Competitive Programming Learning Path

Contributers

Vedansh Priyadarshi
vedansh.priyadarshi@gmail.com
Gagandeep Singh
czgdp1807@gmail.com

An open initiative by
CodeZoned

Abstract

The thing that is unique about this document is that this will contain everything required for competitive programming. People generally have confusion about where to start. We are completely self learned. So, we know where students generally face issues during beginning. This document will cover every topics required in competitive programming. We only assume that you know basic mathematics. The things will eventually get easy as you will move forward in your learning journey. People generally says everything is available on the web for free. Obviously, this is true. But, you need to know form where to learn. We will fulfill that gap through this document. Just follow the instructions step by step. We will also solve lot of good problems. We will also provide you some books at last, in case you want to delve deep.

Is this document for you?

Are you interested in problem solving *or* coding *or* competitions *or* mathematics?
If your answer is *yes*, get ready to be "**CodeZoned**". Make sure to contribute on our Github repository. You can at least star our document *or* make your first pull request through our repository.

If you are senior, experienced in competitive programming, we urge you to help younger budding programmers by contributing to this document. You can also send your suggestions directly to us anytime.

Disclaimer

All the documents, books, courses and links provided in this document are freely available on the web. We are just helping people find them easily.

Introduction

- What is Competitive Programming?
- List of Programming Contests.
- Which programming language is best? See Google Code Jam stats here. Let me tell you, C++ is the best language for competitive programming. Stick with it. JAVA is also good. Some people say language has no effect on code. This depends on person to person.
- Advantages of C++ in competitive programming: See the topic Python v/s C in the "LinksPDF.pdf" file in our repository.
- Learn C++ from Bucky of The Boston School. Solve basic problems on Codeabbey. Now if you have completed this step. Solve some more problems on Project Euler, not all. Now you are ready to go.

Programming Techniques

- Recursive algorithms (video) (topcoder) (problems) (more problems) (backtracking and some problems)
- Bit Manipulation (read) (solve) (tricks) (book)

Time Complexity

- BigO Notation (read) (advanced read) (Khan Academy) (cheat sheet) (misconceptions)
- Maximum subarray, N queens problem

Sorting and Searching

- Sorting (visualize) (intution) (read) (Sorting in C++) (Implementation) (Problems) Read only Bubble, Merge, Counting sort and sorting in $O(n \log n)$ time.
- Searching (visualize) (read and solve) (searching in C++)
- An amazing playlist of searching and sorting.
- Sweep Line algorithm (topcoder)
- Event scheduling problem (wiki) (video)
- Tasks and Deadlines problem (read)

Data Structures

- Dynamic Arrays (vectors read and watch) (iterators and ranges) (stacks, queues and deques watch and pdf)
- Set Structure (set and unordered_set difference, read, watch) (maps read and watch) (priority queue read and watch) (policy based structures read)
- See BigO Cheatsheet for data structures comparison.
- Visualisation of different data structures.

Dynamic Programming

- Highly recommended watch (1 and 2) and read .
- Intution (read all parts).
- Longest increasing subsequence problem (watch).
- Paths on a grid problem (watch).
- Knapsack Problem (watch this or this)
- Subset sum problem (watch)
- Watch this full playlist by Geeksforgeeks.

Graph Algorithms

- Basics of graph. Watch here or here or from Humblefool's mycodeschool(part basics and properties[highly recommended])
- Graph representation using edge list, adjacency matrix, adjacency lists and incidence matrix.
- Graph Transversal (BFS & DFS watch, visualize and implementation[BFS & DFS]). Problems on BFS & DFS.
- Strongly Connected Components
- Biconnectivity, Tarzan's algorithm(visualization) for finding bridges.
- Dijkstra's Algorithm (watch, visualize and implement [example])
- Bellman Ford algorithm (watch, visualize with example)
- Floyd–Warshall Algorithm (watch, visualize (another) and implement and on undirected path.
- Directed Acyclic Graphs (Topological Sort and implementation) Dynamic Programming for shortest path.
- Minimum Spanning Trees (watch). Prim and Kruskal Visualization.

- Euler’s tour algorithm.
- Maximum flow using Ford Fulkerson Method.

Algorithms Design Topics

- Bit-Parallel Algorithms (Hamming Distances), (Counting Subgrids)
- Amortized Analysis
- Ternary Search (read and implement)

Tree Algorithms

- Introduction (mycodeschool)
- Tree transversal (Binary tree transversal) (DP) (wiki)
- Diameter of the binary tree (in *O*(n) time) (watch)
- Lowest common ancestor in tree on *O*(1). (watch)
- Centroid decomposition (watch) (read) (problems)
- Heavy light decomposition (advanced) (read)

Mathematics

- Read whole book ‘Elementary Number Theory with Programming’.
- Watch this if you have less time. A video by O’Reilly.
- Game theory intro. NIM game. Spragur-Grundy theorem.(watch)

Advanced Graph Algorithms

- Strong Connectivity - Kosaraju’s Algorithm (wiki), 2SAT Problem (watch approx. 2hr)
- Complete Paths - Eulerian Paths (basics) (wiki), Hamiltonian Path (basics) (wiki), Applications - De Bruijn sequence(watch), Knight’s tour(numberphile)
- Maximum Flows - (MIT) (Ford–Fulkerson Algorithm)

Computational Geometry

- Graham Scan algorithm (watch) (read) (visualize)
- Online construction of 3D convex hull. (read)
- Bentley Ottomann algorithm (read)
- Geometric sweep algorithms (watch)
- Suggested Book

String Algorithms

- Knuth Morris Pratt (watch) (visualize) (read)
- Aho Corasick (watch) (MIT) (visualize)
- Suffix Arrays (full playlist)
- Suffix Trees (intro) (Ukkenon’s algorithm) (Tandem’s Repeats)

Books Recommendation

Algorithm Design by Jon Kleinberg, Eva Tardos
[Anany Levitin, Maria Levitin]Algorithmic Puzzles
[Halim] Competitive programming 3
Algorithms and programming problems and solutions
Schaums discrete maths
Elementary Number Theory with Programming
Computational Geometry Algorithms and Applications
[Ian Parberry]Problems on algorithms

Click the book and download. Again note that, we found everything on the internet and just sharing with you.