- Introduction
- What's New in the Second Edition
- What This Book Is About
- What's Different About This Book
- Who This Book Is For
- What You Need to Know Before You Read This Book
- The Software You Need to Use This Book
- How This Book Is Organized
- Enjoy Yourself!
- 1 Overview
- What Are Data Structures and Algorithms Good For? Overview of Data Structures
- Overview of Algorithms
- Some Definitions
- Object-Oriented Programming
- Software Engineering
- Java for C++ Programmers
- Java Library Data Structures
- Summary
- Questions
- 2 Arrays
- The Array Workshop Applet
- The Basics of Arrays in Java
- Dividing a Program into Classes
- Class Interfaces
- The Ordered Workshop Applet
- Java Code for an Ordered Array
- Logarithms
- Storing Objects
- Big O Notation
- Whay Not Use Arrays for Everything
- Summary
- Questions
- Experiments
- Programming Projects
- 3 Simple Sorting
- How Would You Do It
- Bubble Sort
- Selection Sort
- Insertion Sort
- Sorting Objects
- Comparing the Simple Sorts
- Summary
- Questions
- Experiments
- Programming Projects
- 4 Stacks and Queues

- A Different Kind of Structure
- Stacks
- Queues
- Priority Queues
- Parsing Arithmetic Expressions
- Summary
- Questions
- Experiments
- Programming Projects
- 5 Linked Lists
- Links
- The Link List Workshop Applet
- A Simple Linked List
- Finding and Deleting Specified Links
- Double-Ended Lists
- Linked-List Efficiency
- Abstract Data Types
- Sorted Lists
- Doubly Linked Lists
- Iterators
- Summary
- Questions
- Experiments
- Programming Projects
- 6 Recursion
- Triangular Numbers
- Factorials
- Anagrams
- A Recursive Binary Search
- The Towers of Hanoi
- Mergesort
- Eliminating Recursion
- Some Interesting Recursive Applications
- Summary
- Questions
- Experiments
- Programming Projects
- 7 Advanced Sorting
- Shellsort
- Paartitioning
- Quicksort
- Radix Sort
- Summary
- Questions
- Experiments
- Programming Projects

- 8 Binary Trees
- Why Use Binary Trees? Tree Terminology
- An Analogy
- How Do Binary Search Trees Work
- Finding a Node
- Inserting a Node
- Traversing the Tree
- Finding Maximum and Minimum Values
- Deleting a Node
- The Efficiency of Binary Trees
- Trees Represented as Arrays
- Duplicate Keys
- The Complete tree
- Java Program
- The Huffman Code
- Summary
- Questions
- Experiments
- Programming Projects
- 9 Red-Black Trees
- Our Approach to the Discussion
- Balanced and Unbalanced Trees
- Using the RBTree Workshop Applet
- Experimenting with the Workshop Applet
- Rotations
- Inserting a New Node
- Deletion
- The Efficiency of Red-Black Trees
- Red-Black Tree Implementation
- Other Balanced Trees
- Summary
- Questions
- Experiments
- 10 2-3-4 Trees and External Storage
- Introduction to 2-3-4 Trees
- The Tree234 Workshop Applet
- Java Code for a 2-3-4 Tree
- 2-3-4 Trees and Red-Black Trees
- Efficiency of 2-3-4 Trees
- 2-3 Trees
- External Storage
- Summary
- Questions
- Experiments
- Programming Projects
- 11 Hash Tables

- Introduction to Hashing
- Open Addressing
- Separate Chaining
- Hash Functions
- Hashing Efficiency
- Hashing and External Storage
- Summary
- Questions
- Experiments
- Programming Projects
- 12 Heaps
- Introduction to Heaps
- The Heap Workshop Applet
- Java Code fo Heaps
- A Tree-based Heap
- Heapsort
- Summary
- Questions
- Experiments
- Programming Projects
- 13 Graphs
- Introduction to Graphs
- Searches
- Minimum Spanning Trees
- Topological Sorting with Directed Graphs
- Connectivity in Directed Graphs
- Summary
- Questions
- Experiments
- Programming Projects
- 14 Weighted Graphs
- Minimum Spanning Tree with Weighted Graphs
- The Shortest-Path Problem
- The All-Pairs Shortest-Path Problem
- Efficien