Contents

New to the Third Edition 17
Preface 19

1 Introduction 27
1.1 What Is an Algorithm? 29
Exercises 1.1 33
1.2 Fundamentals of Algorithmic Problem Solving 35
Understanding the Problem 35
Ascertaining the Capabilities of the Computational Device 35
Choosing between Exact and Approximate Problem Solving 37
Algorithm Design Techniques 37
Designing an Algorithm and Data Structures 38
Methods of Specifying an Algorithm 38
Proving an Algorithm’s Correctness 39
Analyzing an Algorithm 40
Coding an Algorithm 41
Exercises 1.2 43
1.3 Important Problem Types 44
Sorting 45
Searching 46
String Processing 46
Graph Problems 47
Combinatorial Problems 47
Geometric Problems 48
Numerical Problems 48
Exercises 1.3 49
1.4 Fundamental Data Structures 51
  Linear Data Structures 51
  Graphs 54
  Trees 57
  Sets and Dictionaries 61
  Exercises 1.4 63
  Summary 64

2 Fundamentals of the Analysis of Algorithm Efficiency 67
  2.1 The Analysis Framework 68
    Measuring an Input's Size 69
    Units for Measuring Running Time 70
    Orders of Growth 71
    Worst-Case, Best-Case, and Average-Case Efficiencies 73
    Recapitulation of the Analysis Framework 76
    Exercises 2.1 76
  2.2 Asymptotic Notations and Basic Efficiency Classes 78
    Informal Introduction 78
    O-notation 79
    Ω-notation 80
    Θ-notation 81
    Useful Property Involving the Asymptotic Notations 81
    Using Limits for Comparing Orders of Growth 82
    Basic Efficiency Classes 84
    Exercises 2.2 84
  2.3 Mathematical Analysis of Nonrecursive Algorithms 87
    Exercises 2.3 93
  2.4 Mathematical Analysis of Recursive Algorithms 96
    Exercises 2.4 102
  2.5 Example: Computing the nth Fibonacci Number 106
    Exercises 2.5 109
  2.6 Empirical Analysis of Algorithms 110
    Exercises 2.6 115
  2.7 Algorithm Visualization 117
    Summary 120
3 Brute Force and Exhaustive Search

3.1 Selection Sort and Bubble Sort
   Selection Sort 124
   Bubble Sort 126
   Exercises 3.1 128

3.2 Sequential Search and Brute-Force String Matching
   Sequential Search 130
   Brute-Force String Matching 131
   Exercises 3.2 132

3.3 Closest-Pair and Convex-Hull Problems by Brute Force
   Closest-Pair Problem 134
   Convex-Hull Problem 135
   Exercises 3.3 139

3.4 Exhaustive Search
   Traveling Salesman Problem 142
   Knapsack Problem 142
   Assignment Problem 145
   Exercises 3.4 146

3.5 Depth-First Search and Breadth-First Search
   Depth-First Search 148
   Breadth-First Search 151
   Exercises 3.5 154
   Summary 156

4 Decrease-and-Conquer

4.1 Insertion Sort
   Exercises 4.1 162

4.2 Topological Sorting
   Exercises 4.2 168

4.3 Algorithms for Generating Combinatorial Objects
   Generating Permutations 170
   Generating Subsets 172
   Exercises 4.3 174
4.4 Decrease-by-a-Constant-Factor Algorithms
   Binary Search 176
   Fake-Coin Problem 178
   Russian Peasant Multiplication 179
   Josephus Problem 180
   Exercises 4.4 182

4.5 Variable-Size-Decrease Algorithms
   Computing a Median and the Selection Problem 184
   Interpolation Search 187
   Searching and Insertion in a Binary Search Tree 189
   The Game of Nim 190
   Exercises 4.5 192
   Summary 193

5 Divide-and-Conquer 195

5.1 Mergesort 198
   Exercises 5.1 200

5.2 Quicksort 202
   Exercises 5.2 207

5.3 Binary Tree Traversals and Related Properties 208
   Exercises 5.3 211

5.4 Multiplication of Large Integers and
   Strassen’s Matrix Multiplication 212
   Multiplication of Large Integers 213
   Strassen’s Matrix Multiplication 215
   Exercises 5.4 217

5.5 The Closest-Pair and Convex-Hull Problems
   by Divide-and-Conquer 218
   The Closest-Pair Problem 218
   Convex-Hull Problem 221
   Exercises 5.5 223
   Summary 224
6 Transform-and-Conquer

6.1 Presorting
Exercises 6.1

6.2 Gaussian Elimination
LU Decomposition
Computing a Matrix Inverse
Computing a Determinant
Exercises 6.2

6.3 Balanced Search Trees
AVL Trees
2-3 Trees
Exercises 6.3

6.4 Heaps and Heapsort
Notion of the Heap
Heapsort
Exercises 6.4

6.5 Horner's Rule and Binary Exponentiation
Horner's Rule
Binary Exponentiation
Exercises 6.5

6.6 Problem Reduction
Computing the Least Common Multiple
Counting Paths in a Graph
Reduction of Optimization Problems
Linear Programming
Reduction to Graph Problems
Exercises 6.6
Summary

7 Space and Time Trade-Offs

7.1 Sorting by Counting
Exercises 7.1

7.2 Input Enhancement in String Matching
Horspool's Algorithm
Boyer-Moore Algorithm 289
Exercises 7.2 293

7.3 Hashing 295
Open Hashing (Separate Chaining) 296
Closed Hashing (Open Addressing) 298
Exercises 7.3 300

7.4 B-Trees 302
Exercises 7.4 305
Summary 306

8 Dynamic Programming 309

8.1 Three Basic Examples 311
Exercises 8.1 316

8.2 The Knapsack Problem and Memory Functions 318
Memory Functions 320
Exercises 8.2 322

8.3 Optimal Binary Search Trees 323
Exercises 8.3 329

8.4 Warshall’s and Floyd’s Algorithms 330
Warshall’s Algorithm 330
Floyd’s Algorithm for the All-Pairs Shortest-Paths Problem 334
Exercises 8.4 337
Summary 338

9 Greedy Technique 341

9.1 Prim’s Algorithm 344
Exercises 9.1 348

9.2 Kruskal’s Algorithm 351
Disjoint Subsets and Union-Find Algorithms 353
Exercises 9.2 357

9.3 Dijkstra’s Algorithm 359
Exercises 9.3 363
9.4 Huffman Trees and Codes
   Exercises 9.4
   Summary

10 Iterative Improvement

10.1 The Simplex Method
   Geometric Interpretation of Linear Programming
   An Outline of the Simplex Method
   Further Notes on the Simplex Method
   Exercises 10.1

10.2 The Maximum-Flow Problem
   Exercises 10.2

10.3 Maximum Matching in Bipartite Graphs
   Exercises 10.3

10.4 The Stable Marriage Problem
   Exercises 10.4
   Summary

11 Limitations of Algorithm Power

11.1 Lower-Bound Arguments
   Trivial Lower Bounds
   Information-Theoretic Arguments
   Adversary Arguments
   Problem Reduction
   Exercises 11.1

11.2 Decision Trees
   Decision Trees for Sorting
   Decision Trees for Searching a Sorted Array
   Exercises 11.2

11.3 P, NP, and NP-Complete Problems
   P and NP Problems
   NP-Complete Problems
   Exercises 11.3
11.4 Challenges of Numerical Algorithms 438
   Exercises 11.4 445
   Summary 446

12 Coping with the Limitations of Algorithm Power 449

12.1 Backtracking 450
   n-Queens Problem 451
   Hamiltonian Circuit Problem 452
   Subset-Sum Problem 453
   General Remarks 454
   Exercises 12.1 456

12.2 Branch-and-Bound 458
   Assignment Problem 459
   Knapsack Problem 462
   Traveling Salesman Problem 464
   Exercises 12.2 466

12.3 Approximation Algorithms for \(NP\)-Hard Problems 467
   Approximation Algorithms for the Traveling Salesman Problem 469
   Approximation Algorithms for the Knapsack Problem 479
   Exercises 12.3 483

12.4 Algorithms for Solving Nonlinear Equations 485
   Bisection Method 486
   Method of False Position 490
   Newton's Method 490
   Exercises 12.4 493
   Summary 494

Epilogue 497

APPENDIX A

Useful Formulas for the Analysis of Algorithms 501
   Properties of Logarithms 501
   Combinatorics 501
   Important Summation Formulas 502
   Sum Manipulation Rules 502
APPENDIX B

Short Tutorial on Recurrence Relations 505
Sequences and Recurrence Relations 505
Methods for Solving Recurrence Relations 506
Common Recurrence Types in Algorithm Analysis 511

References 519

Hints to Exercises 529

Index 571