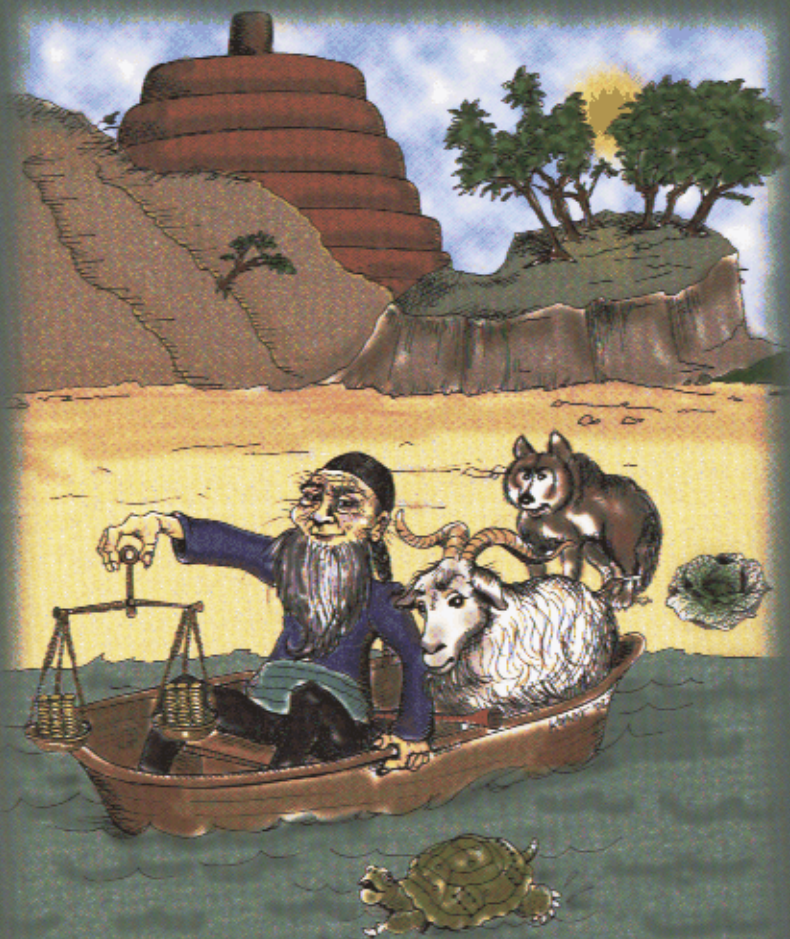


INTERNATIONAL EDITION

Introduction to

The Design & Analysis of Algorithms



Anany Levitin

Contents

Preface	xix
1 Introduction	1
1.1 Notion of Algorithm	3
Exercises 1.1	8
1.2 Fundamentals of Algorithmic Problem Solving	9
Understanding the Problem	9
Ascertaining the Capabilities of a Computational Device	11
Choosing between Exact and Approximate Problem Solving	11
Deciding on Appropriate Data Structures	12
Algorithm Design Techniques	12
Methods of Specifying an Algorithm	13
Proving an Algorithm's Correctness	13
Analyzing an Algorithm	14
Coding an Algorithm	15
Exercises 1.2	17
1.3 Important Problem Types	19
Sorting	19
Searching	20
String Processing	21
Graph Problems	21
Combinatorial Problems	22
Geometric Problems	22
Numerical Problems	23
Exercises 1.3	23

1.4 Fundamental Data Structures	26
Linear Data Structures	26
Graphs	28
Trees	32
Sets and Dictionaries	35
Exercises 1.4	37
Summary	39
2 Fundamentals of the Analysis of Algorithm Efficiency	41
2.1 Analysis Framework	42
Measuring an Input's Size	43
Units for Measuring Running Time	44
Orders of Growth	45
Worst-Case, Best-Case, and Average-Case Efficiencies	47
Recapitulation of the Analysis Framework	50
Exercises 2.1	50
2.2 Asymptotic Notations and Basic Efficiency Classes	52
Informal Introduction	52
O -notation	53
Ω -notation	54
Θ -notation	55
Useful Property Involving the Asymptotic Notations	56
Using Limits for Comparing Orders of Growth	57
Basic Efficiency Classes	58
Exercises 2.2	59
2.3 Mathematical Analysis of Nonrecursive Algorithms	61
Exercises 2.3	67
2.4 Mathematical Analysis of Recursive Algorithms	69
Exercises 2.4	76
2.5 Example: Fibonacci Numbers	78
Explicit Formula for the n th Fibonacci Number	79
Algorithms for Computing Fibonacci Numbers	81
Exercises 2.5	83

2.6 Empirical Analysis of Algorithms	84
Exercises 2.6	90
2.7 Algorithm Visualization	91
Summary	95
3 Brute Force	97
3.1 Selection Sort and Bubble Sort	98
Selection Sort	99
Bubble Sort	100
Exercises 3.1	102
3.2 Sequential Search and Brute-Force String Matching	103
Sequential Search	103
Brute-Force String Matching	104
Exercises 3.2	105
3.3 Closest-Pair and Convex-Hull Problems by Brute Force	107
Closest-Pair Problem	107
Convex-Hull Problem	108
Exercises 3.3	111
3.4 Exhaustive Search	113
Traveling Salesman Problem	113
Knapsack Problem	115
Assignment Problem	115
Exercises 3.4	118
Summary	119
4 Divide-and-Conquer	121
4.1 Mergesort	124
Exercises 4.1	126
4.2 Quicksort	127
Exercises 4.2	132
4.3 Binary Search	133
Exercises 4.3	136

4.4 Binary Tree Traversals and Related Properties	138
Exercises 4.4	140
4.5 Multiplication of Large Integers and Strassen's Matrix Multiplication	142
Multiplication of Large Integers	142
Strassen's Matrix Multiplication	144
Exercises 4.5	146
4.6 Closest-Pair and Convex-Hull Problems by Divide-and-Conquer	147
Closest-Pair Problem	147
Convex-Hull Problem	148
Exercises 4.6	151
Summary	152
5 Decrease-and-Conquer	155
5.1 Insertion Sort	158
Exercises 5.1	161
5.2 Depth-First Search and Breadth-First Search	162
Depth-First Search	163
Breadth-First Search	165
Exercises 5.2	168
5.3 Topological Sorting	170
Exercises 5.3	173
5.4 Algorithms for Generating Combinatorial Objects	175
Generating Permutations	175
Generating Subsets	177
Exercises 5.4	179
5.5 Decrease-by-a-Constant-Factor Algorithms	180
Fake-Coin Problem	180
Multiplication à la Russe	181
Josephus Problem	182
Exercises 5.5	184
5.6 Variable-Size-Decrease Algorithms	185
Computing a Median and the Selection Problem	185

Interpolation Search	187
Searching and Insertion in a Binary Search Tree	188
Exercises 5.6	189
Summary	190

6 Transform-and-Conquer	193
6.1 Presorting	194
Exercises 6.1	197
6.2 Gaussian Elimination	199
<i>LU</i> Decomposition and Other Applications	204
Computing a Matrix Inverse	205
Computing a Determinant	206
Exercises 6.2	207
6.3 Balanced Search Trees	209
AVL Trees	210
2-3 Trees	215
Exercises 6.3	217
6.4 Heaps and Heapsort	218
Notion of the Heap	218
Heapsort	223
Exercises 6.4	224
6.5 Horner's Rule and Binary Exponentiation	225
Horner's Rule	226
Binary Exponentiation	228
Exercises 6.5	231
6.6 Problem Reduction	232
Computing the Least Common Multiple	233
Counting Paths in a Graph	234
Reduction of Optimization Problems	235
Linear Programming	236
Reduction to Graph Problems	239
Exercises 6.6	240
Summary	242

7	Space and Time Tradeoffs	245
7.1	Sorting by Counting	247
	Exercises 7.1	250
7.2	Input Enhancement in String Matching	251
	Horspool's Algorithm	252
	Boyer-Moore Algorithm	255
	Exercises 7.2	259
7.3	Hashing	261
	Open Hashing (Separate Chaining)	262
	Closed Hashing (Open Addressing)	264
	Exercises 7.3	266
7.4	B-Trees	267
	Exercises 7.4	271
	Summary	272
8	Dynamic Programming	275
8.1	Computing a Binomial Coefficient	277
	Exercises 8.1	278
8.2	Warshall's and Floyd's Algorithms	280
	Warshall's Algorithm	280
	Floyd's Algorithm for the All-Pairs Shortest-Paths Problem	284
	Exercises 8.2	288
8.3	Optimal Binary Search Trees	289
	Exercises 8.3	294
8.4	The Knapsack Problem and Memory Functions	295
	Memory Functions	297
	Exercises 8.4	299
	Summary	300

9	Greedy Technique	303
9.1	Prim's Algorithm	305
	Exercises 9.1	309
9.2	Kruskal's Algorithm	311
	Disjoint Subsets and Union-Find Algorithms	314
	Exercises 9.2	318
9.3	Dijkstra's Algorithm	319
	Exercises 9.3	322
9.4	Huffman Trees	324
	Exercises 9.4	328
	Summary	329
10	Limitations of Algorithm Power	331
10.1	Lower-Bound Arguments	332
	Trivial Lower Bounds	333
	Information-Theoretic Arguments	334
	Adversary Arguments	334
	Problem Reduction	336
	Exercises 10.1	337
10.2	Decision Trees	339
	Decision Trees for Sorting Algorithms	340
	Decision Trees for Searching a Sorted Array	342
	Exercises 10.2	344
10.3	P , NP , and NP -complete Problems	345
	P and NP Problems	346
	NP -complete Problems	351
	Exercises 10.3	353
10.4	Challenges of Numerical Algorithms	356
	Exercises 10.4	363
	Summary	364

APPENDIX B

Short Tutorial on Recurrence Relations	417
Sequences and Recurrence Relations	417
Methods for Solving Recurrence Relations	418
Common Recurrence Types in Algorithm Analysis	423
Bibliography	431
Hints to Exercises	439
Index	479