



Syllabus

First semester

0. Previous Knowledge

- 0.1. Algorithm Analysis
- 0.2. RAM Model

1. Introduction

- 1.1. Competitive Programming [Chap. 1]
 - 1.1.1. C++ [Chap. 1.2]
 - 1.1.2. Time to Start the Journey [Chap. 1.3]
 - 1.1.3. Ad Hoc Problems [Chap. 1.4]
- 1.2. Linear Data Structures [Chap. 2.2]
- 1.3. Ad Hoc Math Problems [Chap. 5.2]
- 1.4. Basic String Processing Skills [Chap. 6.2]
- 1.5. Ad Hoc String Processing Problems [Chap. 6.3]
- 1.6. Basic Geometric Objects [Chap. 7.2]

2. Non-linear data structures

- 2.1. Non-Linear Data Structures with Built-in Libraries [Chap. 2.3]
 - 2.1.1. Balanced Binary Search Tree (BST) [Chap. 2.3.1]
 - 2.1.2. Heap [Chap. 2.3.2]
 - 2.1.3. Hash Table [Chap. 2.3.3]
- 2.2. Data Structures without Libraries [Chap. 2.4]
 - 2.2.1. Graph [Chap. 2.4.1]
 - 2.2.2. Union-Find Disjoint Sets [Chap. 2.4.2]
 - 2.2.3. Segment Tree [Chap. 2.4.3]
 - 2.2.4. Binary Indexed Tree (Fenwick Tree) [Chap. 2.4.5]

3. Complete search, divide & conquer, greedy

- 3.1. Introduction to algorithmic heuristics
- 3.2. Complete Search [Chap. 3.2]
 - 3.2.1. Backtrack [Chap. 3.2.1]
- 3.3. Divide and Conquer [Chap. 3.3]
- 3.4. Greedy [Chap. 3.4]



CLUB DE PROGRAMACIÓN COMPETITIVA DE LA FACULTAD DE INGENIERÍA



- 3.5. More Advanced Search Techniques [Chap. 8.2]
 - 3.5.1. Backtracking with Bitmask [Chap. 8.2.1]
 - 3.5.2. Backtracking with Heavy Pruning [Chap. 8.2.2]
- 4. Dynamic programming 1 (basic ideas)
 - 4.1. Dynamic Programming [Chap. 3.5]
 - 4.1.1. Illustration [Chap. 3.5.1]
 - 4.1.2. Classical Problems [Chap. 3.5.2]
 - 4.1.3. Non-classical Problems [Chap. 3.5.3]
 - 4.1.4. DP in Programming Contests [Chap. 3.5.4]
- 5. Dynamic programming 2 (more techniques)
 - 5.1. Combinatorics [Chap. 5.4]
 - 5.1.1. Fibonacci Numbers [Chap. 5.4.1]
 - 5.1.2. Binomial Coefficients [Chap. 5.4.2]
 - 5.1.3. Catalan Numbers [Chap. 5.4.3]
 - 5.2. Probability Theory [Chap. 5.6]
 - 5.3. String Processing with Dynamic Programming [Chap. 6.5]
 - 5.4. More Advanced DP Techniques [Chap. 8.3]
 - 5.4.1. DP with Bitmask [Chap. 8.3.1]
 - 5.4.2. Compilation of Common DP Parameters [Chap. 8.3.2]
 - 5.4.3. Handling Negative Parameter Values with Offset Technique [Chap. 8.3.3]
 - 5.4.4. MLE? Balanced BST as a Memo Table [Chap. 8.3.4]
 - 5.4.5. MLE/TLE? Use Better State Representation [Chap. 8.3.5]
 - 5.4.6. MLE/TLE? Drop One Parameter, Recover it From Others [Chap. 8.3.6]

Second semester

- 6. Graph 1
 - 6.1. Graph Traversal [Chap. 4.2]
 - 6.1.1.** Depth First Search (DFS) [Chap. 4.2.1]
 - 6.1.2.** Breadth First Search (BFS) [Chap. 4.2.2]
 - 6.1.3.** Finding Connected Components (Undirected Graph) [Chap. 4.2.3]
 - 6.1.4. Flood Fill - Labeling/Coloring the Connected Components [Chap. 4.2.4]
 - 6.1.5.** Topological Sort (Directed Acyclic Graph) [Chap. 4.2.5]
 - 6.1.6.** Bipartite Graph Check [Chap. 4.2.6]
 - 6.1.7. Graph Edges Property Check via DFS Spanning Tree [Chap. 4.2.7]
 - 6.1.8.** Finding Articulation Points and Bridges (Undirected Graph) [Chap. 4.2.8]
 - 6.1.9.** Finding Strongly Connected Components (Directed Graph) [Chap. 4.2.9]
 - 6.2. Minimum Spanning Tree [Chap. 4.3]
 - 6.2.1.** Kruskal's Algorithm [Chap. 4.3.2]
 - 6.2.2.** Prim's Algorithm [Chap. 4.3.3]



CLUB DE PROGRAMACIÓN COMPETITIVA DE LA FACULTAD DE INGENIERÍA



- 6.2.3.** Other Applications [Chap. 4.3.4]
 - 6.3. Single-Source Shortest Paths [Chap. 4.4]
 - 6.3.1.** SSSP on Unweighted Graph [Chap. 4.4.2]
 - 6.3.2.** SSSP on Weighted Graph [Chap. 4.4.3]
 - 6.3.3.** SSSP on Graph with Negative Weight Cycle [Chap. 4.4.4]
- 7. Graph 2
 - 7.1. All-Pairs Shortest Paths [Chap. 4.5]
 - 7.1.1.** Explanation of Floyd Warshall's DP Solution [Chap. 4.5.2]
 - 7.1.2.** Other Applications [Chap. 4.5.3]
 - 7.2. Network Flow [Chap. 4.6]
 - 7.2.1.** Ford Fulkerson's Method [Chap. 4.6.2]
 - 7.2.2.** Edmonds Karp's Algorithm [Chap. 4.6.3]
 - 7.2.3. Flow Graph Modeling - Part 1 [Chap. 4.6.4]
 - 7.2.4. Flow Graph Modeling - Part 2 [Chap. 4.6.5]
 - 7.3. Special Graphs [Chap. 4.7]
 - 7.3.1. Directed Acyclic Graph [Chap. 4.7.1]
 - 7.3.2. Tree [Chap. 4.7.2]
 - 7.3.3. Eulerian Graph [Chap. 4.7.3]
 - 7.3.4. Bipartite Graph [Chap. 4.7.4]
- 8. Mathematics
 - 8.1. Number Theory [Chap. 5.5]
 - 8.1.1. Prime Numbers [Chap. 5.5.1]
 - 8.1.2. Greatest Common Divisor & Least Common Multiple [Chap. 5.5.2]
 - 8.1.3. Factorial [Chap. 5.5.3]
 - 8.1.4. Finding Prime Factors with Optimized Trial Divisions [Chap. 5.5.4]
 - 8.1.5. Working with Prime Factors [Chap. 5.5.5]
 - 8.1.6. Functions Involving Prime Factors [Chap. 5.5.6]
 - 8.1.7. Modified Sieve [Chap. 5.5.7]
 - 8.1.8. Modulo Arithmetic [Chap. 5.5.8]
 - 8.1.9. Extended Euclid: Solving Linear Diophantine Equation [Chap. 5.5.9]
 - 8.2. Game Theory [Chap. 5.8]
 - 8.2.1. Decision Tree [Chap. 5.8.1]
 - 8.2.2. Mathematical Insights to Speed-up the Solution [Chap. 5.8.2]
 - 8.2.3. Nim Game [Chap. 5.8.3]
- 9. String Processing and Computational Geometry
 - 9.1. String Matching [Chap. 6.4]
 - 9.1.1. Knuth-Morris-Pratt's KMP Algorithm [Chap. 6.4.2]
 - 9.1.2. String Matching in a 2D Grid [Chap. 6.4.3]



**CLUB DE PROGRAMACIÓN COMPETITIVA
DE LA FACULTAD DE INGENIERÍA**



-
- 9.2. Suffix Trie/Tree/Array [Chap. 6.6]
 - 9.2.1. Suffix Trie and Applications [Chap. 6.6.1]
 - 9.2.2. Suffix Tree [Chap. 6.6.2]
 - 9.2.3. Applications of SuffixTree [Chap. 6.6.3]
 - 9.2.4. Suffix Array [Chap. 6.6.4]
 - 9.2.5. Applications of Suffix Array [Chap. 6.6.5]
 - 9.3. Algorithm on Polygon with Libraries [Chap. 7.3]
 - 9.3.1. Polygon Representation [Chap. 7.3.1]
 - 9.3.2. Perimeter of a Polygon [Chap. 7.3.2]
 - 9.3.3. Area of a Polygon [Chap. 7.3.3]
 - 9.3.4. Checking if a Polygon is Convex [Chap. 7.3].4
 - 9.3.5. Checking if a Point is Inside a Polygon [Chap. 7.3.5]
 - 9.3.6. Cutting Polygon with a Straight Line [Chap. 7.3.6]
 - 9.3.7. Finding the Convex Hull of a Set of Points [Chap. 7.3.7]
 - 10. Advanced topics
 - 10.1. Problem Decomposition [Chap. 8.4]
 - 10.1.1. Two Components: Binary Search the Answer and Other [Chap. 8.4.1]
 - 10.1.2. Two Components: Involving 1D Static RSQ/RMQ [Chap. 8.4.2]
 - 10.1.3. Two Components: Graph Preprocessing and DP [Chap. 8.4.3]
 - 10.1.4. Two Components: Involving Graph [Chap. 8.4.4]
 - 10.1.5. Two Components: Involving Mathematics [Chap. 8.4.5]
 - 10.1.6. Two Components: Complete Search and Geometry [Chap. 8.4.6]
 - 10.1.7. Two Components: Involving Efficient Data Structure [Chap. 8.4.7]
 - 10.1.8. Three Components [Chap. 8.4.8]
 - 10.2. Rare Topics [Chap. 9]