

CONTENTS

Preface ix

1. Fundamentals 3

- 1.1 Basic Programming Model 8
Primitive data types • Loops and conditionals • Arrays • Static methods • Recursion • APIs • Strings • Input and output • Binary search
- 1.2 Data Abstraction 64
Objects • Abstract data types • Implementing ADTs • Designing ADTs
- 1.3 Bags, Queues, and Stacks 120
APIs • Arithmetic expression evaluation • Resizing arrays • Generics • Iterators • Linked lists
- 1.4 Analysis of Algorithms 172
Running time • Computational experiments • Tilde notation • Order-of-growth classifications • Amortized analysis • Memory usage
- 1.5 Case Study: Union-Find 216
Dynamic connectivity • Quick find • Quick union • Weighted quick union

2. Sorting 243

- 2.1 Elementary Sorts 244
Rules of the game • Selection sort • Insertion sort • Shellsort
- 2.2 Mergesort 270
Abstract in-place merge • Top-down mergesort • Bottom-up mergesort • $n \lg n$ lower bound for sorting
- 2.3 Quicksort 288
In-place partitioning • Randomized quicksort • 3-way partitioning
- 2.4 Priority Queues 308
Priority queue API • Elementary implementations • Binary heap • Heapsort
- 2.5 Applications 336
Comparators • Stability • Median and order statistics

3. Searching	361
3.1 Symbol Tables	362
<i>Symbol table API • Ordered symbol table API • Dedup • Frequency counter • Sequential search • Binary search</i>	
3.2 Binary Search Trees	396
<i>Basic implementation • Order-based methods • Deletion</i>	
3.3 Balanced Search Trees	424
<i>2-3 search trees • Red–black BSTs • Deletion</i>	
3.4 Hash Tables	458
<i>Hash functions • Separate chaining • Linear probing</i>	
3.5 Applications	486
<i>Set data type • Whitelist and blacklist filters • Dictionary lookup • Inverted index • File indexing • Sparse matrix–vector multiplication</i>	

4. Graphs	515
4.1 Undirected Graphs	518
<i>Glossary • Undirected graph type • Adjacency-lists representation • Depth-first search • Breadth-first search • Connected components • Degrees of separation</i>	
4.2 Directed Graphs	566
<i>Glossary • Digraph data type • Depth-first search • Directed cycle detection • Precedence-constrained scheduling • Topological sort • Strong connectivity • Kosaraju–Sharir algorithm • Transitive closure</i>	
4.3 Minimum Spanning Trees	604
<i>Cut property • Greedy algorithm • Edge-weighted graph data type • Prim’s algorithm • Kruskal’s algorithm</i>	
4.4 Shortest Paths	638
<i>Properties of shortest paths • Edge-weighted digraph data types • Generic shortest paths algorithm • Dijkstra’s algorithm • Shortest paths in edge-weighted DAGs • Critical-path method • Bellman–Ford algorithm • Negative cycle detection • Arbitrage</i>	

(continued on next page)

5. Strings695
5.1 String Sorts	702
<i>Key-indexed counting • LSD string sort • MSD string sort • 3-way string quicksort</i>	
5.2 Tries	730
<i>String symbol table API • R-way tries • Ternary search tries • Character-based operations</i>	
5.3 Substring Search	758
<i>Brute-force algorithm • Knuth–Morris–Pratt algorithm • Boyer–Moore algorithm • Rabin–Karp fingerprint algorithm</i>	
5.4 Regular Expressions	788
<i>Describing patterns with REs • Applications • Nondeterministic finite-state automata • Simulating an NFA • Building an NFA corresponding to an RE</i>	
5.5 Data Compression	810
<i>Rules of the game • Reading and writing binary data • Limitations • Run-length coding • Huffman compression • LZW compression</i>	
6. Context853
Event-Driven Simulation	856
<i>Hard-disc model • Collision prediction • Collision resolution</i>	
B-Trees	866
<i>Cost model • Search and insert</i>	
Suffix Arrays	875
<i>Suffix sorting • Longest repeated substring • Keyword in context</i>	
Network Flow	886
<i>Maximum flow • Minimum cut • Ford–Fulkerson algorithm</i>	
Reductions	903
<i>Sorting • Shortest path • Bipartite matching • Linear programming</i>	
Intractability	910
<i>Longest-paths problem • P vs. NP • Boolean satisfiability • NP-completeness</i>	
<i>Index</i>933
<i>Algorithms and Clients</i>954