

Lecture Notes Series on Computing - Vol. 14

Algorithms

Design Techniques
and Analysis

Revised Edition

M. H. Alsuwaiyel

 **World Scientific**

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

Contents

Preface	v
PART 1 Basic Concepts and Introduction to Algorithms	1
Chapter 1 Basic Concepts in Algorithmic Analysis	5
1.1 Introduction	5
1.2 Historical Background	6
1.3 Binary Search	8
1.3.1 Analysis of the binary search algorithm	10
1.4 Merging Two Sorted Lists	12
1.5 Selection Sort	14
1.6 Insertion Sort	15
1.7 Bottom-up Merge Sorting	17
1.7.1 Analysis of bottom-up merge sorting	19
1.8 Time Complexity	20
1.8.1 Order of growth	21
1.8.2 The O -notation	25
1.8.3 The Ω -notation	26
1.8.4 The Θ -notation	27
1.8.5 Examples	29
1.8.6 Complexity classes and the o -notation	32
1.9 Space Complexity	33

1.10	Optimal Algorithms	34
1.11	How to Estimate the Running Time of an Algorithm	35
	1.11.1 Counting the number of iterations	35
	1.11.2 Counting the frequency of basic operations	40
	1.11.3 Using recurrence relations	44
1.12	Worst-Case and Average-Case Analyses	45
	1.12.1 Worst-case analysis	46
	1.12.2 Average-case analysis	48
1.13	Amortized Analysis	50
1.14	Input Size and Problem Instance	53
1.15	Divide-and-Conquer Recurrences	55
	1.15.1 Expanding the recurrence	56
	1.15.2 Substitution	60
	1.15.3 Change of variables	64
1.16	Exercises	67
1.17	Bibliographic Notes	76
Chapter 2 Data Structures		77
2.1	Introduction	77
2.2	Linked Lists	77
	2.2.1 Stacks and queues	78
2.3	Graphs	78
	2.3.1 Representation of graphs	80
	2.3.2 Planar graphs	81
2.4	Trees	82
2.5	Rooted Trees	82
	2.5.1 Tree traversals	83
2.6	Binary Trees	83
	2.6.1 Some quantitative aspects of binary trees	85
	2.6.2 Binary search trees	86
2.7	Exercises	86
2.8	Bibliographic Notes	88
Chapter 3 Heaps and the Disjoint Sets Data Structures		89
3.1	Introduction	89
3.2	Heaps	89
	3.2.1 Operations on heaps	90
	3.2.2 Creating a heap	94

3.2.3	Heapsort	98
3.2.4	Min and Max Heaps	99
3.3	Disjoint Sets Data Structures	100
3.3.1	The union by rank heuristic	101
3.3.2	Path compression	103
3.3.3	The union-find algorithms	104
3.3.4	Analysis of the union-find algorithms	105
3.4	Exercises	108
3.5	Bibliographic Notes	111

PART 2 Techniques Based on Recursion 113

Chapter 4 Induction 117

4.1	Introduction	117
4.2	Finding the Majority Element	118
4.3	Integer Exponentiation	119
4.4	Evaluating Polynomials (Horner's Rule)	120
4.5	Radix Sort	121
4.6	Generating Permutations	123
4.6.1	The first algorithm	124
4.6.2	The second algorithm	125
4.7	Exercises	127
4.8	Bibliographic Notes	130

Chapter 5 Divide and Conquer 131

5.1	Introduction	131
5.2	Binary Search	133
5.3	Mergesort	135
5.3.1	How the algorithm works	136
5.3.2	Analysis of the mergesort algorithm	137
5.4	The Divide-and-Conquer Paradigm	139
5.5	Selection: Finding the Median and the k th Smallest Element	142
5.5.1	Analysis of the selection algorithm	145
5.6	Quicksort	147
5.6.1	A partitioning algorithm	147
5.6.2	The sorting algorithm	150

5.6.3	Analysis of the quicksort algorithm	152
5.6.3.1	The worst-case behavior	152
5.6.3.2	The average-case behavior	153
5.6.4	Comparison of sorting algorithms	156
5.7	Multiselection	156
5.8	Multiplication of Large Integers	158
5.9	Matrix Multiplication	160
5.9.1	The traditional algorithm	160
5.9.2	Strassen's algorithm	160
5.9.3	Comparisons of the two algorithms	162
5.10	The Closest Pair Problem	162
5.10.1	Time complexity	165
5.11	Exercises	167
5.12	Bibliographic Notes	172
Chapter 6 Dynamic Programming		173
6.1	Introduction	173
6.2	The Longest Common Subsequence Problem	175
6.3	Matrix Chain Multiplication	178
6.4	The Dynamic Programming Paradigm	184
6.5	The All-Pairs Shortest Path Problem	185
6.6	The Knapsack Problem	187
6.7	Exercises	190
6.8	Bibliographic Notes	196
PART 3 First-Cut Techniques		197
Chapter 7 The Greedy Approach		201
7.1	Introduction	201
7.2	The Shortest Path Problem	202
7.2.1	Improving the time bound	207
7.3	Minimum Cost Spanning Trees (Kruskal's Algorithm)	209
7.4	Minimum Cost Spanning Trees (Prim's Algorithm)	212
7.4.1	Improving the time bound	216
7.5	File Compression	218
7.6	Exercises	221
7.7	Bibliographic Notes	225

Chapter 8	Graph Traversal	227
8.1	Introduction	227
8.2	Depth-First Search	227
	8.2.1 Time complexity of depth-first search	231
8.3	Applications of Depth-First Search	232
	8.3.1 Graph acyclicity	232
	8.3.2 Topological sorting	232
	8.3.3 Finding articulation points in a graph	233
	8.3.4 Strongly connected components	236
8.4	Breadth-First Search	237
8.5	Applications of Breadth-First Search	239
8.6	Exercises	239
8.7	Bibliographic Notes	242
PART 4	Complexity of Problems	245
Chapter 9	NP-complete Problems	249
9.1	Introduction	249
9.2	The Class P	252
9.3	The Class NP	253
9.4	NP-complete Problems	255
	9.4.1 The satisfiability problem	256
	9.4.2 Proving NP-completeness	256
	9.4.3 Vertex cover, independent set, and clique problems	258
	9.4.4 More NP-complete problems	262
9.5	The Class co-NP	263
9.6	The Relationships Between the Three Classes	265
9.7	Exercises	265
9.8	Bibliographic Notes	267
Chapter 10	Introduction to Computational Complexity	269
10.1	Introduction	269
10.2	Model of Computation: The Turing Machine	269
10.3	k -Tape Turing Machines and Time Complexity	270
10.4	Off-line Turing Machines and Space Complexity	273
10.5	Tape Compression and Linear Speed-up	276
10.6	Relationships Between Complexity Classes	277

10.6.1	Space and time hierarchy theorems	280
10.6.2	Padding arguments	282
10.7	Reductions	284
10.8	Completeness	288
10.8.1	NLOGSPACE-complete problems	289
10.8.2	PSPACE-complete problems	290
10.8.3	P-complete problems	291
10.8.4	Some conclusions of completeness	293
10.9	The Polynomial Time Hierarchy	294
10.10	Exercises	299
10.11	Bibliographic Notes	302

Chapter 11 Lower Bounds 305

11.1	Introduction	305
11.2	Trivial Lower Bounds	305
11.3	The Decision Tree Model	306
11.3.1	The search problem	306
11.3.2	The sorting problem	307
11.4	The Algebraic Decision Tree Model	309
11.4.1	The element uniqueness problem	311
11.5	Linear Time Reductions	312
11.5.1	The convex hull problem	312
11.5.2	The closest pair problem	313
11.5.3	The Euclidean minimum spanning tree problem	314
11.6	Exercises	315
11.7	Bibliographic Notes	316

PART 5 Coping with Hardness 317

Chapter 12 Backtracking 321

12.1	Introduction	321
12.2	The 3-Coloring Problem	321
12.3	The 8-Queens Problem	325
12.4	The General Backtracking Method	328
12.5	Branch and Bound	330
12.6	Exercises	335
12.7	Bibliographic Notes	337

Chapter 13	Randomized Algorithms	339
13.1	Introduction	339
13.2	Las Vegas and Monte Carlo Algorithms	340
13.3	Two Simple Examples	341
	13.3.1 A Monte Carlo algorithm	341
	13.3.2 A Las Vegas algorithm	342
13.4	Randomized Quicksort	342
	13.4.1 Expected running time of randomized quicksort	344
13.5	Randomized Selection	345
	13.5.1 Expected running time of randomized selection	345
13.6	Occupancy Problems	348
	13.6.1 Number of balls in each bin	348
	13.6.2 Number of empty bins	349
	13.6.3 Balls falling into the same bin	350
	13.6.4 Filling all bins	351
13.7	Tail Bounds	352
	13.7.1 Markov inequality	352
	13.7.2 Chebyshev inequality	353
	13.7.3 Chernoff bounds	354
	13.7.3.1 Lower tail	354
	13.7.3.2 Upper tail	356
13.8	Application of Chernoff Bounds: Multiselection	357
	13.8.1 Analysis of the algorithm	360
13.9	Random Sampling	363
13.10	The Min-Cut Problem	366
13.11	Testing String Equality	368
13.12	Pattern Matching	370
13.13	Primality Testing	373
13.14	Exercises	378
13.15	Bibliographic Notes	381
Chapter 14	Approximation Algorithms	383
14.1	Introduction	383
14.2	Basic Definitions	384
14.3	Difference Bounds	385
	14.3.1 Planar graph coloring	385
	14.3.2 Hardness result: The knapsack problem	385

14.4	Relative Performance Bounds	386
14.4.1	The bin packing problem	387
14.4.2	The Euclidean traveling salesman problem	389
14.4.3	The vertex cover problem	391
14.4.4	Hardness result: The traveling salesman problem	392
14.5	Polynomial Approximation Schemes	394
14.5.1	The knapsack problem	394
14.6	Fully Polynomial Approximation Schemes	397
14.6.1	The subset-sum problem	398
14.7	Exercises	400
14.8	Bibliographic Notes	403

PART 6 Iterative Improvement for Domain-Specific Problems 405

Chapter 15	Network Flow	409
15.1	Introduction	409
15.2	Preliminaries	409
15.3	The Ford–Fulkerson Method	413
15.4	Maximum Capacity Augmentation	414
15.5	Shortest Path Augmentation	416
15.6	Dinic’s Algorithm	419
15.7	The MPM Algorithm	423
15.8	Exercises	425
15.9	Bibliographic Notes	426

Chapter 16	Matching	427
16.1	Introduction	427
16.2	Preliminaries	427
16.3	The Network Flow Method for Bipartite Graphs	430
16.4	The Hungarian Tree Method for Bipartite Graphs	431
16.5	Maximum Matching in General Graphs	434
16.6	An $O(n^{2.5})$ Algorithm for Bipartite Graphs	440
16.7	Exercises	445
16.8	Bibliographic Notes	447

PART 7	Techniques in Computational Geometry	449
Chapter 17	Geometric Sweeping	453
17.1	Introduction	453
17.2	A Simple Example: Computing the Maximal Points of a Point Set	453
17.3	Geometric Preliminaries	455
17.4	Computing the Intersections of Line Segments	457
17.5	The Convex Hull Problem	460
17.6	Computing the Diameter of a Set of Points	463
17.7	Exercises	468
17.8	Bibliographic Notes	470
Chapter 18	Voronoi Diagrams	473
18.1	Introduction	473
18.2	Nearest-Point Voronoi Diagram	473
	18.2.1 Delaunay triangulation	476
	18.2.2 Construction of the Voronoi diagram	478
18.3	Applications of the Voronoi Diagram	481
	18.3.1 Computing the convex hull	481
	18.3.2 All nearest neighbors	482
	18.3.3 The Euclidean minimum spanning tree	483
18.4	Farthest-Point Voronoi Diagram	484
	18.4.1 Construction of the farthest-point Voronoi diagram	486
18.5	Applications of the Farthest-Point Voronoi Diagram	488
	18.5.1 All farthest neighbors	488
	18.5.2 Smallest enclosing circle	488
18.6	Exercises	489
18.7	Bibliographic Notes	490
Appendix A	Mathematical Preliminaries	493
A.1	Sets, Relations, and Functions	493
	A.1.1 Sets	494
	A.1.2 Relations	495
	A.1.2.1 Equivalence relations	496
	A.1.3 Functions	496

A.2	Proof Methods	497
A.2.1	Direct proof	497
A.2.2	Indirect proof	498
A.2.3	Proof by contradiction	498
A.2.4	Proof by counterexample	499
A.2.5	Mathematical induction	500
A.3	Logarithms	502
A.4	Floor and Ceiling Functions	503
A.5	Factorial and Binomial Coefficients	504
A.5.1	Factorials	504
A.5.2	Binomial coefficients	505
A.6	The Pigeonhole Principle	507
A.7	Summations	508
A.7.1	Approximation of summations by integration	510
A.8	Recurrence Relations	514
A.8.1	Solution of linear homogeneous recurrences	515
A.8.2	Solution of inhomogeneous recurrences	517
A.9	Divide-and-Conquer Recurrences	519
A.10	Exercises	519
Appendix B Introduction to Discrete Probability		523
B.1	Definitions	523
B.2	Conditional Probability and Independence	523
B.2.1	Multiplication rule for conditional probability	525
B.3	Random Variables and Expectation	525
B.4	Discrete Probability Distributions	526
B.4.1	Uniform distribution	526
B.4.2	Bernoulli distribution	527
B.4.3	Binomial distribution	527
B.4.4	Geometric distribution	528
B.4.5	Poisson distribution	528
Bibliography		529
Index		539