Contents

Preface vii

I TECHNIQUES

1 Introduction to Algorithms 3

1.1 Correctness and Efficiency 4
  1.1.1 Correctness 5
  1.1.2 Efficiency 9

1.2 Expressing Algorithms 9

1.3 Keeping Score 10
  1.3.1 The RAM Model of Computation 10
  1.3.2 Best, Worst, and Average-Case Complexity 11

1.4 The Big Oh Notation 13

1.5 Growth Rates 15

1.6 Logarithms 16

1.7 Modeling the Problem 18

1.8 About the War Stories 20

1.9 War Story: Psychic Modeling 21

1.10 Exercises 25

2 Data Structures and Sorting 27

2.1 Fundamental Data Types 28
  2.1.1 Containers 28
  2.1.2 Dictionaries 29
  2.1.3 Binary Search Trees 30
  2.1.4 Priority Queues 31

2.2 Specialized Data Structures 33

2.3 Sorting 33

2.4 Applications of Sorting 34
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.5</td>
<td>Approaches to Sorting</td>
<td></td>
</tr>
<tr>
<td>2.5.1</td>
<td>Data Structures</td>
<td>36</td>
</tr>
<tr>
<td>2.5.2</td>
<td>Incremental Insertion</td>
<td>36</td>
</tr>
<tr>
<td>2.5.3</td>
<td>Divide and Conquer</td>
<td>37</td>
</tr>
<tr>
<td>2.5.4</td>
<td>Randomization</td>
<td>37</td>
</tr>
<tr>
<td>2.5.5</td>
<td>Bucketing Techniques</td>
<td>38</td>
</tr>
<tr>
<td>2.6</td>
<td>War Story: Stripping Triangulations</td>
<td>39</td>
</tr>
<tr>
<td>2.7</td>
<td>War Story: Mystery of the Pyramids</td>
<td>43</td>
</tr>
<tr>
<td>2.8</td>
<td>War Story: String 'em Up</td>
<td>46</td>
</tr>
<tr>
<td>2.9</td>
<td>Exercises</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>Breaking Problems Down</td>
<td>53</td>
</tr>
<tr>
<td>3.1</td>
<td>Dynamic Programming</td>
<td>54</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Fibonacci numbers</td>
<td>54</td>
</tr>
<tr>
<td>3.1.2</td>
<td>The Partition Problem</td>
<td>56</td>
</tr>
<tr>
<td>3.1.3</td>
<td>Approximate String Matching</td>
<td>60</td>
</tr>
<tr>
<td>3.1.4</td>
<td>Longest Increasing Sequence</td>
<td>62</td>
</tr>
<tr>
<td>3.1.5</td>
<td>Minimum Weight Triangulation</td>
<td>64</td>
</tr>
<tr>
<td>3.2</td>
<td>Limitations of Dynamic Programming</td>
<td>65</td>
</tr>
<tr>
<td>3.3</td>
<td>War Story: Evolution of the Lobster</td>
<td>66</td>
</tr>
<tr>
<td>3.4</td>
<td>War Story: What’s Past Is Prolog</td>
<td>69</td>
</tr>
<tr>
<td>3.5</td>
<td>War Story: Text Compression for Bar Codes</td>
<td>72</td>
</tr>
<tr>
<td>3.6</td>
<td>Divide and Conquer</td>
<td>75</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Fast Exponentiation</td>
<td>75</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Binary Search</td>
<td>76</td>
</tr>
<tr>
<td>3.6.3</td>
<td>Square and Other Roots</td>
<td>76</td>
</tr>
<tr>
<td>3.7</td>
<td>Exercises</td>
<td>77</td>
</tr>
<tr>
<td>4</td>
<td>Graph Algorithms</td>
<td>81</td>
</tr>
<tr>
<td>4.1</td>
<td>The Friendship Graph</td>
<td>82</td>
</tr>
<tr>
<td>4.2</td>
<td>Data Structures for Graphs</td>
<td>84</td>
</tr>
<tr>
<td>4.3</td>
<td>War Story: Getting the Graph</td>
<td>86</td>
</tr>
<tr>
<td>4.4</td>
<td>Traversing a Graph</td>
<td>88</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Breadth-First Search</td>
<td>89</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Depth-First Search</td>
<td>91</td>
</tr>
<tr>
<td>4.5</td>
<td>Applications of Graph Traversal</td>
<td>92</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Connected Components</td>
<td>92</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Tree and Cycle Detection</td>
<td>93</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Two-Coloring Graphs</td>
<td>93</td>
</tr>
<tr>
<td>4.5.4</td>
<td>Topological Sorting</td>
<td>94</td>
</tr>
<tr>
<td>4.5.5</td>
<td>Articulation Vertices</td>
<td>95</td>
</tr>
</tbody>
</table>
4.6 Modeling Graph Problems 95
4.7 Minimum Spanning Trees 97
  4.7.1 Prim's Algorithm 98
  4.7.2 Kruskal's Algorithm 99
4.8 Shortest Paths 100
  4.8.1 Dijkstra's Algorithm 100
  4.8.2 All-Pairs Shortest Path 102
4.9 War Story: Nothing but Nets 102
4.10 War Story: Dialing for Documents 105
4.11 Exercises 110

5 Combinatorial Search and Heuristic Methods 115
5.1 Backtracking 116
  5.1.1 Constructing All Subsets 117
  5.1.2 Constructing All Permutations 118
  5.1.3 Constructing All Paths in a Graph 118
5.2 Search Pruning 119
5.3 Bandwidth Minimization 120
5.4 War Story: Covering Chessboards 122
5.5 Heuristic Methods 125
  5.5.1 Simulated Annealing 125
  5.5.2 Neural Networks 129
  5.5.3 Genetic Algorithms 130
5.6 War Story: Annealing Arrays 131
5.7 Parallel Algorithms 134
5.8 War Story: Going Nowhere Fast 135
5.9 Exercises 136

6 Intractable Problems and Approximations 139
6.1 Problems and Reductions 140
6.2 Simple Reductions 141
  6.2.1 Hamiltonian Cycle 142
  6.2.2 Independent Set and Vertex Cover 142
  6.2.3 Clique and Independent Set 144
6.3 Satisfiability 144
  6.3.1 The Theory of NP-Completeness 145
  6.3.2 3-Satisfiability 146
6.4 Difficult Reductions 147
  6.4.1 Integer Programming 147
  6.4.2 Vertex Cover 149
6.5 Other NP-Complete Problems 151
6.6 The Art of Proving Hardness 152
6.7 War Story: Hard Against the Clock 154
6.8 Approximation Algorithms 156
  6.8.1 Approximating Vertex Cover 157
  6.8.2 The Euclidean Traveling Salesman 158
6.9 Exercises 160

7 How to Design Algorithms 163

II RESOURCES

8 A Catalog of Algorithmic Problems 171

8.1 Data Structures 174
  8.1.1 Dictionaries 175
  8.1.2 Priority Queues 180
  8.1.3 Suffix Trees and Arrays 183
  8.1.4 Graph Data Structures 187
  8.1.5 Set Data Structures 191
  8.1.6 Kd-Trees 194

8.2 Numerical Problems 197
  8.2.1 Solving Linear Equations 199
  8.2.2 Bandwidth Reduction 202
  8.2.3 Matrix Multiplication 204
  8.2.4 Determinants and Permanents 207
  8.2.5 Constrained and Unconstrained Optimization 209
  8.2.6 Linear Programming 213
  8.2.7 Random Number Generation 217
  8.2.8 Factoring and Primality Testing 221
  8.2.9 Arbitrary-Precision Arithmetic 224
  8.2.10 Knapsack Problem 228
  8.2.11 Discrete Fourier Transform 232

8.3 Combinatorial Problems 235
  8.3.1 Sorting 236
  8.3.2 Searching 240
  8.3.3 Median and Selection 244
  8.3.4 Generating Permutations 246
  8.3.5 Generating Subsets 250
  8.3.6 Generating Partitions 253
  8.3.7 Generating Graphs 257
  8.3.8 Calendrical Calculations 261
  8.3.9 Job Scheduling 263
  8.3.10 Satisfiability 266
### 8.4 Graph Problems: Polynomial-Time
- 8.4.1 Connected Components
- 8.4.2 Topological Sorting
- 8.4.3 Minimum Spanning Tree
- 8.4.4 Shortest Path
- 8.4.5 Transitive Closure and Reduction
- 8.4.6 Matching
- 8.4.7 Eulerian Cycle / Chinese Postman
- 8.4.8 Edge and Vertex Connectivity
- 8.4.9 Network Flow
- 8.4.10 Drawing Graphs Nicely
- 8.4.11 Drawing Trees
- 8.4.12 Planarity Detection and Embedding

### 8.5 Graph Problems: Hard Problems
- 8.5.1 Clique
- 8.5.2 Independent Set
- 8.5.3 Vertex Cover
- 8.5.4 Traveling Salesman Problem
- 8.5.5 Hamiltonian Cycle
- 8.5.6 Graph Partition
- 8.5.7 Vertex Coloring
- 8.5.8 Edge Coloring
- 8.5.9 Graph Isomorphism
- 8.5.10 Steiner Tree
- 8.5.11 Feedback Edge/Vertex Set

### 8.6 Computational Geometry
- 8.6.1 Robust Geometric Primitives
- 8.6.2 Convex Hull
- 8.6.3 Triangulation
- 8.6.4 Voronoi Diagrams
- 8.6.5 Nearest Neighbor Search
- 8.6.6 Range Search
- 8.6.7 Point Location
- 8.6.8 Intersection Detection
- 8.6.9 Bin Packing
- 8.6.10 Medial-Axis Transformation
- 8.6.11 Polygon Partitioning
- 8.6.12 Simplifying Polygons
- 8.6.13 Shape Similarity
- 8.6.14 Motion Planning
- 8.6.15 Maintaining Line Arrangements
- 8.6.16 Minkowski Sum

### 8.7 Set and String Problems
- 8.7.1 Set Cover
- 8.7.2 Set Packing
9 Algorithmic Resources

9.1 Software Systems
  9.1.1 LEDA
  9.1.2 Netlib
  9.1.3 The Stanford GraphBase
  9.1.4 Combinatorica
  9.1.5 Algorithm Animations with XTango
  9.1.6 Programs from Books

9.2 Data Sources

9.3 Textbooks

9.4 On-Line Resources
  9.4.1 Literature
  9.4.2 People
  9.4.3 Software

9.5 Professional Consulting Services

Bibliography

Index