Algorithms Illuminated
(Omnibus Edition)

Tim Roughgarden

Soundlikeyourself Publishing
New York, NY

Contents

Preface xii

I The Basics 1

1 Introduction 2
- 1.1 Why Study Algorithms? 2
- 1.2 Integer Multiplication 3
- 1.3 Karatsuba Multiplication 6
- 1.4 `MergeSort`: The Algorithm 10
- 1.5 `MergeSort`: The Analysis 14
- 1.6 Guiding Principles for the Analysis of Algorithms 20

Problems 24

2 Asymptotic Notation 27
- 2.1 The Gist 27
- 2.2 Big-O Notation 33
- 2.3 Two Basic Examples 35
- 2.4 Big-Omega and Big-Theta Notation 37
- 2.5 Additional Examples 40

Problems 42

3 Divide-and-Conquer Algorithms 45
- 3.1 The Divide-and-Conquer Paradigm 45
- 3.2 Counting Inversions in $O(n \log n)$ Time 46
- 3.3 Strassen's Matrix Multiplication Algorithm 53
- *3.4 An $O(n \log n)$-Time Algorithm for the Closest Pair 58

Problems 68

4 The Master Method 71
- 4.1 Integer Multiplication Revisited 71
- 4.2 Formal Statement 73
- 4.3 Six Examples 75
- *4.4 Proof of the Master Method 79

Problems 87

5 QuickSort 90
5.1 Overview 90
5.2 Partitioning Around a Pivot Element 93
5.3 The Importance of Good Pivots 98
5.4 Randomized `QuickSort` 101
*5.5 Analysis of Randomized `QuickSort` 104
*5.6 Sorting Requires $\Omega(n \log n)$ Comparisons 112
Problems 116

6 Linear-Time Selection 120
6.1 The `RSelect` Algorithm 120
*6.2 Analysis of `RSelect` 125
*6.3 The `DSelect` Algorithm 129
*6.4 Analysis of `DSelect` 132
Problems 138

II Graph Algorithms and Data Structures 141

7 Graphs: The Basics 142
7.1 Some Vocabulary 142
7.2 A Few Applications 143
7.3 Measuring the Size of a Graph 144
7.4 Representing a Graph 146
Problems 151

8 Graph Search and Its Applications 153
8.1 Overview 153
8.2 Breadth-First Search and Shortest Paths 159
8.3 Computing Connected Components 166
8.4 Depth-First Search 170
8.5 Topological Sort 174
*8.6 Computing Strongly Connected Components 181
8.7 The Structure of the Web 190
Problems 193

9 Dijkstra's Shortest-Path Algorithm 198
9.1 The Single-Source Shortest Path Problem 198
9.2 Dijkstra's Algorithm 202
*9.3 Why Is Dijkstra's Algorithm Correct? 204
9.4 Implementation and Running Time 209
Problems 210

10 The Heap Data Structure 214
10.1 Data Structures: An Overview 214
10.2 Supported Operations 216
10.3 Applications 218

10.4 Speeding Up Dijkstra's Algorithm 222
*10.5 Implementation Details 226
Problems 235

11 Search Trees **237**
11.1 Sorted Arrays 237
11.2 Search Trees: Supported Operations 239
*11.3 Implementation Details 241
*11.4 Balanced Search Trees 251
Problems 254

12 Hash Tables and Bloom Filters **257**
12.1 Supported Operations 257
12.2 Applications 259
*12.3 Implementation: High-Level Ideas 263
*12.4 Further Implementation Details 273
12.5 Bloom Filters: The Basics 277
*12.6 Bloom Filters: Heuristic Analysis 282
Problems 286

III Greedy Algorithms and Dynamic Programming **290**

13 Introduction to Greedy Algorithms **291**
13.1 The Greedy Algorithm Design Paradigm 291
13.2 A Scheduling Problem 293
13.3 Developing a Greedy Algorithm 295
13.4 Proof of Correctness 299
Problems 303

14 Huffman Codes **306**
14.1 Codes 306
14.2 Codes as Trees 309
14.3 Huffman's Greedy Algorithm 313
*14.4 Proof of Correctness 320
Problems 326

15 Minimum Spanning Trees **328**
15.1 Problem Definition 328
15.2 Prim's Algorithm 331
*15.3 Speeding Up Prim's Algorithm via Heaps 335
*15.4 Prim's Algorithm: Proof of Correctness 340
15.5 Kruskal's Algorithm 346
*15.6 Speeding Up Kruskal's Algorithm via Union-Find 349
*15.7 Kruskal's Algorithm: Proof of Correctness 357
15.8 Application: Single-Link Clustering 358
Problems 362

16 Introduction to Dynamic Programming 366
16.1 The Weighted Independent Set Problem 366
16.2 A Linear-Time Algorithm for WIS in Paths 370
16.3 A Reconstruction Algorithm 376
16.4 The Principles of Dynamic Programming 377
16.5 The Knapsack Problem 381
Problems 388

17 Advanced Dynamic Programming 392
17.1 Sequence Alignment 392
*17.2 Optimal Binary Search Trees 400
Problems 411

18 Shortest Paths Revisited 415
18.1 Shortest Paths with Negative Edge Lengths 415
18.2 The Bellman-Ford Algorithm 418
18.3 The All-Pairs Shortest Path Problem 429
18.4 The Floyd-Warshall Algorithm 430
Problems 438

IV Algorithms for NP-Hard Problems 441

19 What Is NP-Hardness? 442
19.1 MST vs. TSP: An Algorithmic Mystery 442
19.2 Possible Levels of Expertise 446
19.3 Easy and Hard Problems 448
19.4 Algorithmic Strategies for NP-Hard Problems 453
19.5 Proving NP-Hardness: A Simple Recipe 457
19.6 Rookie Mistakes and Acceptable Inaccuracies 464
Problems 467

20 Compromising on Correctness: Efficient Inexact Algorithms 471
20.1 Makespan Minimization 471
20.2 Maximum Coverage 481
*20.3 Influence Maximization 490
20.4 The 2-OPT Heuristic Algorithm for the TSP 497
20.5 Principles of Local Search 502
Problems 511

21 Compromising on Speed: Exact Inefficient Algorithms 519
21.1 The Bellman-Held-Karp Algorithm for the TSP 519
*21.2 Finding Long Paths by Color Coding 525
21.3 Problem-Specific Algorithms vs. Magic Boxes 535
21.4 Mixed Integer Programming Solvers 537
21.5 Satisfiability Solvers 540
Problems 545

22 Proving Problems NP-Hard 551
22.1 Reductions Revisited 551
22.2 3-SAT and the Cook-Levin Theorem 553
22.3 The Big Picture 554
22.4 A Template for Reductions 557
22.5 Independent Set Is NP-Hard 558
*22.6 Directed Hamiltonian Path Is NP-Hard 562
22.7 The TSP Is NP-Hard 567
22.8 Subset Sum Is NP-Hard 569
Problems 574

23 P, NP, and All That 577
*23.1 Amassing Evidence of Intractability 577
*23.2 Decision, Search, and Optimization 579
*23.3 $\mathcal{NP}$: Problems with Easily Recognized Solutions 580
*23.4 The P $\neq$ NP Conjecture 584
*23.5 The Exponential Time Hypothesis 587
*23.6 NP-Completeness 590
Problems 594

24 Case Study: The FCC Incentive Auction 596
24.1 Repurposing Wireless Spectrum 596
24.2 Greedy Heuristics for Buying Back Licenses 598
24.3 Feasibility Checking 604
24.4 Implementation as a Descending Clock Auction 609
24.5 The Final Outcome 613
Problems 615

A Quick Review of Proofs By Induction 617
A.1 A Template for Proofs by Induction 617
A.2 Example: A Closed-Form Formula 618
A.3 Example: The Size of a Complete Binary Tree 618

B Quick Review of Discrete Probability 620
B.1 Sample Spaces 620
B.2 Events 621
B.3 Random Variables 622
B.4 Expectation 623
B.5 Linearity of Expectation 624
B.6 Example: Load Balancing 627

Epilogue: A Field Guide to Algorithm Design 630

Hints and Solutions 632

Index 655