Contents

Forewords for CP4 vii

Testimonials of CP1/2/3 xiii

Preface for CP4 xv

Authors' Profiles xxvii

Abbreviations xxix

1 Introduction 1
- 1.1 Competitive Programming . . . 1
- 1.2 The Competitions . . . 3
  - 1.2.1 International Olympiad in Informatics (IOI) . . . 3
  - 1.2.2 International Collegiate Programming Contests (ICPC) . . . 4
  - 1.2.3 Other Programming Contests . . . 6
- 1.3 Tips to be Competitive . . . 6
  - 1.3.1 Tip 1: Type Code Faster! . . . 6
  - 1.3.2 Tip 2: Quickly Identify Problem Types . . . 8
  - 1.3.3 Tip 3: Do Algorithm Analysis . . . 10
  - 1.3.4 Tip 4: Master Programming Languages . . . 15
  - 1.3.5 Tip 5: Master the Art of Testing Code . . . 18
  - 1.3.6 Tip 6: Practice and More Practice . . . 21
  - 1.3.7 Tip 7: Team Work (for ICPC) . . . 22
- 1.4 Getting Started: The Easy Problems . . . 23
  - 1.4.1 Anatomy of a Programming Contest Problem . . . 23
  - 1.4.2 Typical Input/Output Routines . . . 23
  - 1.4.3 Time to Start the Journey . . . 26
  - 1.4.4 Getting Our First Accepted (AC) Verdict . . . 27
- 1.5 Basic String Processing Skills . . . 31
- 1.6 The Ad Hoc Problems . . . 33
- 1.7 Solutions to Non-Starred Exercises . . . 41
- 1.8 Chapter Notes . . . 51

2 Data Structures and Libraries 53
- 2.1 Overview and Motivation . . . 53
- 2.2 Linear DS with Built-in Libraries . . . 55
  - 2.2.1 Array . . . 55
  - 2.2.2 Special Sorting Problems . . . 59
  - 2.2.3 Bitmask . . . 62
  - 2.2.4 Big Integer (Python & Java) . . . 66

2.2.5 Linked Data Structures . . . 69
2.2.6 Special Stack-based Problems . . . 71
2.3 Non-Linear DS with Built-in Libraries . . . 78
2.3.1 Binary Heap (Priority Queue) . . . 78
2.3.2 Hash Table . . . 81
2.3.3 Balanced Binary Search Tree (bBST) . . . 84
2.3.4 Order Statistics Tree . . . 87
2.4 DS with Our Own Libraries . . . 94
2.4.1 Graph . . . 94
2.4.2 Union-Find Disjoint Sets . . . 99
2.4.3 Fenwick (Binary Indexed) Tree . . . 104
2.4.4 Segment Tree . . . 114
2.5 Solution to Non-Starred Exercises . . . 124
2.6 Chapter Notes . . . 127

3 Problem Solving Paradigms **129**
3.1 Overview and Motivation . . . 129
3.2 Complete Search . . . 130
3.2.1 Iterative Complete Search . . . 131
3.2.2 Recursive Complete Search . . . 135
3.2.3 Complete Search Tips . . . 139
3.2.4 Complete Search in Programming Contests . . . 143
3.3 Divide and Conquer . . . 148
3.3.1 Interesting Usages of Binary Search . . . 148
3.3.2 Ternary Search . . . 152
3.3.3 Divide and Conquer in Programming Contests . . . 153
3.4 Greedy . . . 155
3.4.1 Examples . . . 155
3.4.2 Greedy Algorithm in Programming Contests . . . 161
3.5 Dynamic Programming . . . 164
3.5.1 DP Illustration . . . 164
3.5.2 Classical Examples . . . 173
3.5.3 Non-Classical Examples . . . 184
3.5.4 Dynamic Programming in Programming Contests . . . 187
3.6 Solution to Non-Starred Exercises . . . 190
3.7 Chapter Notes . . . 191

4 Graph **193**
4.1 Overview and Motivation . . . 193
4.2 Graph Traversal . . . 195
4.2.1 Overview and Motivation . . . 195
4.2.2 Depth First Search (DFS) . . . 195
4.2.3 Breadth First Search (BFS) . . . 197
4.2.4 Finding Connected Components (Undirected Graph) . . . 198
4.2.5 Flood Fill (Implicit 2D Grid Graph) . . . 199
4.2.6 Topological Sort (Directed Acyclic Graph) . . . 200
4.2.7 Bipartite Graph Check (Undirected Graph) . . . 202
4.2.8 Cycle Check (Directed Graph) . . . 203
4.2.9 Finding Articulation Points and Bridges (Undirected Graph) . . . 205
4.2.10 Finding Strongly Connected Components (Directed Graph) . . . 208

4.2.11 Graph Traversal in Programming Contests 211
4.3 Minimum Spanning Tree (MST) 215
4.3.1 Overview and Motivation 215
4.3.2 Kruskal's Algorithm 215
4.3.3 Prim's Algorithm 217
4.3.4 Other Applications 218
4.3.5 MST in Programming Contests 221
4.4 Single-Source Shortest Paths (SSSP) 223
4.4.1 Overview and Motivation 223
4.4.2 On Unweighted Graph: BFS 223
4.4.3 On Weighted Graph: Dijkstra's 227
4.4.4 On Small Graph (with Negative Cycle): Bellman-Ford 234
4.4.5 SSSP in Programming Contests 237
4.5 All-Pairs Shortest Paths (APSP) 241
4.5.1 Overview and Motivation 241
4.5.2 Floyd-Warshall Algorithm 242
4.5.3 Other Applications 244
4.5.4 APSP in Programming Contests 246
4.6 Special Graphs 249
4.6.1 Directed Acyclic Graph 249
4.6.2 Tree 255
4.6.3 Bipartite Graph 257
4.6.4 Eulerian Graph 260
4.6.5 Special Graphs in Programming Contests 263
4.7 Solution to Non-Starred Exercises 267
4.8 Chapter Notes 270

Bibliography **276**