GUIDE TO COMPETITIVE PROGRAMMING:

-by Striver (Software Engineer(Media.net), ex-Amazon, Candidate Master, 6*, Educator at Unacademy)

It's advice to switch to CPP to avoid TLE issues at Codeforces and some platforms.. Learn C++ STL from codeverse or takeUforward channel ..

Follow CP sheet for practice problems:

CP SHEET: https://bit.lv/tuf CPList

SDE SHEET: https://bit.ly/takeUforward SDE

- 1.Pattern printing problems (Sourabh Shukla Videos on Youtube)
- 2.time complexity analysis (Don't stress to much, keep participating in contests, you will keep learning)
- 3.linear search and circular array representation (any article, no need to practice)
- 4.palindrome and other numbers(perfect, Armstrong) for basic number problems
- 5. Simple Hashing Problem(frequency counting and stuffs)
- 6.Prefix Sum Problems(1D and 2D) {CP Sheet has problems}
- 7.Sliding window technique(CP Sheet has problems)

Basics of number theory

- 1.Binary Search (TakeUforward Playlist and CP Sheet practice problems)
- 2. GCD of 2 numbers in logarithmic time(Euclidean and Extended Euclidean Algorithm) (CodeNCode)
- 3.linear Diphantine Equation (CodeNCode)
- 4. Checking prime in sqrt(n) complexity (Learn from Codeverse Channel)
- 5. Sieve of Eratosthenes(Learn from Codeverse Channel)
- 6.Segmented Sieve (Learn from Codeverse Channel)
- 7. Finding prime factorisation of a number in logn per querry (Learn from Codeverse Channel)
- 8. Euler Totent Function (CodeNcode)
- 9.Fermat Little Theorum (CodeNCode)

Tougher version of number theory (All from CodeNCode)

- 1.Finding xⁿ in log(n)
- 2. Modular Arithmetic
- 3. Modular Inverse of a number
- 3. Modular Exponentiation
- 4. Chinese Remainder Theorum
- 5. Factorial Modulo Mod
- 6. Finding nCr and nPr for gueries (constant time)
- 7.Inclusion Exclusion Principle (combinatorics problems)
- 1.learn about basic sorting algorithms (bubble, selection, insertion)
- 2. do problems which are constructive and have a lot of swapping terms in it.

- 3.solve problems related to two pointer approach. (TakeUforward playlist)
- 4.Bit manipulation(left shift, right shift, xor, or, and, set bit, MSB, LSB etc..) (Codeverse)
- 5. Power set of a given array or string using BIT (TakeUforward has a lecture)
- 6.Number of subarrays with XOR as zero(not an algorithm but a must do problem) (TakeUforward)
- 7. Problems related to greedy algorithm (TakeUforward Playlist)
- 8.Kadane's algorithm (TakeUforward) and problems related to them

Time to learn recursion (Codeverse 3 masterclasses by Striver and TakeUforward Playlist)

After Recursion:

- 1. Meet in the middle algorithm and problems related to it. (Follow CSES for problems)
- 2.Divide and conquer problems{highly recommended to use codeforces only for this}
- 3. Next greater element and next smaller element using stack (TakeUforward)
- 4. problems related to parenthesis.
- 5.largest rectangular area in histogram. (concept is used in a lot of problems) (TakeUforward)
- 6.Problems related to Heap(Priority Queue) {although this gets under the greedy category but by priority queue will help you learn an inbuilt stl)

String algorithms: (Since these are advanced topics, you can read them from where you feel comfortable)

- 1. Rolling Hash Function on strings {cpalgorithms has a wonderful article written on it) {Spoj or codeforces}
- 2. Rabin Karp Algorithm (cpalgorithms has a wonderful blog on it)
- 3. Prefix Function (cpalgorithms)
- 4.KMP Algorithm
- 5.Z-function
- 6.Manchers' Algorithm (once you have wrapped up the above algorithms, solve a bunch of problems(25-30) on them from different platforms.)

Tree Algorithm:

- 1.Tree/Graph representation
- 2.DFS/BFS Traversal in Graph/Tree
- 3.Basic stuffs(diameter of tree, height of tree, level of tree)
- 4.Euler Tour of Tree(Learn and solve problems)
- 5. Finding LCA using Euler Tour{efficient solution uses segment trees)
- 6. Finding LCA using Binary Lifting. 7. Distance between two nodes.
- 8.Subtree Problems. (SPOJ is highly recommended for trees and codeforces D and E problems also)

Graphs: (TakeUforward Graph series)

- 1. Connected Components.
- 2. Topological Sort.
- 3. Cycle Detection in Graph
- 4. Bipartite Check in graph
- 5.SCC using Kosaraju's algorithm
- 6.Dijkstra's Algorithm

- 7.Bellman Ford Algorithm
- 8.Bridges In graphs
- 9. Articulation Point in a graph
- 10. Minimum Spanning Tree using Kruskal's Algo
- 11.Prim's Algorithm
- 12.0/1 BFS(a big saviour)
- 13.Learn Finding Bridges Online(cpalgorithms)

Dynamic Programming:

- 0. Solve all Standard problems on DP from GFG.. (Refer Aditya Verma youtube channel for learning standard DP)
- 1. Solve the AtCoder Educational Contests on Dynamic Programming. (all 26)
- 2. Solve problems from SPOJ(highly recommended, since it doesn't involve any other algorithms)
- 3. Google dynamic programming practice problem codeforces, u'll get a wonderful blog with a lot of problems on it.
- 4.Understand how we write recurrence for Digit DP(codeforces blog)(digit dynamic progg) and solve problems
- 5. read about DP with Bitmasks and solve problems(hackerearth blog)
- 6.DP on trees(gfg articles,rachit jains video)
- 7.SOS DP(cpalgorithm blog)
- 1.Disjoint Set(cpalgorithms)
- 2.Offline Queries using Disjoint Set
- 3. Kruskal's Algorithm using disjoint set

Range Query Algos:

- 1.Sparse Table(not that imp)
- 2. Fenwick Tree and Binary Lifting on Fenwick Tree(read about range update trick also)
- 3.problems on fenwick tree
- 4.Matrix Exponentiation(problems)
- 5. Sqrt Decomposition Technique
- 6. Update and Querry Operations
- 7.Mo's algorithm
- 8.Mo's algorithm on trees
- 9. Segment Tree(a must)(Range Queries and point Updates)
- 10.Lazy propagation on segment trees

Some optional and rare ones:

- 1.Sprague-Grundy Theorum
- 2.Flows and Related Problems
- 3. Heavy Light Decomposition
- 4.Convex Hull Algorithm
- 5.FFT/NTT