
Table of Contents

Preface	i
About This Book	i
Prerequisites	i
Additional References	iii
About the Exercises	iv
Steal This Book!	v
Acknowledgments	vi
Caveat Lector!	vii
 Table of Contents	 ix
 o Introduction	 1
o.1 What is an algorithm?	1
o.2 Multiplication	3

	Lattice Multiplication • Duplation and Mediation • Compass and Straight-edge	
0.3	Congressional Apportionment	8
0.4	A Bad Example	10
0.5	Describing Algorithms	11
	Specifying the Problem • Describing the Algorithm	
0.6	Analyzing Algorithms	14
	Correctness • Running Time	
	Exercises	17
1	Recursion	21
1.1	Reductions	21
1.2	Simplify and Delegate	22
1.3	Tower of Hanoi	24
1.4	Mergesort	26
	Correctness • Analysis	
1.5	Quicksort	29
	Correctness • Analysis	
1.6	The Pattern	31
1.7	Recursion Trees	31
	♥Ignoring Floors and Ceilings Is Okay, Honest	
1.8	♥Linear-Time Selection	35
	Quickselect • Good pivots • Analysis • Sanity Checking	
1.9	Fast Multiplication	40
1.10	Exponentiation	42
	Exercises	44
2	Backtracking	71
2.1	N Queens	71
2.2	Game Trees	74
2.3	Subset Sum	76
	Correctness • Analysis • Variants	
2.4	The General Pattern	79
2.5	Text Segmentation (<i>Interpunctio Verborum</i>)	80
	Index Formulation • ♥Analysis • Variants	
2.6	Longest Increasing Subsequence	86
2.7	Longest Increasing Subsequence, Take 2	89
2.8	Optimal Binary Search Trees	91
	♥Analysis	
	Exercises	93
3	Dynamic Programming	97

3.1	Mātrāvṛtta	97
	Backtracking Can Be Slow • Memo(r)ization: Remember Everything • Dynamic Programming: Fill Deliberately • Don't Remember Everything After All	
3.2	♥Aside: Even Faster Fibonacci Numbers	103
	Whoa! Not so fast!	
3.3	<i>Interpunctio Verborum Redux</i>	105
3.4	The Pattern: Smart Recursion	105
3.5	Warning: Greed is Stupid	107
3.6	Longest Increasing Subsequence	109
	First Recurrence: Is This Next? • Second Recurrence: What's Next?	
3.7	Edit Distance	111
	Recursive Structure • Recurrence • Dynamic Programming	
3.8	Subset Sum	116
3.9	Optimal Binary Search Trees	117
3.10	Dynamic Programming on Trees	120
	Exercises	123
4	Greedy Algorithms	159
4.1	Storing Files on Tape	159
4.2	Scheduling Classes	161
4.3	General Pattern	164
4.4	Huffman Codes	165
4.5	Stable Matching	170
	Some Bad Ideas • The Boston Pool and Gale-Shapley Algorithms • Running Time • Correctness • Optimality!	
	Exercises	176
5	Basic Graph Algorithms	187
5.1	Introduction and History	187
5.2	Basic Definitions	190
5.3	Representations and Examples	192
5.4	Data Structures	195
	Adjacency Lists • Adjacency Matrices • Comparison	
5.5	Whatever-First Search	199
	Analysis	
5.6	Important Variants	201
	Stack: Depth-First • Queue: Breadth-First • Priority Queue: Best-First • Disconnected Graphs • Directed Graphs	
5.7	Graph Reductions: Flood Fill	205
	Exercises	207

6	Depth-First Search	225
6.1	Preorder and Postorder	227
	Classifying Vertices and Edges	
6.2	Detecting Cycles	231
6.3	Topological Sort	232
	Implicit Topological Sort	
6.4	Memoization and Dynamic Programming	234
	Dynamic Programming in Dags	
6.5	Strong Connectivity	237
6.6	Strong Components in Linear Time	238
	Kosaraju and Sharir's Algorithm • ♥Tarjan's Algorithm	
	Exercises	244
7	Minimum Spanning Trees	257
7.1	Distinct Edge Weights	257
7.2	The Only Minimum Spanning Tree Algorithm	259
7.3	Borůvka's Algorithm	261
	This is the MST Algorithm You Want	
7.4	Jarník's ("Prim's") Algorithm	263
	♥Improving Jarník's Algorithm	
7.5	Kruskal's Algorithm	265
	Exercises	268
8	Shortest Paths	273
8.1	Shortest Path Trees	274
8.2	♥Negative Edges	274
8.3	The Only SSSP Algorithm	276
8.4	Unweighted Graphs: Breadth-First Search	278
8.5	Directed Acyclic Graphs: Depth-First Search	282
8.6	Best-First: Dijkstra's Algorithm	284
	No Negative Edges • ♥Negative Edges	
8.7	Relax ALL the Edges: Bellman-Ford	289
	Moore's Improvement • Dynamic Programming Formulation	
	Exercises	297
9	All-Pairs Shortest Paths	309
9.1	Introduction	309
9.2	Lots of Single Sources	310
9.3	Reweightings	311
9.4	Johnson's Algorithm	312
9.5	Dynamic Programming	313
9.6	Divide and Conquer	315

9.7	Funny Matrix Multiplication	316
9.8	(Kleene-Roy-)Floyd-Warshall(-Ingberman)	318
	Exercises	320
10	Maximum Flows & Minimum Cuts	327
10.1	Flows	328
10.2	Cuts	329
10.3	The Maxflow-Mincut Theorem	331
10.4	Ford and Fulkerson's augmenting-path algorithm	334
	♥ Irrational Capacities	
10.5	Combining and Decomposing Flows	336
10.6	Edmonds and Karp's Algorithms	340
	Fattest Augmenting Paths • Shortest Augmenting Paths	
10.7	Further Progress	343
	Exercises	344
11	Applications of Flows and Cuts	353
11.1	Edge-Disjoint Paths	353
11.2	Vertex Capacities and Vertex-Disjoint Paths	354
11.3	Bipartite Matching	355
11.4	Tuple Selection	357
	Exam Scheduling	
11.5	Disjoint-Path Covers	360
	Minimal Faculty Hiring	
11.6	Baseball Elimination	363
11.7	Project Selection	366
	Exercises	368
12	NP-Hardness	379
12.1	A Game You Can't Win	379
12.2	P versus NP	381
12.3	NP-hard, NP-easy, and NP-complete	382
12.4	♥ Formal Definitions (<i>HC SVNT DRACONES</i>)	384
12.5	Reductions and SAT	385
12.6	3SAT (from CIRCUITSAT)	388
12.7	Maximum Independent Set (from 3SAT)	390
12.8	The General Pattern	392
12.9	Clique and Vertex Cover (from Independent Set)	394
12.10	Graph Coloring (from 3SAT)	395
12.11	Hamiltonian Cycle	398
	From Vertex Cover • From 3SAT • Variants and Extensions	
12.12	Subset Sum (from Vertex Cover)	402

Caveat Reductor!	
12.13 Other Useful NP-hard Problems	404
12.14 Choosing the Right Problem	407
12.15 A Frivolous Real-World Example	408
12.16 ♥On Beyond Zebra	412
Polynomial Space • Exponential Time • Excelsior!	
Exercises	415
Index	442
Index of People	446
Index of Pseudocode	449
Image Credits	451
Colophon	453