.

**Electronic Figure Files.** All images from the text are available in bitmap format for use in classroom presentations.

**Source Code.** The source code for the text example programs is available via the author's Web site at <http://www.mathes.dug.edu/drozdek/DSinJava>.

It is also available for student download at [course.com](http://course.com). All teaching tools, outlined above, are available in the Instructor's Resources section of [course.com](http://course.com).

## CHANGES IN THE SECOND EDITION

The new edition primarily extends the old edition by including material on new topics that are currently not covered. The additions include

- Pattern matching algorithms in the new Chapter 13
- A discussion of NP-completeness in the form of a general introduction (Section 2.10), examples of NP-complete problems (Section 8.12), and an outline of Cook's theorem (Appendix B)
- New material on graphs (Sections 8.9.1, 8.10.1.1, 8.10.2.1, and 8.11)
- A discussion of a deletion algorithm for vh-trees (Section 7.1.7)
- An introduction to Java files (Sections 1.3.1–1.3.6)

Moreover, the tables that list methods from `java.util` packages have been updated. There are also many small modifications and additions throughout the book.

## ACKNOWLEDGMENTS

I would like to thank the following reviewers, whose comments and advice helped me to improve this book:

James Ball, Indiana State University

Robin Dawes, Queen's University

Julius Dichter, University of Bridgeport

However, the ultimate content is my responsibility, and I would appreciate hearing from readers about any shortcomings or strengths. My email address is `drozdek@duq.edu`.

*Adam Drozdek*

# Contents

<b>1</b>	<b>OBJECT-ORIENTED PROGRAMMING USING JAVA</b>	<b>1</b>
1.1	Rudimentary Java	1
1.1.1	Variable Declarations	1
1.1.2	Operators	4
1.1.3	Decision Statements	5
1.1.4	Loops	6
1.1.5	Exception Handling	6
1.2	Object-Oriented Programming in Java	8
1.2.1	Encapsulation	8
1.2.2	Abstract Data Types	16
1.2.3	Inheritance	18
1.2.4	Polymorphism	21
1.3	Input and Output	24
1.3.1	Reading and Writing Bytes	26
1.3.2	Reading Lines	27
1.3.3	Reading Tokens: Words and Numbers	28
1.3.4	Reading and Writing Primitive Data Types	29
1.3.5	Reading and Writing Objects	29
1.3.6	Random Access File	30
1.4	Java and Pointers	31
1.5	Vectors in <code>java.util</code>	35
1.6	Data Structures and Object-Oriented Programming	42
1.7	Case Study: Random Access File	42
1.8	Exercises	51
1.9	Programming Assignments	53
	Bibliography	55

<b>2</b>	<b>COMPLEXITY ANALYSIS</b>	<b>56</b>
2.1	Computational and Asymptotic Complexity	56
2.2	Big-O Notation	57
2.3	Properties of Big-O Notation	59
2.4	$\Omega$ and $\Theta$ Notations	61
2.5	Possible Problems	62
2.6	Examples of Complexities	62
2.7	Finding Asymptotic Complexity: Examples	64
2.8	The Best, Average, and Worst Cases	66
2.9	Amortized Complexity	69
2.10	NP-Completeness	73
2.11	Exercises	76
	Bibliography	79
<b>3</b>	<b>LINKED LISTS</b>	<b>80</b>
3.1	Singly Linked Lists	80
3.1.1	Insertion	86
3.1.2	Deletion	88
3.1.3	Search	93
3.2	Doubly Linked Lists	95
3.3	Circular Lists	99
3.4	Skip Lists	101
3.5	Self-Organizing Lists	107
3.6	Sparse Tables	111
3.7	Lists in <code>java.util</code>	114
3.7.1	<code>LinkedList</code>	114
3.7.2	<code>ArrayList</code>	120
3.8	Concluding Remarks	123
3.9	Case Study: A Library	124
3.10	Exercises	134
3.11	Programming Assignments	136
	Bibliography	139

<b>4</b>	<b>STACKS AND QUEUES</b>	<b>140</b>
4.1	Stacks	140
4.1.1	Stacks in <code>java.util</code>	148
4.2	Queues	149
4.3	Priority Queues	157
4.4	Case Study: Exiting a Maze	158
4.5	Exercises	164
4.6	Programming Assignments	166
	Bibliography	168
<b>5</b>	<b>RECURSION</b>	<b>169</b>
5.1	Recursive Definitions	169
5.2	Method Calls and Recursion Implementation	172
5.3	Anatomy of a Recursive Call	174
5.4	Tail Recursion	178
5.5	Nontail Recursion	179
5.6	Indirect Recursion	185
5.7	Nested Recursion	187
5.8	Excessive Recursion	188
5.9	Backtracking	191
5.10	Concluding Remarks	198
5.11	Case Study: A Recursive Descent Interpreter	199
5.12	Exercises	207
5.13	Programming Assignments	210
	Bibliography	212
<b>6</b>	<b>BINARY TREES</b>	<b>214</b>
6.1	Trees, Binary Trees, and Binary Search Trees	214
6.2	Implementing Binary Trees	219
6.3	Searching a Binary Search Tree	221
6.4	Tree Traversal	223
6.4.1	Breadth-First Traversal	224
6.4.2	Depth-First Traversal	225
6.4.3	Stackless Depth-First Traversal	231

6.5	Insertion	239
6.6	Deletion	242
6.6.1	Deletion by Merging	243
6.6.2	Deletion by Copying	246
6.7	Balancing a Tree	249
6.7.1	The DSW Algorithm	252
6.7.2	AVL Trees	255
6.8	Self-Adjusting Trees	260
6.8.1	Self-Restructuring Trees	261
6.8.2	Splaying	262
6.9	Heaps	267
6.9.1	Heaps as Priority Queues	269
6.9.2	Organizing Arrays as Heaps	272
6.10	Polish Notation and Expression Trees	275
6.10.1	Operations on Expression Trees	277
6.11	Case Study: Computing Word Frequencies	280
6.12	Exercises	289
6.13	Programming Assignments	292
	Bibliography	296

7

MULTIWAY TREES

299

7.1	The Family of B-Trees	300
7.1.1	B-Trees	301
7.1.2	B*-Trees	312
7.1.3	B <sup>+</sup> -Trees	313
7.1.4	Prefix B <sup>+</sup> -Trees	316
7.1.5	Bit-Trees	319
7.1.6	R-Trees	320
7.1.7	2-4 Trees	323
7.1.8	Trees in <code>java.util</code>	338
7.2	Tries	349
7.3	Concluding Remarks	358
7.4	Case Study: Spell Checker	358
7.5	Exercises	369
7.6	Programming Assignments	370
	Bibliography	374

# 8 GRAPHS 376

- 8.1 Graph Representation 377
- 8.2 Graph Traversals 379
- 8.3 Shortest Paths 383
  - 8.3.1 All-to-All Shortest Path Problem 390
- 8.4 Cycle Detection 392
  - 8.4.1 Union-Find Problem 393
- 8.5 Spanning Trees 395
- 8.6 Connectivity 399
  - 8.6.1 Connectivity in Undirected Graphs 399
  - 8.6.2 Connectivity in Directed Graphs 402
- 8.7 Topological Sort 405
- 8.8 Networks 407
  - 8.8.1 Maximum Flows 407
  - 8.8.2 Maximum Flows of Minimum Cost 417
- 8.9 Matching 421
  - 8.9.1 Stable Matching Problem 426
  - 8.9.2 Assignment Problem 428
  - 8.9.3 Matching in Nonbipartite Graphs 430
- 8.10 Eulerian and Hamiltonian Graphs 432
  - 8.10.1 Eulerian Graphs 432
  - 8.10.2 Hamiltonian Graphs 436
- 8.11 Graph Coloring 442
- 8.12 NP-Complete Problems in Graph Theory 445
  - 8.12.1 The Clique Problem 445
  - 8.12.2 The 3-Colorability Problem 446
  - 8.12.3 The Vertex Cover Problem 448
  - 8.12.4 The Hamiltonian Cycle Problem 449
- 8.13 Case Study: Distinct Representatives 450
- 8.14 Exercises 460
- 8.15 Programming Assignments 464
  - Bibliography 466



## 9 SORTING 469

- 9.1 Elementary Sorting Algorithms 470
  - 9.1.1 Insertion Sort 470
  - 9.1.2 Selection Sort 474
  - 9.1.3 Bubble Sort 475
- 9.2 Decision Trees 477
- 9.3 Efficient Sorting Algorithms 481
  - 9.3.1 Shell Sort 481
  - 9.3.2 Heap Sort 484
  - 9.3.3 Quicksort 488
  - 9.3.4 Mergesort 494
  - 9.3.5 Radix Sort 497
- 9.4 Sorting in `java.util` 502
- 9.5 Concluding Remarks 505
- 9.6 Case Study: Adding Polynomials 507
- 9.7 Exercises 515
- 9.8 Programming Assignments 516
  - Bibliography 517

## 10 HASHING 519

- 10.1 Hash Functions 520
  - 10.1.1 Division 520
  - 10.1.2 Folding 520
  - 10.1.3 Mid-Square Function 521
  - 10.1.4 Extraction 521
  - 10.1.5 Radix Transformation 522
- 10.2 Collision Resolution 522
  - 10.2.1 Open Addressing 522
  - 10.2.2 Chaining 528
  - 10.2.3 Bucket Addressing 530
- 10.3 Deletion 531
- 10.4 Perfect Hash Functions 532
  - 10.4.1 Cichelli's Method 533
  - 10.4.2 The FHCD Algorithm 536
- 10.5 Hash Functions for Extendible Files 538
  - 10.5.1 Extendible Hashing 539
  - 10.5.2 Linear Hashing 541