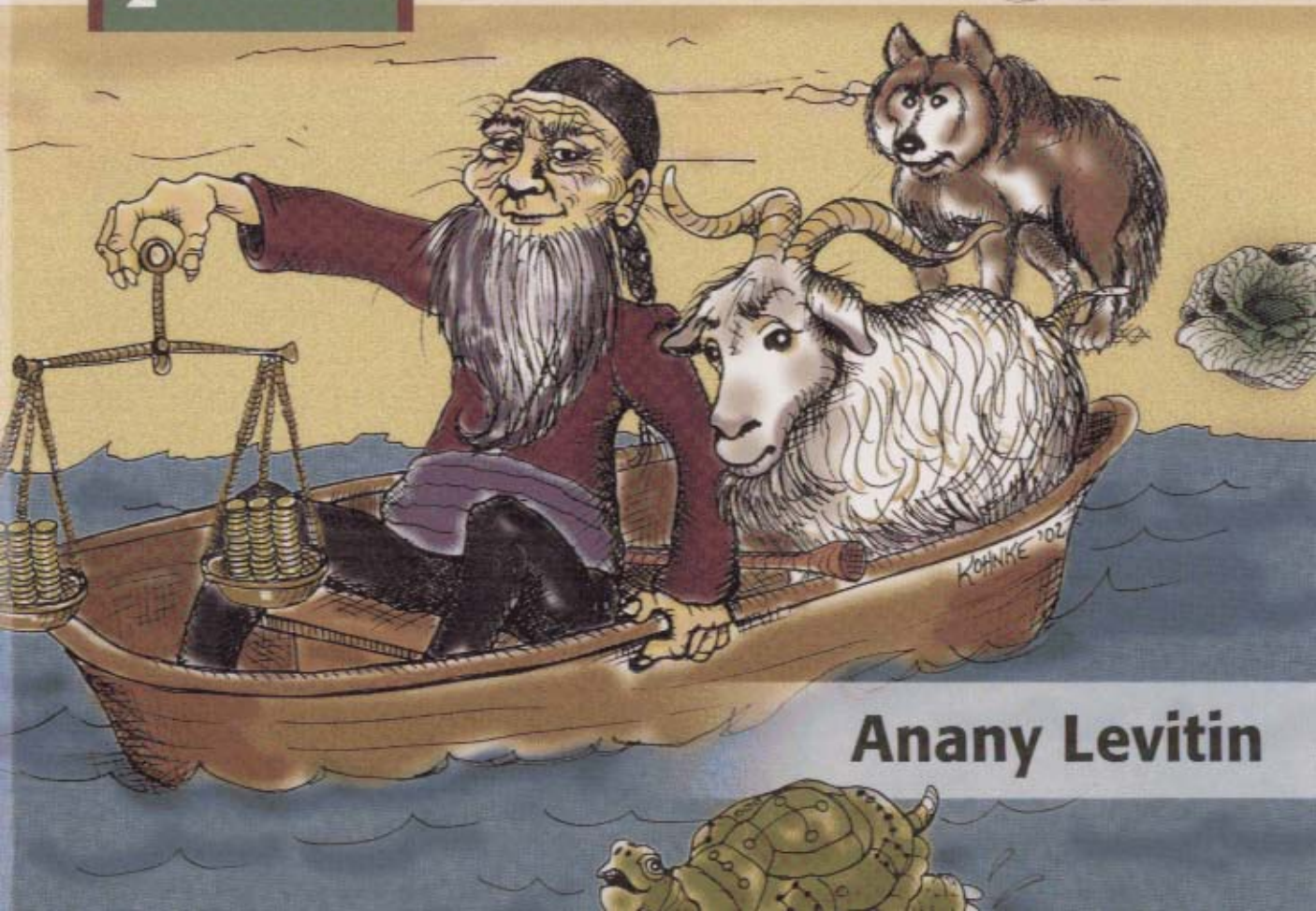
Introduction to The Design & Analysis of Algorithms

2ND EDITION



Anany Levitin

Contents

Preface xvii

1 Introduction 1

1.1 What is an Algorithm? 3
Exercises 1.1 8

1.2 Fundamentals of Algorithmic Problem Solving 9
Understanding the Problem 9
Ascertaining the Capabilities of a Computational Device 11
Choosing between Exact and Approximate Problem Solving 11
Deciding on Appropriate Data Structures 12
Algorithm Design Techniques 12
Methods of Specifying an Algorithm 12
Proving an Algorithm's Correctness 13
Analyzing an Algorithm 14
Coding an Algorithm 15
Exercises 1.2 17

1.3 Important Problem Types 19
Sorting 19
Searching 20
String Processing 21
Graph Problems 21
Combinatorial Problems 22
Geometric Problems 22
Numerical Problems 23
Exercises 1.3 23

1.4 Fundamental Data Structures 26
Linear Data Structures 26
Graphs 28
Trees 32
Sets and Dictionaries 36
Exercises 1.4 38
Summary 39

2 Fundamentals of the Analysis of Algorithm Efficiency 41

2.1 Analysis Framework 42
Measuring an Input's Size 43
Units for Measuring Running Time 44
Orders of Growth 45
Worst-Case, Best-Case, and Average-Case Efficiencies 47
Recapitulation of the Analysis Framework 50
Exercises 2.1 50

2.2 Asymptotic Notations and Basic Efficiency Classes 52
Informal Introduction 52
O-notation 53
Ω-notation 54
Θ-notation 55
Useful Property Involving the Asymptotic Notations 56
Using Limits for Comparing Orders of Growth 57
Basic Efficiency Classes 58
Exercises 2.2 59

2.3 Mathematical Analysis of Nonrecursive Algorithms 61
Exercises 2.3 67

2.4 Mathematical Analysis of Recursive Algorithms 69
Exercises 2.4 76

2.5 Example: Fibonacci Numbers 78
Explicit Formula for the nth Fibonacci Number 79
Algorithms for Computing Fibonacci Numbers 80
Exercises 2.5 83

2.6 Empirical Analysis of Algorithms 84
Exercises 2.6 90

2.7 Algorithm Visualization 91
Summary 95

3 Brute Force 97

3.1 Selection Sort and Bubble Sort 98
Selection Sort 98
Bubble Sort 100
Exercises 3.1 102

3.2 Sequential Search and Brute-Force String Matching 103
Sequential Search 103
Brute-Force String Matching 104
Exercises 3.2 105

3.3 Closest-Pair and Convex-Hull Problems by Brute Force 107
Closest-Pair Problem 107
Convex-Hull Problem 108
Exercises 3.3 112

3.4 Exhaustive Search 114
Traveling Salesman Problem 114
Knapsack Problem 115
Assignment Problem 116
Exercises 3.4 119
Summary 120

4 Divide-and-Conquer 123

4.1 Mergesort 125
Exercises 4.1 128

4.2 Quicksort 129
Exercises 4.2 134

4.3 Binary Search 135
Exercises 4.3 138

4.4 Binary Tree Traversals and Related Properties 139
Exercises 4.4 142

4.5 Multiplication of Large Integers and Strassen's Matrix Multiplication 144
Multiplication of Large Integers 144
Strassen's Matrix Multiplication 146
Exercises 4.5 148

4.6 Closest-Pair and Convex-Hull Problems by Divide-and-Conquer 149
Closest-Pair Problem 149
Convex-Hull Problem 150
Exercises 4.6 154
Summary 155

5 Decrease-and-Conquer 157

5.1 Insertion Sort 160
Exercises 5.1 163

5.2 Depth-First Search and Breadth-First Search 164
Depth-First Search 165
Breadth-First Search 167
Exercises 5.2 170

5.3 Topological Sorting 172
Exercises 5.3 176

5.4 Algorithms for Generating Combinatorial Objects 177
Generating Permutations 178
Generating Subsets 180
Exercises 5.4 181

5.5 Decrease-by-a-Constant-Factor Algorithms 183
Fake-Coin Problem 183
Multiplication à la Russe 184
Josephus Problem 185
Exercises 5.5 187

5.6 Variable-Size-Decrease Algorithms 188
Computing a Median and the Selection Problem 188

Interpolation Search 190
Searching and Insertion in a Binary Search Tree 191
The Game of Nim 192
Exercises 5.6 194
Summary 195

6 Transform-and-Conquer 197

6.1 Presorting 198
Exercises 6.1 201

6.2 Gaussian Elimination 203
LU Decomposition and Other Applications 208
Computing a Matrix Inverse 209
Computing a Determinant 210
Exercises 6.2 212

6.3 Balanced Search Trees 214
AVL Trees 214
2-3 Trees 218
Exercises 6.3 222

6.4 Heaps and Heapsort 223
Notion of the Heap 223
Heapsort 227
Exercises 6.4 229

6.5 Horner's Rule and Binary Exponentiation 230
Horner's Rule 231
Binary Exponentiation 233
Exercises 6.5 236

6.6 Problem Reduction 237
Computing the Least Common Multiple 238
Counting Paths in a Graph 239
Reduction of Optimization Problems 240
Linear Programming 241
Reduction to Graph Problems 244
Exercises 6.6 245
Summary 247

7 Space and Time Tradeoffs 249

7.1 *Sorting by Counting* 250
Exercises 7.1 253

7.2 Input Enhancement in String Matching 255
Horspool's Algorithm 255
Boyer-Moore Algorithm 259
Exercises 7.2 264

7.3 Hashing 265
Open Hashing (Separate Chaining) 266
Closed Hashing (Open Addressing) 268
Exercises 7.3 270

7.4 *B-Trees* 272
Exercises 7.4 275
Summary 276

8 Dynamic Programming 279

8.1 Computing a Binomial Coefficient 280
Exercises 8.1 282

8.2 Warshall's and Floyd's Algorithms 284
Warshall's Algorithm 284
Floyd's Algorithm for the All-Pairs Shortest-Paths Problem 288
Exercises 8.2 292

8.3 Optimal Binary Search Trees 293
Exercises 8.3 298

8.4 The Knapsack Problem and Memory Functions 299
Memory Functions 301
Exercises 8.4 303
Summary 304

9 Greedy Technique 307

9.1 Prim's Algorithm 308
Exercises 9.1 313

9.2 Kruskal's Algorithm 315
Disjoint Subsets and Union-Find Algorithms 317
Exercises 9.2 321

9.3 Dijkstra's Algorithm 323
Exercises 9.3 327

9.4 Huffman Trees 328
Exercises 9.4 332
Summary 333

10 Iterative Improvement 335

10.1 The Simplex Method 336
Geometric Interpretation of Linear Programming 337
An Outline of the Simplex Method 341
Further Notes on the Simplex Method 347
Exercises 10.1 349

10.2 The Maximum-Flow Problem 351
Exercises 10.2 361

10.3 Maximum Matching in Bipartite Graphs 363
Exercises 10.3 369

10.4 The Stable Marriage Problem 371
Exercises 10.4 375
Summary 376

11 Limitations of Algorithm Power 379

11.1 Lower-Bound Arguments 380
Trivial Lower Bounds 381
Information-Theoretic Arguments 382

Adversary Arguments 382
Problem Reduction 383
Exercises 11.1 385

11.2 Decision Trees 386
Decision Trees for Sorting Algorithms 387
Decision Trees for Searching a Sorted Array 390
Exercises 11.2 392

11.3 *P*, *NP*, and *NP*-complete Problems 393
P and *NP* Problems 394
NP-Complete Problems 398
Exercises 11.3 401

11.4 Challenges of Numerical Algorithms 404
Exercises 11.4 411
Summary 413

12 Coping with the Limitations of Algorithm Power 415

12.1 Backtracking 416
n-Queens Problem 417
Hamiltonian Circuit Problem 418
Subset-Sum Problem 419
General Remarks 421
Exercises 12.1 422

12.2 Branch-and-Bound 424
Assignment Problem 425
Knapsack Problem 428
Traveling Salesman Problem 430
Exercises 12.2 432

12.3 Approximation Algorithms for *NP*-hard Problems 434
Approximation Algorithms for the Traveling Salesman Problem 436
Approximation Algorithms for the Knapsack Problem 446
Exercises 12.3 451

12.4 Algorithms for Solving Nonlinear Equations 452
Bisection Method 454
Method of False Position 457

Newton's Method 458
Exercises 12.4 461
Summary 462

Epilogue 465

APPENDIX A

Useful Formulas for the Analysis of Algorithms 469
Properties of Logarithms 469
Combinatorics 469
Important Summation Formulas 470
Sum Manipulation Rules 470
Approximation of a Sum by a Definite Integral 471
Floor and Ceiling Formulas 471
Miscellaneous 471

APPENDIX B

Short Tutorial on Recurrence Relations 473
Sequences and Recurrence Relations 473
Methods for Solving Recurrence Relations 474
Common Recurrence Types in Algorithm Analysis 479

Bibliography 487

Hints to Exercises 497

Index 541