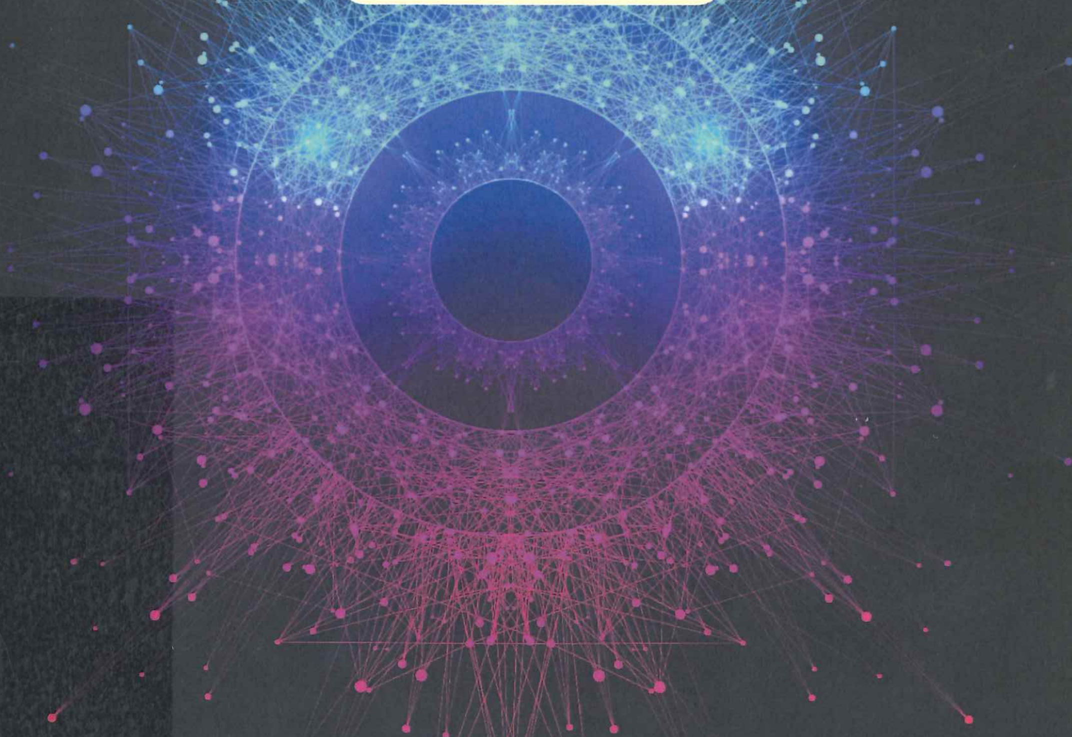
Lecture Notes Series on Computing - Vol. 15

M. H. Alsuwaiyel

Algorithms

Design Techniques and Analysis

Second Edition



World Scientific

Lecture Notes Series on Computing - Vol. 15

Algorithms
Design Techniques and Analysis

Second Edition

M. H. Alsuwaiyel
King Fahd University of Petroleum & Minerals (KFUPM), Saudi Arabia

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI • TOKYO

Contents

Preface vii

About the Author xi

Acknowledgments xiii

Part 1 Basic Concepts and Introduction to Algorithms **1**

Chapter 1 Basic Concepts in Algorithmic Analysis **3**

1.1 Introduction 3
1.2 Historical Background 4
1.3 Binary Search 6
1.3.1 Analysis of the binary search algorithm 8
1.4 Merging Two Sorted Lists 10
1.5 Selection Sort 12
1.6 Insertion Sort 13
1.7 Bottom-Up Merge Sorting 15
1.7.1 Analysis of bottom-up merge sorting 17
1.8 Time Complexity 19
1.8.1 Order of growth 19
1.8.2 The O-notation 23
1.8.3 The Ω-notation 24
1.8.4 The Θ-notation 26
1.8.5 Examples 27
1.8.6 Complexity classes and the o-notation 30

1.9 Space Complexity . . . 31
1.10 Optimal Algorithms . . . 32
1.11 How to Estimate the Running Time of an Algorithm . . . 33
1.11.1 Counting the number of iterations[c] . . . 33
1.11.2 Counting the frequency of basic operations . . . 39
1.11.3 Using recurrence relations . . . 43
1.12 Worst Case and Average Case Analysis . . . 44
1.12.1 Worst case analysis . . . 45
1.12.2 Average case analysis . . . 47
1.13 Amortized Analysis . . . 48
1.14 Input Size and Problem Instance . . . 52
1.15 Divide-and-Conquer Recurrences . . . 54
1.15.1 Expanding the recurrence . . . 54
1.15.2 Substitution . . . 58
1.15.3 Change of variables . . . 63
1.16 Practice Problems . . . 66
1.17 Exercises . . . 68
1.18 Bibliographic Notes . . . 78

Chapter 2 Data Structures **79**

2.1 Introduction . . . 79
2.2 Linked Lists . . . 79
2.2.1 Stacks and queues . . . 80
2.3 Graphs . . . 80
2.3.1 Representation of graphs . . . 82
2.3.2 Planar graphs . . . 83
2.4 Trees . . . 84
2.5 Rooted Trees . . . 84
2.5.1 Tree traversals . . . 85
2.6 Binary Trees . . . 85
2.6.1 Some quantitative aspects of binary trees . . . 87
2.6.2 Binary search trees . . . 87
2.7 Practice Problems . . . 88
2.8 Exercises . . . 89
2.9 Bibliographic Notes . . . 91

Chapter 3 Heaps and the Disjoint Sets Data Structures 93

3.1 Introduction 93
3.2 Heaps 93
3.2.1 Operations on heaps 95
3.2.1.1 Sift-up 95
3.2.1.2 Sift-down 96
3.2.1.3 Insert 97
3.2.1.4 Delete 98
3.2.1.5 Delete-max 98
3.2.2 Creating a heap 99
3.2.3 Heapsort 102
3.2.4 Min and max heaps 103
3.3 Disjoint Sets Data Structures 104
3.3.1 The union by rank heuristic 105
3.3.2 Path compression 107
3.3.3 The union-find algorithms 108
3.3.4 Analysis of the union-find algorithms 110
3.4 Practice Problems 112
3.5 Exercises 113
3.6 Bibliographic Notes 117

Part 2 Techniques Based on Recursion 119

Chapter 4 Induction 121

4.1 Introduction 121
4.2 Finding the Majority Element 122
4.3 Integer Exponentiation 123
4.4 Evaluating Polynomials (Horner's Rule) 124
4.5 Radix Sort 125
4.6 Generating Subsets of a Set 127
4.6.1 The first algorithm 128
4.6.2 The second algorithm 129
4.6.3 Iterative algorithm 131
4.7 Generating Permutations 133
4.7.1 The first algorithm 133
4.7.2 The second algorithm 135
4.7.3 Iterative algorithm 137

4.8 Practice Problems 139
4.9 Exercises 140
4.10 Bibliographic Notes 143

Chapter 5 Divide and Conquer **145**

5.1 Introduction 145
5.2 Binary Search 147
5.2.1 Analysis of the recursive binary search algorithm 147
5.3 Mergesort 149
5.3.1 How the algorithm works 150
5.3.2 Analysis of the mergesort algorithm 152
5.4 The Divide-and-Conquer Paradigm 154
5.5 Selection: Finding the Median and the kth Smallest Element 156
5.5.1 Analysis of the selection algorithm 159
5.6 Quicksort 161
5.6.1 A partitioning algorithm 162
5.6.2 The sorting algorithm 164
5.6.3 Analysis of the quicksort algorithm 166
5.6.3.1 The worst case behavior 167
5.6.3.2 The average case behavior 168
5.6.4 Comparison of sorting algorithms 170
5.7 Quickselect 171
5.8 Multiselection 172
5.9 Multiplication of Large Integers 174
5.10 Matrix Multiplication 175
5.10.1 The traditional algorithm 176
5.10.2 Strassen's algorithm 176
5.10.2.1 Time complexity 177
5.10.3 Comparison of the two algorithms 177
5.11 The Closest Pair Problem 178
5.11.1 Time complexity 181
5.12 A Dominance Problem 183
5.13 Practice Problems 184
5.14 Exercises 186
5.15 Bibliographic Notes 192

Chapter 6 Dynamic Programming 193

6.1 Introduction 193
6.2 The Longest Common Subsequence Problem 195
6.2.1 The algorithm 196
6.3 Matrix Chain Multiplication 198
6.3.1 The dynamic programming algorithm 201
6.4 The Dynamic Programming Paradigm 204
6.5 The All-Pairs Shortest Path Problem 205
6.5.1 The algorithm 205
6.6 The Knapsack Problem 207
6.6.1 The algorithm 208
6.7 Practice Problems 210
6.8 Exercises 211
6.9 Bibliographic Notes 215

Part 3 First-Cut Techniques 217

Chapter 7 The Greedy Approach 219

7.1 Introduction 219
7.2 The Shortest Path Problem 221
7.2.1 Implementation of the shortest path algorithm 223
7.2.2 Correctness 224
7.2.3 Time complexity 224
7.2.4 Improving the time bound 225
7.3 Minimum Cost Spanning Trees (Kruskal's Algorithm) 227
7.3.1 Implementation of Kruskal's algorithm 229
7.3.2 Correctness 229
7.3.3 Time complexity 230
7.4 Minimum Cost Spanning Trees (Prim's Algorithm) . . . 231
7.4.1 Implementation of Prim's algorithm 232
7.4.2 Correctness 234
7.4.3 Time complexity 234
7.4.4 Improving the time bound 235

7.5 File Compression 237
7.5.1 The algorithm 238
7.5.2 Time complexity 239
7.6 Practice Problems 239
7.7 Exercises 240
7.8 Bibliographic Notes 244

Chapter 8 Graph Traversal **245**

8.1 Introduction 245
8.2 Depth-First Search 245
8.2.1 The case of undirected graphs 247
8.2.2 The case of directed graphs 247
8.2.3 Time complexity of depth-first search 249
8.3 Applications of Depth-First Search 250
8.3.1 Graph acyclicity 250
8.3.2 Topological sorting 250
8.3.3 Finding articulation points in a graph 251
8.3.4 Strongly connected components 254
8.4 Breadth-First Search 255
8.4.1 Time complexity 257
8.5 Applications of Breadth-First Search 257
8.6 Practice Problems 258
8.7 Exercises 259
8.8 Bibliographic Notes 261

Part 4 Complexity of Problems 263

Chapter 9 NP-Complete Problems **265**

9.1 Introduction 265
9.2 The Class P 268
9.3 The Class NP 269
9.4 NP-Complete Problems 271
9.4.1 The satisfiability problem 272
9.4.2 Proving NP-completeness 273
9.4.3 3-Satisfiability 275

9.4.4 Vertex cover, independent set and clique problems 276
9.4.5 More NP-complete problems 279
9.5 The Class co-NP 280
9.6 The Relationships between the Three Classes 282
9.7 Practice Problems 283
9.8 Exercises 284
9.9 Bibliographic Notes 285

Chapter 10 Introduction to Computational Complexity 287

10.1 Introduction 287
10.2 Model of Computation: The Turing Machine 287
10.3 k-Tape Turing Machines and Time Complexity 288
10.4 Off-line Turing Machines and Space Complexity 291
10.5 Tape Compression and Linear Speed-up 294
10.6 Relationships between Complexity Classes 295
10.6.1 Space and time hierarchy theorems 298
10.6.2 Padding arguments 300
10.7 Reductions 302
10.8 Completeness 306
10.8.1 NLOGSPACE-complete problems 307
10.8.2 PSPACE-complete problems 308
10.8.3 P-complete problems 309
10.8.4 Some conclusions of completeness 311
10.9 The Polynomial Time Hierarchy 313
10.10 Practice Problems 317
10.11 Exercises 318
10.12 Bibliographic Notes 321

Chapter 11 Lower Bounds 323

11.1 Introduction 323
11.2 Trivial Lower Bounds 323
11.3 The Decision Tree Model 324
11.3.1 The search problem 324
11.3.2 The sorting problem 325
11.3.3 Finding the maximum 327

11.3.4 Finding the largest and second largest elements 327
11.3.4.1 Lower bound 327
11.3.4.2 Optimal algorithm 328
11.4 The Algebraic Decision Tree Model 329
11.4.1 The element uniqueness problem 331
11.4.2 The set equality problem 332
11.4.3 The set inclusion problem 332
11.4.4 The set disjointness problem 333
11.5 Linear Time Reductions 333
11.5.1 The convex hull problem 334
11.5.2 The closest pair problem 335
11.5.3 The Euclidean minimum spanning tree problem 335
11.5.4 The diameter of a point set 336
11.6 Practice Problems 337
11.7 Exercises 337
11.8 Bibliographic Notes 339

Part 5 Coping with Hardness **341**

Chapter 12 Backtracking **345**

12.1 Introduction 345
12.2 The 3-Coloring Problem 345
12.2.1 The algorithm 347
12.3 The 8-Queens Problem 349
12.3.1 The algorithm 350
12.4 The General Backtracking Method 352
12.5 Branch and Bound 354
12.6 Practice Problems 359
12.7 Exercises 360
12.8 Bibliographic Notes 362

Chapter 13 Randomized Algorithms **363**

13.1 Introduction 363
13.2 Las Vegas and Monte Carlo Algorithms 364

13.3 Two Simple Examples 365
13.3.1 A Monte Carlo algorithm 365
13.3.2 A Las Vegas algorithm 366
13.4 Randomized Quicksort 366
13.4.1 Expected running time of randomized quicksort 368
13.5 Randomized Quickselect 369
13.5.1 Expected running time of randomized quickselect 369
13.5.2 The dice problem 372
13.5.3 Application of the dice problem: Quickselect 374
13.6 Occupancy Problems 375
13.6.1 Number of balls in each bin 375
13.6.2 Number of empty bins 376
13.6.3 Balls falling into the same bin 377
13.6.4 Filling all bins 378
13.7 Tail Bounds 379
13.7.1 Markov inequality 379
13.7.2 Chebyshev inequality 380
13.7.3 Chernoff bounds 381
13.7.3.1 Lower tail 381
13.7.3.2 Upper tail 383
13.8 Application of Chernoff Bounds: Multiselection 384
13.8.1 Analysis of the algorithm 387
13.9 Random Sampling 390
13.10 The Min-Cut Problem 393
13.11 Testing String Equality 395
13.12 Pattern Matching 397
13.13 Primality Testing 400
13.14 Practice Problems 405
13.15 Exercises 406
13.16 Bibliographic Notes 409

Chapter 14 Approximation Algorithms 411

14.1 Introduction 411
14.2 Basic Definitions 411

14.3 Difference Bounds 413
14.3.1 Planar graph coloring 413
14.3.2 Hardness result: The knapsack problem 413
14.4 Relative Performance Bounds 414
14.4.1 The bin packing problem 415
14.4.2 The Euclidean traveling salesman problem 417
14.4.3 The vertex cover problem 419
14.4.4 Hardness result: The traveling salesman problem 420
14.5 Polynomial Approximation Schemes 422
14.5.1 The knapsack problem 422
14.6 Fully Polynomial Approximation Schemes 426
14.6.1 The subset-sum problem 426
14.7 Practice Problems 429
14.8 Exercises 429
14.9 Bibliographic Notes 432

Part 6 Iterative Improvement for Domain-Specific Problems 433

Chapter 15 Network Flow 435

15.1 Introduction 435
15.2 Preliminaries 435
15.3 The Ford–Fulkerson Method 439
15.4 Maximum Capacity Augmentation 440
15.5 Shortest Path Augmentation 442
15.6 Dinic's Algorithm 445
15.7 The MPM Algorithm 448
15.8 Practice Problems 451
15.9 Exercises 451
15.10 Bibliographic Notes 453

Chapter 16 Matching 455

16.1 Introduction 455
16.2 Preliminaries 455

16.3 The Network Flow Method for Bipartite Graphs 458
16.4 The Hungarian Tree Method for Bipartite Graphs . . . 459
16.5 Maximum Matching in General Graphs 462
16.6 An $O(n^{2.5})$ Algorithm for Bipartite Graphs 468
16.7 Practice Problems 473
16.8 Exercises 474
16.9 Bibliographic Notes 476

Part 7 Techniques in Computational Geometry 479

Chapter 17 Geometric Sweeping 481

17.1 Introduction 481
17.2 A Simple Example: Computing the Maximal Points of a Point Set 481
17.3 Geometric Preliminaries 483
17.4 Computing the Intersections of Line Segments 485
17.5 The Convex Hull Problem 489
17.6 Computing the Diameter of a Set of Points 492
17.7 Practice Problems 496
17.8 Exercises 497
17.9 Bibliographic Notes 499

Chapter 18 Voronoi Diagrams 501

18.1 Introduction 501
18.2 Nearest-Point Voronoi Diagram 501
18.2.1 Delaunay triangulation 504
18.2.2 Construction of the Voronoi diagram 506
18.3 Applications of the Voronoi Diagram 509
18.3.1 Computing the convex hull 509
18.3.2 All nearest neighbors 510
18.3.3 The Euclidean minimum spanning tree 511
18.4 Farthest-Point Voronoi Diagram 512
18.4.1 Construction of the farthest-point Voronoi diagram 513
18.5 Applications of the Farthest-Point Voronoi Diagram . . 515
18.5.1 All farthest neighbors 516
18.5.2 Smallest enclosing circle 516

18.6 Practice Problems 517
18.7 Exercises 517
18.8 Bibliographic Notes 519

Part 8 Techniques in Parallel Algorithms 521

Chapter 19 Parallel Algorithms 523

19.1 Introduction 523
19.1.1 Classifications of parallel architectures 525
19.2 Shared-Memory Computers (PRAM) 525
19.2.1 The balanced tree method 526
19.2.2 Brent theorem 528
19.2.3 Parallel prefix 529
19.2.3.1 Array packing 531
19.2.3.2 Parallel quicksort 532
19.2.4 Parallel search 533
19.2.5 Pointer jumping 535
19.2.6 Merging by ranking 536
19.2.6.1 Computing ranks 536
19.2.6.2 Merging 540
19.2.6.3 Parallel bottom-up merge sorting 540
19.2.7 The zero-one principle 541
19.2.8 Odd–even merging 542
19.2.9 Bitonic merging and sorting 546
19.2.9.1 Bitonic sorting 551
19.2.10 Pipelined merge sort 553
19.2.10.1 The algorithm 555
19.2.10.2 Maintaining ranks 557
19.2.10.3 Analysis of the algorithm 559
19.2.11 Selection 560
19.2.12 Multiselection 563
19.2.13 Matrix multiplication 567
19.2.14 Transitive closure 568
19.2.15 Shortest paths 569
19.2.16 Computing the convex hull of a set of points 569

19.3 Interconnection-Network Computers 574
19.4 The Hypercube 576
19.4.1 The butterfly 577
19.4.2 Embeddings of the hypercube 580
19.4.2.1 Gray codes 581
19.4.2.2 Embedding of a linear array into the hypercube 582
19.4.2.3 Embedding of a mesh into the hypercube 582
19.4.2.4 Embedding of a binary tree into the hypercube 583
19.4.3 Broadcasting on the hypercube 584
19.4.4 Permutation routing in the hypercube 584
19.4.4.1 The greedy algorithm 585
19.4.4.2 The randomized algorithm 585
19.4.5 Permutation routing in the butterfly 588
19.4.6 Computing parallel prefix on the hypercube 590
19.4.7 Hyperquicksort 591
19.4.8 Selection on the hypercube 593
19.4.9 Multiselection on the hypercube 594
19.4.10 Computing parallel prefix on the butterfly 597
19.5 The Linear Array and the Mesh 598
19.5.1 Broadcasting in the linear array and the mesh 599
19.5.2 Computing parallel prefix on the mesh 600
19.5.3 Odd–even transposition sort 601
19.5.4 Shearsort 602
19.5.5 Odd–even merging and sorting on the mesh . . 604
19.5.6 Computing the convex hull of a set of points on the mesh 606
19.5.6.1 The first algorithm 607
19.5.6.2 The second algorithm 608

19.5.7 Three-dimensional mesh 612
19.5.7.1 Sorting on three-dimensional meshes 613
19.5.8 Fast Fourier transform 616
19.5.8.1 Implementation on the butterfly 618
19.5.8.2 Iterative FFT on the butterfly . . . 620
19.5.8.3 The inverse Fourier transform . . . 622
19.5.8.4 Applications of FFT 624
19.5.8.5 Product of polynomials 624
19.5.8.6 Computing the convolution of two vectors 626
19.6 Systolic Computation 627
19.6.1 An on-chip bubble sorter 627
19.7 Practice Problems 631
19.8 Exercises 634
19.9 Bibliographic Notes 640
19.10 Solutions 641

Appendix A. Mathematical Preliminaries **661**

A.1 Sets, Relations and Functions 661
A.1.1 Sets 662
A.1.2 Relations 663
A.1.2.1 Equivalence relations 664
A.1.3 Functions 664
A.2 Proof Methods 665
A.2.1 Notation 665
A.2.2 Direct proof 666
A.2.3 Indirect proof 666
A.2.4 Proof by contradiction 666
A.2.5 Proof by counterexample 667
A.2.6 Mathematical induction 668
A.3 Logarithms 670
A.4 Floor and Ceiling Functions 671
A.5 Factorial and Binomial Coefficients 672
A.5.1 Factorials 672
A.5.2 Binomial coefficients 673

A.6 The Pigeonhole Principle 675
A.7 Summations 676
A.7.1 Approximation of summations by integration 679
A.8 Recurrence Relations 682
A.8.1 Solution of linear homogeneous recurrences 683
A.8.2 Solution of inhomogeneous recurrences 685
A.9 Divide-and-Conquer Recurrences 688
A.10 Exercises 688

Appendix B. Introduction to Discrete Probability 691

B.1 Definitions 691
B.2 Conditional Probability and Independence 691
B.2.1 Multiplication rule for conditional probability 693
B.3 Random Variables and Expectation 693
B.4 Discrete Probability Distributions 694
B.4.1 Uniform distribution 694
B.4.2 Bernoulli distribution 695
B.4.3 Binomial distribution 695
B.4.4 Geometric distribution 696
B.4.5 Poisson distribution 696

Bibliography 697

Index 709