Design and Analysis of Algorithms

SECOND EDITION

S. Sridhar
Professor
Department of Information Science and Technology
College of Engineering, Guindy Campus
Anna University, Chennai

OXFORD
UNIVERSITY PRESS

Detailed Contents

Features of the Book *iv*
Preface *vi*

1 Introduction to Algorithms 1
1.1 Introduction 1
1.2 Need for Algorithmic Thinking 1
1.3 Overview of Problems 3
1.3.1 Computational Problems, Instance and Size 3
1.4 Overview of Algorithms 4
1.5 Need for Algorithm Efficiency 6
1.6 Fundamental Stages of Problem Solving 7
1.6.1 Understanding the Problem 7
1.6.2 Planning an Algorithm 7
1.6.3 Designing an Algorithm 8
1.6.4 Validating and Verifying an Algorithm 9
1.6.5 Analysing an Algorithm 9
1.6.6 Implementing an Algorithm and Performing Empirical Analysis 11
1.6.7 Post (or Postmortem) Analysis 11
1.7 Classification of Algorithms 11
1.7.1 Based on Implementation 11
1.7.2 Based on Design 13
1.7.3 Based on Area of Specialization 13
1.7.4 Based on Tractability 13

2 Basics of Algorithm Writing 17
2.1 Computational Thinking 17
2.1.1 Decomposition 18
2.1.2 Pattern Matching 18
2.1.3 Abstraction 18
2.1.4 Algorithm Design 18
2.2 Design Approaches 19
2.2.1 Bottom-up Approach 20
2.3 Algorithm Specifications 21
2.3.1 Guidelines for Writing Algorithms 22
2.4 Iterative Algorithms 26
2.5 Recursive Algorithms 33
2.6 Algorithm Correctness 39

3 Basics of Algorithm Analysis 44
3.1 Basics of Algorithm Complexity 44
3.2 Introduction to Time Complexity 45
3.3 Analysis of Iterative Algorithms 47
3.3.1 Measuring Input Size 47
3.3.2 Measuring Running Time 48
3.4 Best-, Worst-, and Average-case Complexity 55
3.5 Rate of Growth 56
3.5.1 Measuring Larger Inputs 56
3.6 Asymptotic Analysis 59
3.6.1 Asymptotic Notations 60
3.6.2 Limits and Asymptotic Notations 64
3.6.3 Asymptotic Rules 68
3.6.4 Asymptotic Complexity Classes 69
3.7 Empirical Analysis and Algorithm Visualization 70
3.7.1 Empirical Analysis 70

4 Mathematical Analysis of Recursive Algorithms 76
4.1 Introduction to Recurrence Equations 76
4.1.1 Linear Recurrences 77
4.1.2 Non-linear Recurrences 79
4.2 Formulation of Recurrence Equations 80
4.3 Techniques for Solving Recurrence Equations 82
4.3.1 Guess-and-verify Method 82
4.3.2 Substitution Method 84
4.3.3 Recurrence-tree Method 87
4.3.4 Difference Method 90
4.4 Solving Recurrence Equations Using Polynomial Reduction 91
4.4.1 Solving Homogeneous Equations 91
4.4.2 Solving Non-homogeneous Equations 95
4.5 Divide-and-conquer Recurrences 95
4.5.1 Master Theorem 96
4.6 Transformations 101
4.6.1 Domain Transformation 102

5 Data Structures—I 107
5.1 Data Structures and Algorithms 107
5.2 Lists 108
5.2.1 Linear Lists 108
5.2.2 Linked Lists 111
5.3 Stacks 115
5.3.1 Representation and Operations on Stacks 115
5.4 Queues 117
5.4.1 Queue Representation 117

5.5 Trees 119
5.5.1 Tree Terminology 119
5.5.2 Classification of Trees 121
5.5.3 Binary Tree Representation 122
5.5.4 Binary Tree Operations 123
5.6 Graphs 127
5.6.1 Terminologys and Types of Graphs 127
5.6.2 Graph Representation 130
5.7 Elementary Graph Algorithms 131
5.7.1 Graph Traversal Algorithms 131
5.7.2 Cycle Detection 134
5.7.3 Connected Components 137
5.7.4 Strongly Connected Components 138
5.7.5 Biconnected Components 139
5.7.6 Spanning Tree and Minimum-cost Spanning Tree 141

6 Data Structures—II 147
6.1 Introduction to Binary Search Tree 147
6.2 AVL Trees 152
6.3 Binary Heaps 156
6.4 Binomial Heaps 163
6.5 Fibonacci Heap 168
6.6 Disjoint Sets 170
6.7 Amortized Analysis 173
6.7.1 Aggregate Method 174
6.7.2 Accounting Method 174
6.7.3 Potential Method 175

7 Brute Force Approaches 179
7.1 Introduction 179
7.1.1 Advantages and Disadvantages of Brute Force Method 180
7.2 Linear Search 180
7.2.1 Analysis of Recursion Programs 181
7.3 Sorting Problem 182
7.3.1 Bubble Sort 183
7.3.2 Selection Sort 185
7.4 Computational Geometry Problems 187
7.4.1 Closest-pair Problem 187
7.4.2 Convex Hull Problem 189
7.5 Exhaustive Searching 191
7.5.1 15-puzzle Problem 191
7.5.2 8-queen Problem 193
7.5.3 Magic Squares 193
7.5.4 Container Loading Problem 193
7.5.5 Knapsack Problem 195
7.5.6 Assignment Problem 197

8 Divide-and-conquer Approach 202
8.1 Introduction 202
8.1.1 Recurrence Equation for Divide and Conquer 203
8.1.2 Advantages and Disadvantages of Divide-and-conquer Paradigm 203
8.2 Merge Sort 204
8.3 Quicksort 208
8.3.1 Partitioning Algorithms 209
8.3.2 Variants of Quicksort 213
8.4 Multiplication of Long Integers 214
8.4.1 Divide and Conquer Approach 215
8.5 Strassen Matrix Multiplication 217
8.6 Tiling Problem 221
8.7 Closest-pair Problem 223
8.8 Convex Hull 225
8.8.1 Quickhull 225
8.8.2 Merge Hull 226
8.9 Fourier Transform for Polynomial Multiplication 227
8.9.1 Improvement of Polynomial Multiplication using Value Representation 228
8.9.2 Divide and Conquer Approach and Fourier Transform 228
8.9.3 Fast Discrete Fourier Transform (FFT) using Divide and Conquer Approach 231
8.9.4 Polynomial Multiplication 232

9 Decrease-and-conquer Approach 237
9.1 Introduction 237
9.2 Decrease by Constant Method 238
9.2.1 Insertion Sort 239
9.2.2 Topological Sort 242
9.2.3 Generating Permutations 246
9.2.4 Generating Subsets 248
9.3 Decrease by Constant Factor Method 249
9.3.1 Binary Search 249
9.3.2 Fake Coin Detection 251
9.3.3 Russian Peasant Multiplication Problem 252
9.4 Decrease by Variable Factor Method 253
9.4.1 Interpolation Search 253
9.4.2 Selection and Ordered Statistics 254
9.4.3 Finding Median 257
9.4.4 Selection (Deterministic and Random Selection) 257
9.4.5 Finding Median using Median of Medians 259

10 Time–Space Tradeoffs 264
10.1 Introduction to Time-Space Tradeoffs 264
10.2 Counting Sort 264
10.3 Bucket Sort 268
10.4 Radix Sort 270
10.5 Shell Sort 271
10.6 Hashing and Hash Tables 273
10.6.1 Introduction to Hash Functions 273
10.6.2 Hash Functions Construction 273

10.6.3 Hash Table Operations *275*
10.6.4 Collision Resolution Techniques *276*
10.7 Introduction to *M*-ary Trees 278
10.7.1 B-Tree *279*
10.7.2 B+ Trees *284*

11 Greedy Algorithms 288
11.1 Introduction to the Greedy Approach 288
11.2 Components of Greedy Algorithms 289
11.2.1 Properties of Greedy Algorithms *290*
11.3 Coin Change Problem 291
11.4 Scheduling Problems 292
11.4.1 Shortest Job First Scheduling *292*
11.4.2 Scheduling with Deadline *294*
11.4.3 Activity Selection Problem *296*
11.5 Knapsack Problem 298
11.6 Optimal Storage of Tapes 301
11.7 Optimal Merge Problem 302
11.7.1 Evaluation of Optimal Tree of Merging *303*
11.8 Huffman Coding 304
11.8.1 Efficiency of Huffman Code Using Weighted Tree *306*
11.9 Tree Vertex Splitting Problem 308
11.10 Minimum Spanning Trees 310
11.10.1 Kruskal's Algorithm *310*
11.10.2 Prim's Algorithm *315*
11.10.3 Single-Source Shortest-Path Problems *319*

12 Transform-and-conquer Approach 328
12.1 Introduction to Transform and Conquer 328
12.2 Introduction to Instance Simplification 329
12.3 Gaussian Elimination and Matrix Operations 330
12.4 Applications of Gaussian Elimination 336
12.4.1 LU Decomposition *337*
12.4.2 Solving a set of Linear Equations using LU Decomposition *340*
12.4.3 LUP Decomposition *342*
12.4.4 Solving Linear Equations using LU Decomposition *344*
12.4.5 Finding Matrix Inverse using Gaussian Elimination *344*
12.4.6 Finding Determinant using LU Decomposition *346*
12.5 Change of Representation and Heap Sort 348
12.6 Polynomial Evaluation Using Horner's Method 351
12.7 Binary Exponentiation 353
12.7.1 Left-to-right Binary Exponentiation *355*
12.7.2 Right-to-left Binary Exponentiation *356*
12.8 Problem Reduction 357

13 Dynamic Programming 361
13.1 Basics of Dynamic Programming 361
13.2 Components of Dynamic Programming 363
13.3 Fibonacci Problem 365
13.3.1 Bottom-up Approach *365*
13.3.2 Top-down Approach and Memoization *366*
13.4 Computing Binomial Coefficients 367
13.5 Multistage Graph Problem 370
13.5.1 Forward Computation Procedure *370*
13.5.2 Backward Computation Procedure *374*
13.6 Transitive Closure and Warshall Algorithm 376
13.7 Floyd–Warshall All-Pairs Shortest-Path Algorithm 379
13.8 Bellman-Ford Algorithm 382
13.9 Travelling Salesperson Problem 385
13.10 Chain Matrix Multiplication 387
13.11 Knapsack Problem 394
13.12 Optimal Binary Search Trees 397
13.12.1 Dynamic Programming Approach for Constructing Optimal BSTs *398*
13.13 Flow-Shop Scheduling Problem 402
13.13.1 Single-machine Sequencing Problem *402*
13.13.2 Two-machine Sequencing Problem *403*

14 Backtracking 410
14.1 Introduction 410
14.2 Basics of Backtracking 411
14.2.1 Generation of State-space Trees *412*
14.2.2 Searching State-space Trees *413*
14.2.3 Implementation of Backtracking Algorithm *415*
14.3 *N*-Queen Problem 416
14.4 Sum of Subsets Problem 418
14.5 Vertex Colouring Problem 421
14.6 Hamiltonian Circuit Problem 423
14.7 Generating Permutation 427
14.8 Graham Scan 428

15 Branch-and-bound Technique 434
15.1 Introduction 434
15.2 Branch and Bound Technique 435
15.3 Search Techniques for Branch-and-Bound Technique 436
15.3.1 BFS Using Branch and Bound Algorithm—FIFOB *437*
15.3.2 LIFO with Branch and Bound *438*
15.3.3 Least Cost with Branch and Bound *438*
15.4 15-puzzle Game 439
15.5 Assignment Problem 442
15.6 Traveling Salesperson Problem 447
15.7 Knapsack Problem 449

16 String Algorithms 457
16.1 Introduction to String Processing 457
16.2 Basic String Algorithms 458
16.2.1 Length of Strings 458
16.2.2 Concatenation of Two Strings 459
16.2.3 Finding Substrings 459
16.3 Longest Common Subsequences 460
16.3.1 Brute Force Method 460
16.3.2 Dynamic Programming 461
16.4 String Edit Distance Using Dynamic Programming 465
16.5 Naïve String-Matching Algorithm 468
16.6 Pattern Matching Using Finite Automata 469
16.7 Rabin–Karp Algorithm 471
16.7.1 Stage 1 471
16.7.2 Stage 2 472
16.8 Knuth–Morris–Pratt Algorithm 474
16.8.1 Stage 1 475
16.8.2 Stage 2 476
16.9 Boyer–Moore String Matching Algorithm 478
16.9.1 Bad-character Shift Rule 478
16.9.2 Good-suffix Shift Rule 478
16.10 Boyar-Moore- Harspool Algorithm 480
16.10.1 Stage 1 480
16.10.2 Stage 2 482

17 Iterative Improvement and Linear Programming 485
17.1 Introduction to Iterative Improvement 485
17.2 Linear Programming 485
17.2.1 Advantages and Limitations of Linear Programming 486
17.3 Formulation of Lpps 486
17.4 Graphical Method for Solving LPP 489
17.5 Simplex Method 493
17.6 Minimization Problems 498
17.7 Duality 501
17.8 Max-Flow Problem 503
17.8.1 Ford–Fulkerson Algorithm 504
17.8.2 Max-flow and Min-cut 506
17.9 Bipartite Matching Problem 507
17.9.1 Bipartite Graph 507
17.10 Stable Marriage Problem 510

18 Basics of Computational Complexity 516
18.1 Introduction to Computational Complexity 516
18.2 Upper and Lower Bound Theory 517
18.2.1 Methods of Finding Lower Bounds 518
18.3 Decision Problems 519
18.4 Turing Machines 520
18.4.1 Languages and Turing Machine 521
18.5 Complexity Classes 521
18.5.1 Class P 522
18.5.2 Class NP 523
18.5.3 NP-hard Problems 523
18.6 Theory of NP-Complete Problems 524
18.7 Reductions 525
18.7.1 Turing Reduction 525
18.7.2 Karp Reduction 526
18.8 Proof of NP-Completeness 527
18.9 Example Problems for Proving NP-Completeness 528
18.9.1 SAT is NP-complete 528
18.9.2 Problem 3-CNF-SAT is NP-complete 529
18.9.3 Clique Decision Problem is NP-complete 530
18.9.4 Sum of Subsets (from 3-CNF-SAT) 531

19 Randomized Algorithms 535
19.1 Introduction to Randomized Algorithms 535
19.2 Generation of Random Numbers 537
19.3 Hiring Problem 539
19.3.1 Probabilistic Analysis of Deterministic Hiring Algorithm 539
19.4 Primality Testing Algorithm 542
19.5 Comparing Two Strings using Randomization Algorithm 543
19.6 Randomized Quicksort 545

20 Approximation Algorithms 549
20.1 Introduction to Approximation Algorithm 549
20.2 Goodness Factor 550
20.3 Design approaches for Approximation Algorithms 551
20.4 Heuristic Approaches for Travelling Salesperson Problem 552
20.4.1 Nearest Neighbor 552
20.4.2 Multi-Fragment Heuristic Method 554
20.4.3 Twice-the-path Heuristic 555
20.4.4 Christofides Algorithm 556
20.5 Greedy Approximation Algorithms 558
20.5.1 0/1 Knapsack Problem 558
20.5.2 Continuous Knapsack Problem 559
20.5.3 An approximation scheme for 0/1 Knapsack problem 559
20.5.4 Vertex Cover Problem 559
20.5.5 Set Cover Problem 561
20.6 Designing Approximation Algorithms Using Dynamic Programming 563
20.6.1 Knapsack Problem 563
20.6.2 Subset Sum Problem 564

21 Parallel Algorithms 570
21.1 Introduction to Parallel Processing 570
21.2 Classification of Parallel Systems 570
21.2.1 Flynn Classification 571
21.2.2 Address-space–based (or Memory Mechanism-based) Classification 572
21.2.3 Classification Based on Interconnection Networks 573
21.3 Introduction to PRAM Model 574
21.4 Parallel Algorithm Specifications and Analysis 575
21.4.1 Parallel Algorithm Analysis 577
21.5 Simple Parallel Algorithms 578
21.5.1 Prefix Computation 578
21.5.2 List Ranking 580
21.5.3 Euler Tour 581
21.6 Parallel Searching and Parallel Sorting 582
21.6.1 Parallel Searching 582
21.6.2 Odd–Even Swap Sort 583
21.7 Additional Parallel Algorithms 584
21.7.1 Parallel Matrix Multiplication 584
21.7.2 Parallel Graph Algorithms 587

Appendix A—Mathematical Basics 591
Appendix B—Proof Techniques 603
Appendix C—Python and Anaconda 612
Appendix D—Python Packages 622
Appendix E—Laboratory Manual 637
Bibliography 675
Index 677