CLUB DE PROGRAMACIÓN COMPETITIVA
DE LA FACULTAD DE INGENIERÍA

CPCFI's Syllabus

Part I
Fall Semester

0 Previous Knowledge

1. Algorithm Analysis
2. RAM Model

1 Introduction to Competitive Programming

1. Linear Data Structures [Chap. 2.2]
2. Ad Hoc Math Problems [Chap. 5.2]
3. Basic String Processing Skills [Chap. 6.2]
4. Ad Hoc String Processing Problems [Chap. 6.3]
5. Basic Geometric Objects [Chap. 7.2]

2 Non-Linear Data Structures

1. NLDS with Built-in Libraries
   - 1.1. Balanced Binary Search Tree (BST)
   - 1.2. Heap
   - 1.3. Hash Table
   - 1.4. Priority Queue
2. NLDS without Built-in Libraries
   - 2.1. Minimum Stack / Minimum Queue
   - 2.2. Sparse Table
   - 2.3. Disjoint Set Union
   - 2.4. Binary Indexed Tree (Fenwick Tree)
   - 2.5. Segment Tree
   - 2.6. Sqrt Decomposition
   - 2.7. Treap
   - 2.8. Sqrt Tree

3 Complete Search, Divide & Conquer, Greedy

1. Introduction to algorithmic heuristics
2. Complete Search [Chap. 3.2]
   - 2.1. Backtrack [Chap. 3.2.1]
3. Divide and Conquer [Chap. 3.3]
4. Greedy [Chap. 3.4]
5. More Advanced Search Techniques [Chap. 8.2]
   - 5.1. Backtracking wit Bitmask [Chap. 8.2.1]
   - 5.2. Backtracking with Heavy Prunning [Chap. 8.2.2]

4 Dynamic Programming I

1. Dynamic Programming [Chap. 3.5]
   - 1.1. Illustration [Chap. 3.5.1]
   - 1.2. Classical Problems [Chap. 3.5.2]
   - 1.3. Non-classical Problems [Chap. 3.5.3]
   - 1.4. DP in Programming Contests [Chap. 3.5.4]

5 Dynamic Programming II

1. Combinatronics [Chap. 5.4]
   - 1.1. Fibonacci Numbers [Chap. 5.4.1]
   - 1.2. Binomial Coefficients [Chap. 5.4.2]
   - 1.3. Catalan Numbers [Chap. 5.4.3]
2. Probability Theory [Chap. 5.6]
3. String Processing with Dynamic Programming [Chap. 6.5]
4. More Advanced DP Techniques [Chap. 8.3]
   - 4.1. DP with Bitmask [Chap. 8.3.1]
   - 4.2. Compilation of Common DP Parameters [Chap. 8.3.2]
   - 4.3. Handling Negative Parameter Values with Offset Technique [Chap. 8.3.3]
   - 4.4. MLE? Balanced BST as a Memo Table [Chap. 8.3.4]
   - 4.5. MLE/TLE? Use Better State Representation [Chap. 8.3.5]
   - 4.6. MLE/TLE? Drop One Parameter, Recover it From Others [Chap. 8.3.6]

Part II
Spring Semester

6 Graphs I

1. Graph Traversal
   - 1.1. Depth First Search(DFS)
   - 1.2. Breadth First Search(BFS)
   - 1.3. Finding Connected Components (Undirected Graph)
   - 1.4. Flood Fill - Labeling/Coloring the Connected Components
   - 1.5. Topological Sort (Directed Acyclic Graph)
   - 1.6. Bipartite Graph Check
   - 1.7. Graph Edges Property Check via DFS Spanning Tree
   - 1.8. Finding Articulation Points and Bridges (Undirected Graph)
   - 1.9. Finding Strongly Connected Components (Directed Graph)
2. Minimum Spanning Tree
   - 2.1. Kruskal's Algorithm
   - 2.2. Prim's Algorithm
   - 2.3. Other Applications
3. Single-Source Shortest Paths
   - 3.1. SSSP on Unweighted Graph
   - 3.2. SSSP on Weighted Graph
   - 3.3. SSSP on Graph with Negative Weight Cycle

7 Graphs II

1. All-Pairs Shortest Paths
   - 1.1. Floyd Warshall's algorithm
   - 1.2. Other Applications
2. Cycles
   - 2.1. Checking a graph for acyclicity
   - 2.2. Negative cycle in a graph
   - 2.3. Eulerian Path
3. Lowest Common Ancestor
   - 3.1. Lowest Common Ancestor
   - 3.2. Binary Lifting

3.3. Farach-Colton and Bender Algorithm
3.4. Solving RMQ by LCA
3.5. Tarjan's Offline Algorithm

4. Network Flow
   4.1. Ford Fulkerson's Algorithm
   4.2. Edmonds Karp's Algorithm
   4.3. Push-Relabel Algorithm
   4.4. Dinic's Algorithm
   4.5. MPM Algorithm
   4.6. Flow with Demands
   4.7. Minimum-cost Flow
   4.8. Assignment problem
5. Other Topics
   5.1. Edge/Vertex Connectivity
   5.2. Tree Painting
   5.3. 2-SAT
   5.4. Heavy-light Decomposition

8 Mathematics

1. Number Theory [Chap. 5.5]
   1.1. Prime Numbers [Chap. 5.5.1]
   1.2. Greatest Common Divisor & Least Common Multiple [Chap. 5.5.2]
   1.3. Factorial [Chap. 5.5.3]
   1.4. Finding Prime Factors with Optimized Trial Divisions [Chap. 5.5.4]
   1.5. Working with Prime Factors [Chap. 5.5.5]
   1.6. Functions Involving Prime Factors [Chap. 5.5.6]
   1.7. Modified Sieve [Chap. 5.5.7]
   1.8. Modulo Arithmetic [Chap. 5.5.8]
   1.9. Extended Euclid: Solving Linear Diophantine Equation [Chap. 5.5.9]
2. Game Theory [Chap. 5.8]
   2.1. Decision Tree [Chap. 5.8.1]
   2.2. Mathematical Insights to Speed-up the Solution [Chap. 5.8.2]
   2.3. Nim Game [Chap. 5.8.3]

9 String Processing and Computational Geometry

1. String Matching [Chap. 6.4]
   - 1.1. Knuth-Morris-Pratt's KMP Algorithm [Chap. 6.4.2]
   - 1.2. String Matching in a 2D Grid [Chap. 6.4.3]
2. Suffix Trie/Tree/Array [Chap. 6.6]
   - 2.1. Suffix Trie and Applications [Chap. 6.6.1]
   - 2.2. Suffix Tree [Chap. 6.6.2]
   - 2.3. Applications of SuffixTree [Chap. 6.6.3]
   - 2.4. Suffix Array [Chap. 6.6.4]
   - 2.5. Applications of Suffix Array [Chap. 6.6.5]
3. Algorithm on Polygon with Libraries [Chap. 7.3]
   - 3.1. Polygon Representation [Chap. 7.3.1]
   - 3.2. Perimeter of a Polygon [Chap. 7.3.2]
   - 3.3. Area of a Polygon [Chap. 7.3.3]
   - 3.4. Checking if a Polygon is Convex [Chap. 7.3].4
   - 3.5. Checking if a Point is Inside a Polygon [Chap. 7.3.5]
   - 3.6. Cutting Polygon with a Straight Line [Chap. 7.3.6]
   - 3.7. Finding the Convex Hull of a Set of Points [Chap. 7.3.7]

10 Advanced Topics

1. Problem Decomposition [Chap. 8.4]
   - 1.1. Two Components: Binary Search the Answer and Other [Chap. 8.4.1]
   - 1.2. Two Components: Involving 1D Static RSQ/RMQ [Chap. 8.4.2]
   - 1.3. Two Components: Graph Preprocessing and DP [Chap. 8.4.3]
   - 1.4. Two Components: Involving Graph [Chap. 8.4.4]
   - 1.5. Two Components: Involving Mathematics [Chap. 8.4.5]
   - 1.6. Two Components: Complete Search and Geometry [Chap. 8.4.6]
   - 1.7. Two Components: Involving Efficient Data Structure [Chap. 8.4.7]
   - 1.8. Three Components [Chap. 8.4.8]
2. Rare Topics [Chap. 9]

Bibliography

- Halim, S., & Halim, F. (2013). *Competitive Programming 3.* Handbook For ACM ICPC and IOI Contestants.
- *Main Page - Competitive Programming Algorithms.* (2014). CP-Algorithms. https://cp-algorithms.com/