



CONTENTS

1	Introduction	1
1.1.	What's the Book About?	1
1.2.	Mathematics Review	3
1.2.1.	Exponents	3
1.2.2.	Logarithms	3
1.2.3.	Series	4
1.2.4.	Modular Arithmetic	5
1.2.5.	The <i>P</i> Word	6
1.3.	A Brief Introduction to Recursion	8
	Summary	12
	Exercises	12
	References	13
2	Algorithm Analysis	15
2.1.	Mathematical Background	15
2.2.	Model	18
2.3.	What to Analyze	18
2.4.	Running Time Calculations	20
2.4.1.	A Simple Example	21
2.4.2.	General Rules	21
2.4.3.	Solutions for the Maximum Subsequence Sum Problem	24
2.4.4.	Logarithms in the Running Time	28
2.4.5.	Checking Your Analysis	33
2.4.6.	A Grain of Salt	33
	Summary	34
	Exercises	35
	References	39

3	Lists, Stacks, and Queues	41
3.1.	Abstract Data Types (ADTs)	41
3.2.	The List ADT	42
3.2.1.	Simple Array Implementation of Lists	43
3.2.2.	Linked Lists	43
3.2.3.	Programming Details	44
3.2.4.	Common Errors	49
3.2.5.	Doubly Linked Lists	51
3.2.6.	Circularly Linked Lists	52
3.2.7.	Examples	52
3.2.8.	Cursor Implementation of Linked Lists	57
3.3.	The Stack ADT	62
3.3.1.	Stack Model	62
3.3.2.	Implementation of Stacks	63
3.3.3.	Applications	71
3.4.	The Queue ADT	79
3.4.1.	Queue Model	79
3.4.2.	Array Implementation of Queues	79
3.4.3.	Applications of Queues	84
	Summary	85
	Exercises	85
4	Trees	89
4.1.	Preliminaries	89
4.1.1.	Implementation of Trees	90
4.1.2.	Tree Traversals with an Application	91
4.2.	Binary Trees	95
4.2.1.	Implementation	96
4.2.2.	Expression Trees	97
4.3.	The Search Tree ADT—Binary Search Trees	100
4.3.1.	MakeEmpty	101
4.3.2.	Find	101
4.3.3.	FindMin and FindMax	103
4.3.4.	Insert	104
4.3.5.	Delete	105
4.3.6.	Average-Case Analysis	107
4.4.	AVL Trees	110
4.4.1.	Single Rotation	112
4.4.2.	Double Rotation	115

4.5.	Splay Trees	123
4.5.1.	A Simple Idea (That Does Not Work)	124
4.5.2.	Splaying	126
4.6.	Tree Traversals (Revisited)	132
4.7.	B-Trees	133
	Summary	138
	Exercises	139
	References	146
5	Hashing	149
5.1.	General Idea	149
5.2.	Hash Function	150
5.3.	Separate Chaining	152
5.4.	Open Addressing	157
5.4.1.	Linear Probing	157
5.4.2.	Quadratic Probing	160
5.4.3.	Double Hashing	164
5.5.	Rehashing	165
5.6.	Extendible Hashing	168
	Summary	171
	Exercises	172
	References	175
6	Priority Queues (Heaps)	177
6.1.	Model	177
6.2.	Simple Implementations	178
6.3.	Binary Heap	179
6.3.1.	Structure Property	179
6.3.2.	Heap Order Property	180
6.3.3.	Basic Heap Operations	182
6.3.4.	Other Heap Operations	186
6.4.	Applications of Priority Queues	189
6.4.1.	The Selection Problem	189
6.4.2.	Event Simulation	191

6.5.	<i>d</i> -Heaps	192
6.6.	Leftist Heaps	193
6.6.1.	Leftist Heap Property	193
6.6.2.	Leftist Heap Operations	194
6.7.	Skew Heaps	200
6.8.	Binomial Queues	202
6.8.1.	Binomial Queue Structure	202
6.8.2.	Binomial Queue Operations	204
6.8.3.	Implementation of Binomial Queues	205
	Summary	212
	Exercises	212
	References	216
7	Sorting	219
7.1.	Preliminaries	219
7.2.	Insertion Sort	220
7.2.1.	The Algorithm	220
7.2.2.	Analysis of Insertion Sort	221
7.3.	A Lower Bound for Simple Sorting Algorithms	221
7.4.	Shellsort	222
7.4.1.	Worst-Case Analysis of Shellsort	224
7.5.	Heapsort	226
7.5.1.	Analysis of Heapsort	228
7.6.	Mergesort	230
7.6.1.	Analysis of Mergesort	232
7.7.	Quicksort	235
7.7.1.	Picking the Pivot	236
7.7.2.	Partitioning Strategy	237
7.7.3.	Small Arrays	240
7.7.4.	Actual Quicksort Routines	240
7.7.5.	Analysis of Quicksort	241
7.7.6.	A Linear-Expected-Time Algorithm for Selection	245
7.8.	Sorting Large Structures	247
7.9.	A General Lower Bound for Sorting	247
7.9.1.	Decision Trees	247
7.10.	Bucket Sort	250
7.11.	External Sorting	250
7.11.1.	Why We Need New Algorithms	251
7.11.2.	Model for External Sorting	251

7.11.3.	The Simple Algorithm	251
7.11.4.	Multiway Merge	253
7.11.5.	Polyphase Merge	254
7.11.6.	Replacement Selection	255
	Summary	256
	Exercises	257
	References	261
8	The Disjoint Set ADT	263
8.1.	Equivalence Relations	263
8.2.	The Dynamic Equivalence Problem	264
8.3.	Basic Data Structure	265
8.4.	Smart Union Algorithms	269
8.5.	Path Compression	271
8.6.	Worst Case for Union-by-Rank and Path Compression	273
8.6.1.	Analysis of the Union/Find Algorithm	273
8.7.	An Application	279
	Summary	279
	Exercises	280
	References	281
9	Graph Algorithms	283
9.1.	Definitions	283
9.1.1.	Representation of Graphs	284
9.2.	Topological Sort	286
9.3.	Shortest-Path Algorithms	290
9.3.1.	Unweighted Shortest Paths	291
9.3.2.	Dijkstra's Algorithm	295
9.3.3.	Graphs with Negative Edge Costs	304
9.3.4.	Acyclic Graphs	305
9.3.5.	All-Pairs Shortest Path	308
9.4.	Network Flow Problems	308
9.4.1.	A Simple Maximum-Flow Algorithm	309
9.5.	Minimum Spanning Tree	313
9.5.1.	Prim's Algorithm	314
9.5.2.	Kruskal's Algorithm	316

9.6.	Applications of Depth-First Search	319
9.6.1.	Undirected Graphs	320
9.6.2.	Biconnectivity	322
9.6.3.	Euler Circuits	326
9.6.4.	Directed Graphs	329
9.6.5.	Finding Strong Components	331
9.7.	Introduction to <i>NP</i> -Completeness	332
9.7.1.	Easy vs. Hard	333
9.7.2.	The Class <i>NP</i>	334
9.7.3.	<i>NP</i> -Complete Problems	335
	Summary	337
	Exercises	337
	References	343
10	Algorithm Design Techniques	347
10.1.	Greedy Algorithms	347
10.1.1.	A Simple Scheduling Problem	348
10.1.2.	Huffman Codes	351
10.1.3.	Approximate Bin Packing	357
10.2.	Divide and Conquer	365
10.2.1.	Running Time of Divide and Conquer Algorithms	366
10.2.2.	Closest-Points Problem	368
10.2.3.	The Selection Problem	373
10.2.4.	Theoretical Improvements for Arithmetic Problems	376
10.3.	Dynamic Programming	380
10.3.1.	Using a Table Instead of Recursion	380
10.3.2.	Ordering Matrix Multiplications	383
10.3.3.	Optimal Binary Search Tree	387
10.3.4.	All-Pairs Shortest Path	390
10.4.	Randomized Algorithms	392
10.4.1.	Random Number Generators	394
10.4.2.	Skip Lists	397
10.4.3.	Primality Testing	399
10.5.	Backtracking Algorithms	401
10.5.1.	The Turnpike Reconstruction Problem	403
10.5.2.	Games	407
	Summary	413
	Exercises	415
	References	422

11	Amortized Analysis	427
11.1.	An Unrelated Puzzle	428
11.2.	Binomial Queues	428
11.3.	Skew Heaps	433
11.4.	Fibonacci Heaps	435
11.4.1.	Cutting Nodes in Leftist Heaps	436
11.4.2.	Lazy Merging for Binomial Queues	439
11.4.3.	The Fibonacci Heap Operations	442
11.4.4.	Proof of the Time Bound	443
11.5.	Splay Trees	445
	Summary	449
	Exercises	450
	References	451
12	Advanced Data Structures and Implementation	453
12.1.	Top-Down Splay Trees	453
12.2.	Red Black Trees	457
12.2.1.	Bottom-Up Insertion	462
12.2.2.	Top-Down Red Black Trees	463
12.2.3.	Top-Down Deletion	465
12.3.	Deterministic Skip Lists	469
12.4.	AA-Trees	476
12.5.	Treaps	482
12.6.	k -d Trees	485
12.7.	Pairing Heaps	488
	Summary	494
	Exercises	495
	References	497