Contents

1 Introduction 1
1.1 What is Competitive Programming? 1
1.1.1 Programming Contests 2
1.1.2 Tips for Practicing 3
1.2 About This Book 3
1.3 CSES Problem Set 5
1.4 Other Resources 7

2 Programming Techniques 9
2.1 Language Features 9
2.1.1 Input and Output 10
2.1.2 Working with Numbers 12
2.1.3 Shortening Code 14
2.2 Recursive Algorithms 15
2.2.1 Generating Subsets 15
2.2.2 Generating Permutations 16
2.2.3 Backtracking 18
2.3 Bit Manipulation 20
2.3.1 Bit Operations 21
2.3.2 Representing Sets 23

3 Efficiency 27
3.1 Time Complexity 27
3.1.1 Calculation Rules 27
3.1.2 Common Time Complexities 30
3.1.3 Estimating Efficiency 31
3.1.4 Formal Definitions 32
3.2 Examples 32
3.2.1 Maximum Subarray Sum 32
3.2.2 Two Queens Problem 35

4 Sorting and Searching 37
4.1 Sorting Algorithms 37
4.1.1 Bubble Sort 38

4.1.2 Merge Sort . . . 39
4.1.3 Sorting Lower Bound . . . 40
4.1.4 Counting Sort . . . 41
4.1.5 Sorting in Practice . . . 41
4.2 Solving Problems by Sorting . . . 43
4.2.1 Sweep Line Algorithms . . . 44
4.2.2 Scheduling Events . . . 45
4.2.3 Tasks and Deadlines . . . 45
4.3 Binary Search . . . 46
4.3.1 Implementing the Search . . . 47
4.3.2 Finding Optimal Solutions . . . 48

5 Data Structures . . . 51
5.1 Dynamic Arrays . . . 51
5.1.1 Vectors . . . 52
5.1.2 Iterators and Ranges . . . 53
5.1.3 Other Structures . . . 54
5.2 Set Structures . . . 55
5.2.1 Sets and Multisets . . . 55
5.2.2 Maps . . . 57
5.2.3 Priority Queues . . . 58
5.2.4 Policy-Based Sets . . . 59
5.3 Experiments . . . 60
5.3.1 Set Versus Sorting . . . 60
5.3.2 Map Versus Array . . . 61
5.3.3 Priority Queue Versus Multiset . . . 62

6 Dynamic Programming . . . 63
6.1 Basic Concepts . . . 63
6.1.1 When Greedy Fails . . . 63
6.1.2 Finding an Optimal Solution . . . 64
6.1.3 Counting Solutions . . . 67
6.2 Further Examples . . . 68
6.2.1 Longest Increasing Subsequence . . . 69
6.2.2 Paths in a Grid . . . 70
6.2.3 Knapsack Problems . . . 71
6.2.4 From Permutations to Subsets . . . 72
6.2.5 Counting Tilings . . . 74

7 Graph Algorithms . . . 77
7.1 Basics of Graphs . . . 78
7.1.1 Graph Terminology . . . 78
7.1.2 Graph Representation . . . 80
7.2 Graph Traversal . . . 83
7.2.1 Depth-First Search . . . 83

7.2.2 Breadth-First Search . . . 85
7.2.3 Applications . . . 86
7.3 Shortest Paths . . . 87
7.3.1 Bellman–Ford Algorithm . . . 88
7.3.2 Dijkstra's Algorithm . . . 89
7.3.3 Floyd–Warshall Algorithm . . . 92
7.4 Directed Acyclic Graphs . . . 94
7.4.1 Topological Sorting . . . 94
7.4.2 Dynamic Programming . . . 96
7.5 Successor Graphs . . . 97
7.5.1 Finding Successors . . . 98
7.5.2 Cycle Detection . . . 99
7.6 Minimum Spanning Trees . . . 100
7.6.1 Kruskal's Algorithm . . . 101
7.6.2 Union-Find Structure . . . 103
7.6.3 Prim's Algorithm . . . 106

8 Algorithm Design Topics . . . 107
8.1 Bit-Parallel Algorithms . . . 107
8.1.1 Hamming Distances . . . 107
8.1.2 Counting Subgrids . . . 108
8.1.3 Reachability in Graphs . . . 110
8.2 Amortized Analysis . . . 111
8.2.1 Two Pointers Method . . . 111
8.2.2 Nearest Smaller Elements . . . 113
8.2.3 Sliding Window Minimum . . . 114
8.3 Finding Minimum Values . . . 115
8.3.1 Ternary Search . . . 115
8.3.2 Convex Functions . . . 116
8.3.3 Minimizing Sums . . . 117

9 Range Queries . . . 119
9.1 Queries on Static Arrays . . . 119
9.1.1 Sum Queries . . . 120
9.1.2 Minimum Queries . . . 121
9.2 Tree Structures . . . 122
9.2.1 Binary Indexed Trees . . . 122
9.2.2 Segment Trees . . . 125
9.2.3 Additional Techniques . . . 128

10 Tree Algorithms . . . 131
10.1 Basic Techniques . . . 131
10.1.1 Tree Traversal . . . 132
10.1.2 Calculating Diameters . . . 134
10.1.3 All Longest Paths . . . 135

10.2 Tree Queries . . . 137
10.2.1 Finding Ancestors . . . 137
10.2.2 Subtrees and Paths . . . 138
10.2.3 Lowest Common Ancestors . . . 140
10.2.4 Merging Data Structures . . . 142
10.3 Advanced Techniques . . . 144
10.3.1 Centroid Decomposition . . . 144
10.3.2 Heavy-Light Decomposition . . . 145

11 Mathematics . . . 147
11.1 Number Theory . . . 147
11.1.1 Primes and Factors . . . 148
11.1.2 Sieve of Eratosthenes . . . 150
11.1.3 Euclid's Algorithm . . . 151
11.1.4 Modular Exponentiation . . . 153
11.1.5 Euler's Theorem . . . 153
11.1.6 Solving Equations . . . 155
11.2 Combinatorics . . . 156
11.2.1 Binomial Coefficients . . . 157
11.2.2 Catalan Numbers . . . 159
11.2.3 Inclusion-Exclusion . . . 161
11.2.4 Burnside's Lemma . . . 163
11.2.5 Cayley's Formula . . . 164
11.3 Matrices . . . 164
11.3.1 Matrix Operations . . . 165
11.3.2 Linear Recurrences . . . 167
11.3.3 Graphs and Matrices . . . 169
11.3.4 Gaussian Elimination . . . 170
11.4 Probability . . . 173
11.4.1 Working with Events . . . 174
11.4.2 Random Variables . . . 175
11.4.3 Markov Chains . . . 178
11.4.4 Randomized Algorithms . . . 179
11.5 Game Theory . . . 181
11.5.1 Game States . . . 181
11.5.2 Nim Game . . . 182
11.5.3 Sprague–Grundy Theorem . . . 184

12 Advanced Graph Algorithms . . . 189
12.1 Strong Connectivity . . . 189
12.1.1 Kosaraju's Algorithm . . . 190
12.1.2 2SAT Problem . . . 192
12.2 Complete Paths . . . 193
12.2.1 Eulerian Paths . . . 194

12.2.2 Hamiltonian Paths . . . 195
12.2.3 Applications . . . 196
12.3 Maximum Flows . . . 198
12.3.1 Ford–Fulkerson Algorithm . . . 199
12.3.2 Disjoint Paths . . . 202
12.3.3 Maximum Matchings . . . 203
12.3.4 Path Covers . . . 205
12.4 Depth-First Search Trees . . . 207
12.4.1 Biconnectivity . . . 207
12.4.2 Eulerian Subgraphs . . . 209

13 Geometry . . . 211
13.1 Geometric Techniques . . . 211
13.1.1 Complex Numbers . . . 211
13.1.2 Points and Lines . . . 213
13.1.3 Polygon Area . . . 216
13.1.4 Distance Functions . . . 218
13.2 Sweep Line Algorithms . . . 220
13.2.1 Intersection Points . . . 220
13.2.2 Closest Pair Problem . . . 221
13.2.3 Convex Hull Problem . . . 224

14 String Algorithms . . . 225
14.1 Basic Topics . . . 225
14.1.1 Trie Structure . . . 226
14.1.2 Dynamic Programming . . . 227
14.2 String Hashing . . . 228
14.2.1 Polynomial Hashing . . . 228
14.2.2 Applications . . . 229
14.2.3 Collisions and Parameters . . . 230
14.3 Z-Algorithm . . . 231
14.3.1 Constructing the Z-Array . . . 232
14.3.2 Applications . . . 233
14.4 Suffix Arrays . . . 234
14.4.1 Prefix Doubling Method . . . 235
14.4.2 Finding Patterns . . . 236
14.4.3 LCP Arrays . . . 236

15 Additional Topics . . . 239
15.1 Square Root Techniques . . . 239
15.1.1 Data Structures . . . 240
15.1.2 Subalgorithms . . . 241
15.1.3 Integer Partitions . . . 243
15.1.4 Mo's Algorithm . . . 244

15.2 Segment Trees Revisited . . . 245
15.2.1 Lazy Propagation . . . 246
15.2.2 Dynamic Trees . . . 249
15.2.3 Data Structures in Nodes . . . 251
15.2.4 Two-Dimensional Trees . . . 253
15.3 Treaps . . . 253
15.3.1 Splitting and Merging . . . 253
15.3.2 Implementation . . . 255
15.3.3 Additional Techniques . . . 257
15.4 Dynamic Programming Optimization . . . 258
15.4.1 Convex Hull Trick . . . 258
15.4.2 Divide and Conquer Optimization . . . 260
15.4.3 Knuth's Optimization . . . 261
15.5 Miscellaneous . . . 262
15.5.1 Meet in the Middle . . . 263
15.5.2 Counting Subsets . . . 263
15.5.3 Parallel Binary Search . . . 265
15.5.4 Dynamic Connectivity . . . 266

Appendix A: Mathematical Background . . . 269

References . . . 277

Index . . . 279

http://www.springer.com/978-3-319-72546-8

Guide to Competitive Programming
Learning and Improving Algorithms Through Contests
Laaksonen, A.
2017, XII, 283 p. 266 illus., 96 illus. in color., Softcover
ISBN: 978-3-319-72546-8