

COMPETITIVE PROGRAMMING ROADMAP

BASIC

First learn basic syntax of C++ and entire STL.

- 1: Pattern printing problems & Famous Array problems
- 2: Analysis of time complexity
- 3: Linear Search problems
- 4: Circular array using simple array
- 5: Palindrome, Perfect, Armstrong & other number problems
- 6: Simple Hashing problems (frequency counting and stuffs)
- 7: Prefix Sum Problems 1D/2D {codeforces}
- 8: Sliding window technique (2/5 Contests)

INTERMEDIATE

- 1: Binary Search problems (2/5 Contests)
- 2: Find GCD of 2 numbers in $\log N$ (Euclidean and Extended euclidean algo)
- 3: Linear Diophantine Equation
- 4: Checking Prime in \sqrt{n} complexity
- 5: Sieve of Eratosthenes (Very imp to perform query probs on the prime)
- 6: Segmented Sieve
- 7: Finding the prime factorization of a number in $\log n$ per query
- 8: Euler Totient function
- 9: Fermat little theorem
- 10: Wilson's theorem

(GFG & Hackerearth articles for problem 8,9,10)

NUMBER THEORY

- 1: Finding x^n in $\log N$ time
- 2: Modular Arithmetic
- 3: Module Inverse of a number
- 4: Modular Exponentiation
- 4: Chines remainder theorem
- 5: Factorial Modulo Mod
- 6: Finding nCr & nPr for queries (Constant Time)
- 7: Inclusion Exclusion principle (combinatorics problems) {hackerearth has wonderful bunch} {codeforces, don't solve problem with dp tag}

SOME ADVANCED

- 1: Learn about basic sorting Algorithms (Bubble, Selection, Insertion)
 - 2: Do problems on constructive and having swap terms in it
 - 3: Solve problem related to two pointer approach
 - 4: Bit Manipulation problems (Left shift, Right shift, XOR, NOT, AND, Set bit, MSB LSB etc.)
{Hackerearth has good tuts}
 - 5: Power set of a given array or string using BIT
 - 6: Number of subarrays with XOR as ZERO (Not algorithm, but a must do problem)
 - 7: Greedy algorithm 'Tag'
 - 8: Kadane's Algorithms and problem related to them (Under Greedy)
 - 9: Job sequencing and activity selection problem (Under Greedy)
- (After 8 & 9, Solve problems with greedy tag on codeforces)

RECURSION

- 1: Simple recursion problem like finding factorial, sum of digits etc.

- 2: Implement Binary search using recursion
- 3: Implement modular exponentiation using recursion
- 4: Solve recursion problem like finding subset with given sum and other problems to get a strong grip

(Recursion & Backtracking{below Ad.2} refer to Leetcode and GFG's practice)

ADVANCED

- 1: Learn Merge Sort & Quick sort algorithms (Solve problems)
- 2: Do backtracking problems like Sudoku and N-Queen problem (Help in DP path problems)
- 3: Meet in the middle algorithm and problems related to it
- 4: Divide & Conquer problems on {Codeforces- Highly recommend}
- 5: Find next greater / next smaller element using stack
- 6: Problems related to parenthesis (referring to point no. 5, It's eg. of that)
- 7: Largest rectangular area in Histogram (Concept used in lot of problems)
- 8: Problems related to Heap (Solve using Priority Queue) (Do with Inbuilt STL Library)

Practice Hard on above problems

MORE ADVANCED (1-4 HR TO SOLVE)

- 1: Hashing on strings, understand when collision happens {cpalgorithm site, SPOJ, or Codeforces}
- 2: Rabin karp algorithm (Uses Hashing on Strings)
- 3: Prefix function
- 4: KMP Algorithm
- 5: Z-Function
- 6: Manacher's Algorithm

(Solve bunch of problem on above topic (25-30))

TREES

- 1: Tree / Graph representation (How to represent using adjacency matrix or vector)
- 2: DFS/BFS traversal in tree /graph
- 3: Diameter of a tree/Height & Level of Tree (BASIC STUFF)
- 4: Euler Tour of tree
- 5: Finding LCA using Euler Tour / Binary Lifting (Efficient solution uses segment trees)
- 6: Distance between two nodes
- 7: Subtree Problems

(SPOJ is highly recommended, Codeforces D & E problems)

GRAPH

- 1: Connected Components
- 2: Topological sort
- 3: Cyclic detection in graph
- 4: Bipartite check in graph
- 5: SCC using Kosaraju's Algorithm
- 6: Dijkstra's Algorithm
- 7: Bellman Ford Algorithm
- 8: Floyd Warshall algorithm (Solve more problems on above topic - Hackerearth/Codeforces/SPOJ)
- 9: Bridge in Graph
- 10: Articulation point in graph
- 11: Minimum spanning tree & Kruskal's Algorithm

12: Prim's Algorithm

13: 0/1 BFS in linear time {cpalgorithms site}

14: Finding bridges online {cpalgorithms site}

DYNAMIC PROGRAMMING (Avoid videos with Iterative DP)

1: Start with Recursion & Memoization with strong knowledge {GFG}

2: LCS, Knapsack problem solve (YouTube- Take u forward)

3: Solve AtCoder Educational contest on DP 26/26 solve

4: Solve DP problem from (Specifically SPOJ) then Codeforces

5: Understand how we write recurrence for Digit DP (Codeforces blog)

6: Read DP with bitmasks and solve on hackerearth

7: DP in trees (Rachit Jain video)

8: SOS DP (cpalgorithms site)

9: Practice More on above topic

MORE

1: Disjoint Set(Using all optimizations) {cpalgorithms site}

2: Offline Queries using Disjoint Set {Colorful array problems from SPOJ}

3: Kruskal's Algorithm using Disjoint Set

4: Sparse Table (Not Imp because with Seg. Tree we can do all stuff)

5: Fenwick Tree (Read about Range Update Trick also) (YT- TakeuForward)

6: Binary Lifting on fenwick tree (More Solve problem)

AND MORE

1: Matrix Exponentiation

2: Sqrt Decomposition {GFG/cpalgorithms/Codeforces} (When Seg Tree is hard to implement)

3: Update and query operations

4: Mo's Algorithm (Solve powerful array from Codeforces blog)

- 5: Mo's Algorithm on Trees (Not much IMP)
- 6: Segment Tree (Most Imp topic - Range queries and point updates)
- 7: Lazy propagation on segment tree (Range updates)

AT LAST (For Long Challenge Problems)

- 1: Sprague-Grundy Theorem {Gaurav Sen- YT}
- 2: Flows and related problem {cpalgorithms site}
- 3: Heavy light decomposition {by Anudeep}
- 4: Convex Hull Algorithm {codeforces}
- 5: FFT/NTT {Can be learned with any of the mathematical section}

NOTE:

This will help you to solve the problems till E-Level 😊

Participate in Contest 😊