



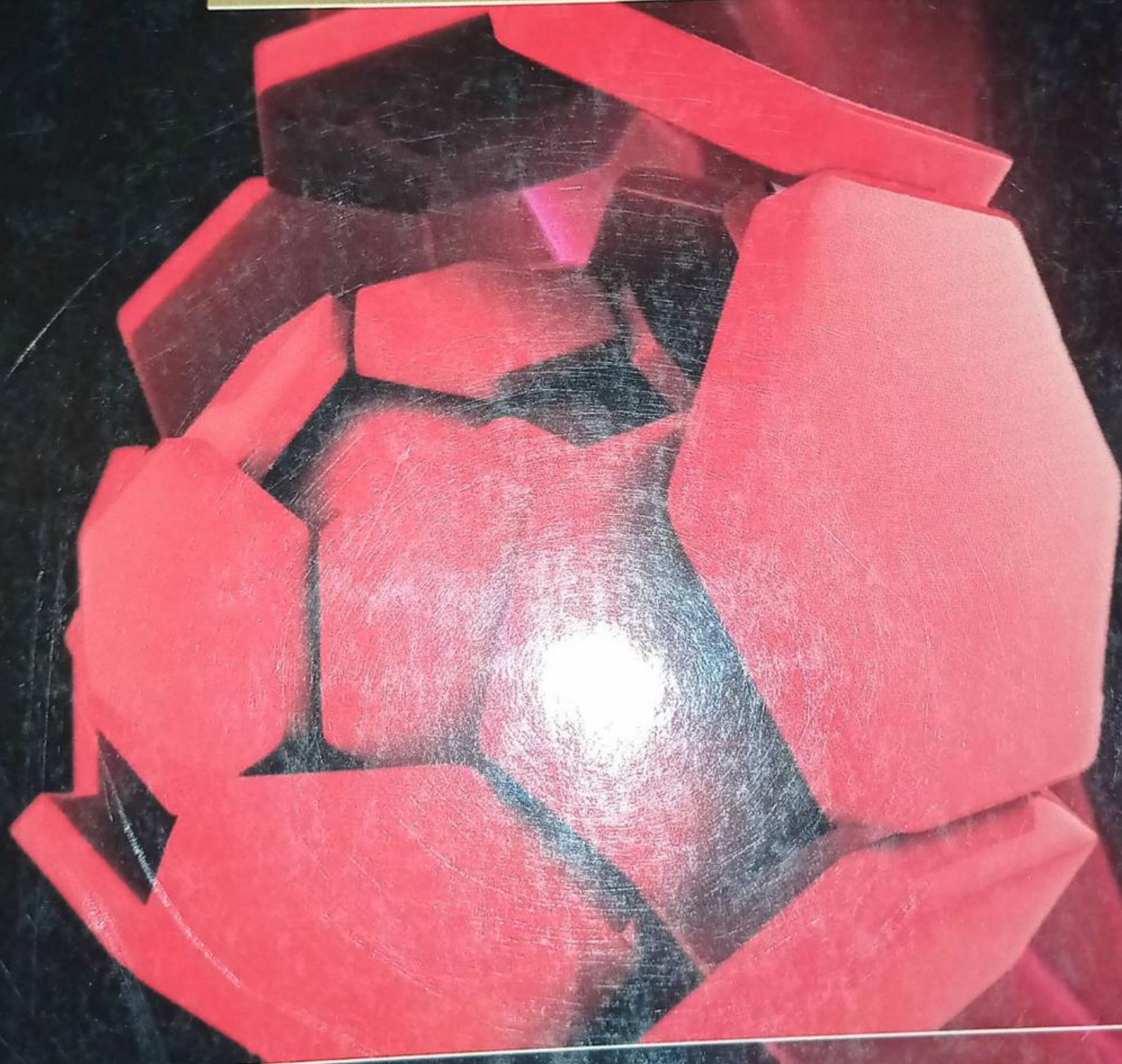
Universities Press

COMPUTER SCIENCE

Fundamentals of Computer

Algorithms

SECOND EDITION



Ellis Horowitz
Sartaj Sahni
Sanguthevar Rajasekaran

Contents

PREFACE

1 INTRODUCTION

1.1	WHAT IS AN ALGORITHM?	1
1.2	ALGORITHM SPECIFICATION	1
1.2.1	Pseudocode Conventions	5
1.2.2	Recursive Algorithms	5
1.3	PERFORMANCE ANALYSIS	10
1.3.1	Space Complexity	14
1.3.2	Time Complexity	15
1.3.3	Amortized Complexity	17
1.3.4	Asymptotic Notation (O , Ω , Θ)	28
1.3.5	Practical Complexities	39
1.3.6	Performance Measurement	47
1.4	RANDOMIZED ALGORITHMS	50
1.4.1	Basics of Probability Theory	62
1.4.2	Randomized Algorithms: An Informal Description	67
1.4.3	Identifying the Repeated Element	68
1.4.4	Primality Testing	70
1.4.5	Advantages and Disadvantages	74
1.5	REFERENCES AND READINGS	77

2 ELEMENTARY DATA STRUCTURES

2.1	STACKS AND QUEUES	79
2.2	TREES	87
2.2.1	Terminology	87
2.2.2	Binary Trees	88
2.3	DICTIONARIES	92

2.3.1	Binary Search Trees	93
2.4	PRIORITY QUEUES	100
2.4.1	Heaps	101
2.4.2	Heap Sort	107
2.5	SETS AND DISJOINT SET UNION	110
2.5.1	Introduction	110
2.5.2	Union and Find Operations	111
2.6	GRAPHS	121
2.6.1	Introduction	121
2.6.2	Definitions	121
2.6.3	Graph Representations	126
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	153
3.5	MERGE SORT	159
3.6	QUICK SORT	168
3.6.1	Performance Measurement	172
3.6.2	Randomized Sorting Algorithms	174
3.7	SELECTION	178
3.7.1	A Worst-Case Optimal Algorithm	182
3.7.2	Implementation of Select2	186
3.8	STRASSEN'S MATRIX MULTIPLICATION	192
3.9	CONVEX HULL	197
3.9.1	Some Geometric Primitives	198
3.9.2	The QuickHull Algorithm	199
3.9.3	Graham's Scan	200
3.9.4	An $O(n \log n)$ Divide-and-Conquer Algorithm	203
3.10	REFERENCES AND READINGS	207
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	239
4.6.3	An Optimal Randomized Algorithm (*)	244
4.7	OPTIMAL STORAGE ON TAPES	249
4.8	OPTIMAL MERGE PATTERNS	253
4.9	SINGLE-SOURCE SHORTEST PATHS	260
4.10	REFERENCES AND READINGS	268
4.11	ADDITIONAL EXERCISES	269
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	288
5.5	OPTIMAL BINARY SEARCH TREES (*)	293
5.6	STRING EDITING	302
5.7	0/1 KNAPSACK	305
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	327
6	BASIC TRAVERSAL AND SEARCH TECHNIQUES	333
6.1	TECHNIQUES FOR BINARY TREES	333
6.2	TECHNIQUES FOR GRAPHS	338
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	380
7.5	HAMILTONIAN CYCLES	384
7.6	KNAPSACK PROBLEM	387
7.7	REFERENCES AND READINGS	393
7.8	ADDITIONAL EXERCISES	394
8	BRANCH AND BOUND	399
8.1	THE METHOD	399
8.1.1	Least Cost (LC) Search	400
8.1.2	The 15-puzzle: An Example	402
8.1.3	Control Abstractions for LC-Search	406
8.1.4	Bounding	408
8.1.5	FIFO Branch-and-Bound	411
8.1.6	LC Branch-and-Bound	412
8.2	0/1 KNAPSACK PROBLEM	413
8.2.1	LC Branch-and-Bound Solution	414
8.2.2	FIFO Branch-and-Bound Solution	417
8.3	TRAVELING SALESPERSON (*)	422
8.4	EFFICIENCY CONSIDERATIONS	431
8.5	REFERENCES AND READINGS	434
9	ALGEBRAIC PROBLEMS	436
9.1	THE GENERAL METHOD	436
9.2	EVALUATION AND INTERPOLATION	439
9.3	THE FAST FOURIER TRANSFORM	448
9.3.1	An In-place Version of the FFT	454
9.3.2	Some Remaining Points	457
9.4	MODULAR ARITHMETIC	458
9.5	EVEN FASTER EVALUATION AND INTERPOLATION	466
9.6	REFERENCES AND READINGS	474

10 LOWER BOUND THEORY	
10.1 COMPARISON TREES	476
10.1.1 Ordered Searching	477
10.1.2 Sorting	478
10.1.3 Selection	478
10.2 ORACLES AND ADVERSARY ARGUMENTS	483
10.2.1 Merging	485
10.2.2 Largest and Second Largest	486
10.2.3 State Space Method	487
10.2.4 Selection	489
10.3 LOWER BOUNDS THROUGH REDUCTIONS	490
10.3.1 Finding the Convex Hull	493
10.3.2 Disjoint Sets Problem	494
10.3.3 On-line Median Finding	494
10.3.4 Multiplying Triangular Matrices	496
10.3.5 Inverting a Lower Triangular Matrix	496
10.3.6 Computing the Transitive Closure	497
10.4 TECHNIQUES FOR ALGEBRAIC PROBLEMS (*)	499
10.5 REFERENCES AND READINGS	503
11 \mathcal{NP}-HARD AND \mathcal{NP}-COMPLETE PROBLEMS	512
11.1 BASIC CONCEPTS	514
11.1.1 Nondeterministic Algorithms	514
11.1.2 The Classes \mathcal{NP} -hard and \mathcal{NP} -complete	515
11.2 COOK'S THEOREM (*)	523
11.3 \mathcal{NP} -HARD GRAPH PROBLEMS	527
11.3.1 Clique Decision Problem (CDP)	536
11.3.2 Node Cover Decision Problem (NCDP)	537
11.3.3 Chromatic Number Decision Problem (CNDP)	538
11.3.4 Directed Hamiltonian Cycle (DHC) (*)	539
11.3.5 Traveling Salesperson Decision Problem (TSP)	541
11.3.6 AND/OR Graph Decision Problem (AOG)	545
11.4 \mathcal{NP} -HARD SCHEDULING PROBLEMS	545
11.4.1 Scheduling Identical Processors	552
11.4.2 Flow Shop Scheduling	553
	555

11.4.3	Job Shop Scheduling	557
11.5	\mathcal{NP} -HARD CODE GENERATION PROBLEMS	559
11.5.1	Code Generation with Common Subexpressions	562
11.5.2	Implementing Parallel Assignment Instructions	565
11.6	SOME SIMPLIFIED \mathcal{NP} -HARD PROBLEMS	568
11.7	REFERENCES AND READINGS	572
11.8	ADDITIONAL EXERCISES	572
12	APPROXIMATION ALGORITHMS	576
12.1	INTRODUCTION	576
12.2	ABSOLUTE APPROXIMATIONS	580
12.2.1	Planar Graph Coloring	580
12.2.2	Maximum Programs Stored Problem	580
12.2.3	\mathcal{NP} -hard Absolute Approximations	582
12.3	ϵ -APPROXIMATIONS	584
12.3.1	Scheduling Independent Tasks	584
12.3.2	Bin Packing	588
12.3.3	\mathcal{NP} -hard ϵ -approximation Problems	591
12.4	POLYNOMIAL TIME APPROXIMATION SCHEMES	597
12.4.1	Scheduling Independent Tasks	597
12.4.2	0/1 Knapsack	599
12.5	FULLY POLYNOMIAL TIME APPROXIMATION SCHEMES	604
12.5.1	Rounding	605
12.5.2	Interval Partitioning	609
12.5.3	Separation	610
12.6	PROBABILISTICALLY GOOD ALGORITHMS (*)	615
12.7	REFERENCES AND READINGS	618
12.8	ADDITIONAL EXERCISES	618
13	PRAM ALGORITHMS	623
13.1	INTRODUCTION	623
13.2	COMPUTATIONAL MODEL	626
13.3	FUNDAMENTAL TECHNIQUES AND ALGORITHMS	632
13.3.1	Prefix Computation	633

13.3.2 List Ranking	635
13.4 SELECTION	644
13.4.1 Maximal Selection with n^2 Processors	645
13.4.2 Finding the Maximum Using n Processors	646
13.4.3 Maximal Selection Among Integers	646
13.4.4 General Selection Using n^2 Processors	649
13.4.5 A Work-Optimal Randomized Algorithm (*)	649
13.5 MERGING	653
13.5.1 A Logarithmic Time Algorithm	653
13.5.2 Odd-Even Merge	654
13.5.3 A Work-Optimal Algorithm	656
13.5.4 An $O(\log \log m)$ -Time Algorithm	658
13.6 SORTING	660
13.6.1 Odd-Even Merge Sort	660
13.6.2 An Alternative Randomized Algorithm	661
13.6.3 Preparata's Algorithm	662
13.6.4 Reischuk's Randomized Algorithm (*)	663
13.7 GRAPH PROBLEMS	667
13.7.1 An Alternative Algorithm for Transitive Closure	670
13.7.2 All-Pairs Shortest Paths	670
13.8 COMPUTING THE CONVEX HULL	671
13.9 LOWER BOUNDS	674
13.9.1 A lower bound on average-case sorting	676
13.9.2 Finding the maximum	677
13.10 REFERENCES AND READINGS	679
13.11 ADDITIONAL EXERCISES	680
14 MESH ALGORITHMS	
14.1 COMPUTATIONAL MODEL	683
14.2 PACKET ROUTING	683
14.2.1 Packet Routing on a Linear Array	685
14.2.2 A Greedy Algorithm for PPR on a Mesh	686
14.2.3 A Randomized Algorithm with Small Queues	691
14.3 FUNDAMENTAL ALGORITHMS	695
14.3.1 Broadcasting	696
14.3.2 Prefix Computation	697
14.3.3 Data Concentration	701
14.3.4 Sparse Enumeration Sort	702

14.4 SELECTION	706
14.4.1 A Randomized Algorithm for $n = p$ (*)	707
14.4.2 Randomized Selection for $n > p$ (*)	707
14.4.3 A Deterministic Algorithm For $n > p$	708
14.5 MERGING	712
14.5.1 Rank Merge on a Linear Array	713
14.5.2 Odd-Even Merge on a Linear Array	713
14.5.3 Odd-Even Merge on a Mesh	714
14.6 SORTING	716
14.6.1 Sorting on a Linear Array	716
14.6.2 Sorting on a Mesh	718
14.7 GRAPH PROBLEMS	723
14.7.1 An $n \times n$ Mesh Algorithm for Transitive Closure	725
14.7.2 All-Pairs Shortest Paths	726
14.8 COMPUTING THE CONVEX HULL	727
14.9 REFERENCES AND READINGS	732
14.10 ADDITIONAL EXERCISES	734
15 HYPERCUBE ALGORITHMS	738
15.1 COMPUTATIONAL MODEL	738
15.1.1 The Hypercube	738
15.1.2 The Butterfly Network	741
15.1.3 Embedding of Other Networks	742
15.2 PPR ROUTING	747
15.2.1 A Greedy Algorithm	747
15.2.2 A Randomized Algorithm	747
15.3 FUNDAMENTAL ALGORITHMS	751
15.3.1 Broadcasting	751
15.3.2 Prefix Computation	752
15.3.3 Data Concentration	753
15.3.4 Sparse Enumeration Sort	756
15.4 SELECTION	759
15.4.1 A Randomized Algorithm for $n = p$ (*)	759
15.4.2 Randomized Selection for $n > p$ (*)	759
15.4.3 A Deterministic Algorithm for $n > p$	760
15.5 MERGING	762
15.5.1 Odd-Even Merge	762
15.5.2 Bitonic Merge	764

15.6 SORTING 765
 15.6.1 Odd-Even Merge Sort 765
 15.6.2 Bitonic Sort 767
15.7 GRAPH PROBLEMS 768
15.8 COMPUTING THE CONVEX HULL 770
15.9 REFERENCES AND READINGS 771
15.10 ADDITIONAL EXERCISES 772

INDEX