

Table of Contents

Preface	ix
----------------------	-----------

Part I.

1. Algorithms Matter	3
Understand the Problem	4
Experiment if Necessary	5
Algorithms to the Rescue	8
Side Story	9
The Moral of the Story	10
References	11
2. The Mathematics of Algorithms	12
Size of a Problem Instance	12
Rate of Growth of Functions	14
Analysis in the Best, Average, and Worst Cases	18
Performance Families	22
Mix of Operations	35
Benchmark Operations	36
One Final Point	38
References	38

3. Patterns and Domains	39
Patterns: A Communication Language	39
Algorithm Pattern Format	41
Pseudocode Pattern Format	42
Design Format	43
Empirical Evaluation Format	44
Domains and Algorithms	46
Floating-Point Computations	47
Manual Memory Allocation	50
Choosing a Programming Language	53
References	54

Part II.

4. Sorting Algorithms	57
Overview	57
Insertion Sort	63
Median Sort	67
Quicksort	78
Selection Sort	85
Heap Sort	86
Counting Sort	91
Bucket Sort	93
Criteria for Choosing a Sorting Algorithm	99
References	103
5. Searching	105
Overview	105
Sequential Search	106
Binary Search	112
Hash-based Search	116
Binary Tree Search	129
6. Graph Algorithms	136
Overview	136
Depth-First Search	142
Breadth-First Search	149
Single-Source Shortest Path	153
All Pairs Shortest Path	165
Minimum Spanning Tree Algorithms	169
References	171

7. Path Finding in AI	172
Overview	172
Depth-First Search	181
Breadth-First Search	190
A*Search	194
Comparison	204
Minimax	207
NegMax	213
AlphaBeta	217
References	224
8. Network Flow Algorithms	226
Overview	226
Maximum Flow	229
Bipartite Matching	239
Reflections on Augmenting Paths	242
Minimum Cost Flow	246
Transshipment	246
Transportation	247
Assignment	248
Linear Programming	249
References	250
9. Computational Geometry	251
Overview	251
Convex Hull Scan	260
LineSweep	268
Nearest Neighbor Queries	280
Range Queries	292
References	298

Part III.

10. When All Else Fails	301
Variations on a Theme	301
Approximation Algorithms	302
Offline Algorithms	302
Parallel Algorithms	303
Randomized Algorithms	303
Algorithms That Can Be Wrong, but with Diminishing Probability	310
References	313

11. Epilogue	314
Overview	314
Principle: Know Your Data	314
Principle: Decompose the Problem into Smaller Problems	315
Principle: Choose the Right Data Structure	316
Principle: Add Storage to Increase Performance	317
Principle: If No Solution Is Evident, Construct a Search	318
Principle: If No Solution Is Evident, Reduce Your Problem to Another Problem That Has a Solution	318
Principle: Writing Algorithms Is Hard—Testing Algorithms Is Harder	319

Part IV.

Appendix: Benchmarking	323
Index	337