

Contents

PREFACE	v
1 INTRODUCTION	1
1.1 WHAT IS AN ALGORITHM?	1
1.2 ALGORITHM SPECIFICATION	5
1.2.1 Introduction	5
1.2.2 Recursive Algorithms	7
1.3 PERFORMANCE ANALYSIS	11
1.3.1 Space Complexity	12
1.3.2 Time Complexity	14
1.3.3 Amortized Complexity	24
1.3.4 Asymptotic Notation (O , Ω , Θ)	36
1.3.5 Practical Complexities	44
1.3.6 Performance Measurement	45
1.4 RANDOMIZED ALGORITHMS	59
1.4.1 Basics of Probability Theory	59
1.4.2 Randomized Algorithms: An Informal Description	63
1.4.3 Identifying the Repeated Element	64
1.4.4 Primality Testing	66
1.4.5 Advantages and Disadvantages	70
1.5 REFERENCES AND READINGS	73
2 ELEMENTARY DATA STRUCTURES	75
2.1 STACKS AND QUEUES	75
2.2 TREES	84
2.2.1 Terminology	84
2.2.2 Binary Trees	85
2.3 DICTIONARIES	90

2.3.1	Binary Search Trees	90
2.4	PRIORITY QUEUES	98
2.4.1	Heaps	99
2.4.2	Heapsort	105
2.5	SETS AND DISJOINT SET UNION	109
2.5.1	Introduction	109
2.5.2	Union and Find Operations	110
2.6	GRAPHS	120
2.6.1	Introduction	120
2.6.2	Definitions	121
2.6.3	Graph Representations	125
2.7	REFERENCES AND READINGS	135
3	DIVIDE-AND-CONQUER	136
3.1	GENERAL METHOD	136
3.2	DEFECTIVE CHESSBOARD	141
3.3	BINARY SEARCH	145
3.4	FINDING THE MAXIMUM AND MINIMUM	154
3.5	MERGE SORT	159
3.6	QUICKSORT	168
3.6.1	Performance Measurement	173
3.6.2	Randomized Sorting Algorithms	173
3.7	SELECTION	178
3.7.1	A Worst-Case Optimal Algorithm	182
3.7.2	Implementation of <code>Select2</code>	185
3.8	STRASSEN'S MATRIX MULTIPLICATION	192
3.9	CONVEX HULL	196
3.9.1	Some Geometric Primitives	197
3.9.2	The <code>QuickHull</code> Algorithm	198
3.9.3	Graham's Scan	200
3.9.4	An $O(n \log n)$ Divide-and-Conquer Algorithm	201
3.10	REFERENCES AND READINGS	206
3.11	ADDITIONAL EXERCISES	207
4	THE GREEDY METHOD	210
4.1	THE GENERAL METHOD	210
4.2	CONTAINER LOADING	214
4.3	KNAPSACK PROBLEM	218

4.4	TREE VERTEX SPLITTING	222
4.5	JOB SEQUENCING WITH DEADLINES	227
4.6	MINIMUM-COST SPANNING TREES	236
4.6.1	Prim's Algorithm	237
4.6.2	Kruskal's Algorithm	240
4.6.3	An Optimal Randomized Algorithm (*)	244
4.7	OPTIMAL STORAGE ON TAPES	249
4.8	OPTIMAL MERGE PATTERNS	254
4.9	SINGLE-SOURCE SHORTEST PATHS	261
4.10	REFERENCES AND READINGS	268
4.11	ADDITIONAL EXERCISES	270
5	DYNAMIC PROGRAMMING	272
5.1	THE GENERAL METHOD	272
5.2	MULTISTAGE GRAPHS	276
5.3	ALL PAIRS SHORTEST PATHS	284
5.4	SINGLE-SOURCE SHORTEST PATHS: GENERAL WEIGHTS	289
5.5	OPTIMAL BINARY SEARCH TREES (*)	294
5.6	STRING EDITING	303
5.7	0/1-KNAPSACK	306
5.8	RELIABILITY DESIGN	315
5.9	THE TRAVELING SALESPERSON PROBLEM	318
5.10	FLOW SHOP SCHEDULING	321
5.11	REFERENCES AND READINGS	327
5.12	ADDITIONAL EXERCISES	328
6	BASIC TRAVERSAL AND SEARCH TECHNIQUES	333
6.1	TECHNIQUES FOR BINARY TREES	333
6.2	TECHNIQUES FOR GRAPHS	339
6.2.1	Breadth First Search and Traversal	340
6.2.2	Depth First Search and Traversal	343
6.3	CONNECTED COMPONENTS AND SPANNING TREES	345
6.4	BICONNECTED COMPONENTS AND DFS	349
6.5	REFERENCES AND READINGS	358

7	BACKTRACKING	359
7.1	THE GENERAL METHOD	359
7.2	THE 8-QUEENS PROBLEM	373
7.3	SUM OF SUBSETS	377
7.4	GRAPH COLORING	379
7.5	HAMILTONIAN CYCLES	383
7.6	KNAPSACK PROBLEM	387
7.7	REFERENCES AND READINGS	393
7.8	ADDITIONAL EXERCISES	394
8	BRANCH AND BOUND	398
8.1	THE METHOD	398
8.1.1	Least Cost (LC) Search	399
8.1.2	The 15-puzzle: An Example	401
8.1.3	Control Abstractions for LC-Search	405
8.1.4	Bounding	407
8.1.5	FIFO Branch-and-Bound	410
8.1.6	LC Branch-and-Bound	411
8.2	0/1 KNAPSACK PROBLEM	412
8.2.1	LC Branch-and-Bound Solution	413
8.2.2	FIFO Branch-and-Bound Solution	416
8.3	TRAVELING SALESPERSON (*)	421
8.4	EFFICIENCY CONSIDERATIONS	430
8.5	REFERENCES AND READINGS	433
9	ALGEBRAIC PROBLEMS	435
9.1	THE GENERAL METHOD	435
9.2	EVALUATION AND INTERPOLATION	438
9.3	THE FAST FOURIER TRANSFORM	449
9.3.1	An In-place Version of the FFT	455
9.3.2	Some Remaining Points	458
9.4	MODULAR ARITHMETIC	459
9.5	EVEN FASTER EVALUATION AND INTERPOLATION	467
9.6	REFERENCES AND READINGS	475

10 LOWER BOUND THEORY	477
10.1 COMPARISON TREES	478
10.1.1 Ordered Searching	479
10.1.2 Sorting	479
10.1.3 Selection	484
10.2 ORACLES AND ADVERSARY ARGUMENTS	486
10.2.1 Merging	487
10.2.2 Largest and Second Largest	488
10.2.3 State Space Method	490
10.2.4 Selection	491
10.3 LOWER BOUNDS THROUGH REDUCTIONS	494
10.3.1 Finding the Convex Hull	495
10.3.2 Disjoint Sets Problem	495
10.3.3 On-line Median Finding	497
10.3.4 Multiplying Triangular Matrices	497
10.3.5 Inverting a Lower Triangular Matrix	498
10.3.6 Computing the Transitive Closure	500
10.4 TECHNIQUES FOR ALGEBRAIC PROBLEMS (*)	504
10.5 REFERENCES AND READINGS	513
11 \mathcal{NP}-HARD AND \mathcal{NP}-COMPLETE PROBLEMS	515
11.1 BASIC CONCEPTS	515
11.1.1 Nondeterministic Algorithms	516
11.1.2 The Classes \mathcal{NP} -hard and \mathcal{NP} -complete	524
11.2 COOK'S THEOREM (*)	528
11.3 \mathcal{NP} -HARD GRAPH PROBLEMS	537
11.3.1 Clique Decision Problem (CDP)	538
11.3.2 Node Cover Decision Problem (NCDP)	539
11.3.3 Chromatic Number Decision Problem (CNDP)	540
11.3.4 Directed Hamiltonian Cycle (DHC) (*)	542
11.3.5 Traveling Salesperson Decision Problem (TSP)	546
11.3.6 AND/OR Graph Decision Problem (AOG)	546
11.4 \mathcal{NP} -HARD SCHEDULING PROBLEMS	553
11.4.1 Scheduling Identical Processors	554
11.4.2 Flow Shop Scheduling	556

11.4.3	Job Shop Scheduling	558
11.5	\mathcal{NP} -HARD CODE GENERATION PROBLEMS	560
11.5.1	Code Generation with Common Subexpressions	563
11.5.2	Implementing Parallel Assignment Instructions	566
11.6	SOME SIMPLIFIED \mathcal{NP} -HARD PROBLEMS	569
11.7	REFERENCES AND READINGS	573
11.8	ADDITIONAL EXERCISES	573
12	APPROXIMATION ALGORITHMS	577
12.1	INTRODUCTION	577
12.2	ABSOLUTE APPROXIMATIONS	581
12.2.1	Planar Graph Coloring	581
12.2.2	Maximum Programs Stored Problem	582
12.2.3	\mathcal{NP} -hard Absolute Approximations	583
12.3	ϵ -APPROXIMATIONS	585
12.3.1	Scheduling Independent Tasks	585
12.3.2	Bin Packing	589
12.3.3	\mathcal{NP} -hard ϵ -approximation Problems	592
12.4	POLYNOMIAL TIME APPROXIMATION SCHEMES	598
12.4.1	Scheduling Independent Tasks	598
12.4.2	0/1 Knapsack	600
12.5	FULLY POLYNOMIAL TIME APPROXIMATION SCHEMES	605
12.5.1	Rounding	606
12.5.2	Interval Partitioning	610
12.5.3	Separation	611
12.6	PROBABILISTICALLY GOOD ALGORITHMS (*)	616
12.7	REFERENCES AND READINGS	618
12.8	ADDITIONAL EXERCISES	619
13	PRAM ALGORITHMS	624
13.1	INTRODUCTION	624
13.2	COMPUTATIONAL MODEL	627
13.3	FUNDAMENTAL TECHNIQUES AND ALGORITHMS	633
13.3.1	Prefix Computation	634

13.3.2	List Ranking	636
13.4	SELECTION	645
13.4.1	Maximal Selection with n^2 Processors	646
13.4.2	Finding the Maximum Using n Processors	647
13.4.3	Maximal Selection Among Integers	647
13.4.4	General Selection Using n^2 Processors	650
13.4.5	A Work-Optimal Randomized Algorithm (*)	650
13.5	MERGING	654
13.5.1	A Logarithmic Time Algorithm	654
13.5.2	Odd-Even Merge	655
13.5.3	A Work-Optimal Algorithm	657
13.5.4	An $O(\log \log m)$ -Time Algorithm	659
13.6	SORTING	661
13.6.1	Odd-Even Merge Sort	661
13.6.2	An Alternative Randomized Algorithm	662
13.6.3	Preparata's Algorithm	663
13.6.4	Reischuk's Randomized Algorithm (*)	664
13.7	GRAPH PROBLEMS	668
13.7.1	An Alternative Algorithm for Transitive Closure	671
13.7.2	All-Pairs Shortest Paths	671
13.8	COMPUTING THE CONVEX HULL	672
13.9	LOWER BOUNDS	676
13.9.1	A lower bound on average-case sorting	677
13.9.2	Finding the maximum	678
13.10	REFERENCES AND READINGS	680
13.11	ADDITIONAL EXERCISES	682
14	MESH ALGORITHMS	684
14.1	COMPUTATIONAL MODEL	684
14.2	PACKET ROUTING	686
14.2.1	Packet Routing on a Linear Array	687
14.2.2	A Greedy Algorithm for PPR on a Mesh	691
14.2.3	A Randomized Algorithm with Small Queues	692
14.3	FUNDAMENTAL ALGORITHMS	696
14.3.1	Broadcasting	697
14.3.2	Prefix Computation	698
14.3.3	Data Concentration	702
14.3.4	Sparse Enumeration Sort	703

14.4	SELECTION	707
14.4.1	A Randomized Algorithm for $n = p$ (*)	708
14.4.2	Randomized Selection for $n > p$ (*)	708
14.4.3	A Deterministic Algorithm For $n > p$	709
14.5	MERGING	713
14.5.1	Rank Merge on a Linear Array	714
14.5.2	Odd-Even Merge on a Linear Array	714
14.5.3	Odd-Even Merge on a Mesh	715
14.6	SORTING	717
14.6.1	Sorting on a Linear Array	717
14.6.2	Sorting on a Mesh	719
14.7	GRAPH PROBLEMS	724
14.7.1	An $n \times n$ Mesh Algorithm for Transitive Closure	726
14.7.2	All-Pairs Shortest Paths	727
14.8	COMPUTING THE CONVEX HULL	728
14.9	REFERENCES AND READINGS	733
14.10	ADDITIONAL EXERCISES	735
15	HYPERCUBE ALGORITHMS	739
15.1	COMPUTATIONAL MODEL	739
15.1.1	The Hypercube	739
15.1.2	The Butterfly Network	742
15.1.3	Embedding of Other Networks	743
15.2	PPR ROUTING	748
15.2.1	A Greedy Algorithm	748
15.2.2	A Randomized Algorithm	748
15.3	FUNDAMENTAL ALGORITHMS	752
15.3.1	Broadcasting	752
15.3.2	Prefix Computation	753
15.3.3	Data Concentration	754
15.3.4	Sparse Enumeration Sort	757
15.4	SELECTION	760
15.4.1	A Randomized Algorithm for $n = p$ (*)	760
15.4.2	Randomized Selection for $n > p$ (*)	760
15.4.3	A Deterministic Algorithm for $n > p$	761
15.5	MERGING	763
15.5.1	Odd-Even Merge	763
15.5.2	Bitonic Merge	765

15.6 SORTING	766
15.6.1 Odd-Even Merge Sort	766
15.6.2 Bitonic Sort	768
15.7 GRAPH PROBLEMS	769
15.8 COMPUTING THE CONVEX HULL	771
15.9 REFERENCES AND READINGS	772
15.10 ADDITIONAL EXERCISES	773