

Follow the below Ladder

1.Basic-----

- 1: Pattern printing problems(ApnaCollege)
- 2: Circular array using simple array(gfg)
- 3: Palindrome, Perfect number, armstrong(gfg)
- 4: Simple Hashing problems, frequency count(gfg)
- 5: Prefix Sum Problems 1D/2D -solve problems
- 6: Sliding window technique(1/5)(Aditya)

2.Intermediate-----

0. Linear search Problems (1/5)
- 1: Binary Search problems (2/5)
- 2: Find GCD of 2 numbers in LogN (Euclidean and Extended euclideanAlgo)
- 3: Linear Diophantine Equation
- 4: Prime in \sqrt{n} complexity
- 5: Sieve of Eratosthenes
- 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 - HE

3.Number Theory-----

- 1: Finding x^n in LogN
- 2: Modular Arithmetic
- 3: Module Inverse of a number

- 4: Chinese remainder theorem
- 5: Factorial Modulo Mod
- 6: Finding nCr & nPr in queries
- 7: Inclusion Exclusion principle -HE
- 8: Modular Exponentiation

4. Some Advanced -----

- 1: Learn about basic sorting Algorithms (Bubble, Selection, Insertion)
- 2: Constructive and having swap terms in it
- 3: Solve problems related to Two Pointer Approach
- 4: Bit Manipulation problems (Left shift, Right shift, Set bit, MSB LSB etc) (Hackerearth as good tuts)
- 5: Power set of a given array or string using BIT
- 6: Number of subarray with XOR as ZERO (Not algorithm, but a must do problem)
- 7: Problems related to Greedy Algorithms Tag - CF
- 8: Kadane's Algorithms and problem related to them
- 9: Job sequence and activity selection problem

5. Recursion (All Basic)-----

- 1: Recursion problems like finding factorial (pepcoding & Aditya Verma)
- 2: Implement Binary search using recursion
- 3: Implement modular exponent
- 4: Solve recursion problem like finding subset with given sum and other problems

6.Advanced Recursion-----

- 1: Learn Merge Sort & Quick sort algorithms - count inversion
- 2: Do backtracking problems like Sudoku and N-Queen problem (Help in DP path problems)
- 3: Meet in the middle algo and problem
- 4: Divide & Conquer problems on Codeforces
- 5: Find next greater / Next smaller element using stack
- 6: problems related to parenthesis using stack
- 7: Largest rectangular area in Histogram
- 8: Problem related to Heap (Priority Queue)

7.More Advanced Don't GiveUP (1-4 hr in a problem)-----

- 1: Hashing on strings, know when collision happens (cpalgorithm site)
- 2: Rabin karp algo(it uses hashing)
- 3: Prefix function
- 4: KMP Algo
- 5: Z-Function
- 6: Manacher's Algo (Solve bunch of problem in above topic)

8.Trees – SPOJ - CF-----

- 1: Tree / Graph representation
- 2: DFS/BFS traversal in tree /graph
- 3: Diameter of a tree/Height

- 4: Euler Tour of tree
- 5: Finding LCA using Euler Tour and using Binary Lifting
- 6: Distance b/w two nodes
- 7: Subtree Problems (Solve prob on above tree prob)

9.Graph-----

- 1: Connected Components
- 2: Topological sort
- 3: Cyclic detection in graph
- 4: Bipartite check in graph
- 5: Shortest Connected Component using Kosaraju's algo
- 6: Dijkstra's Algo
- 7: Bellmanford Algo
- 8: Floyd warshall algo (Solve more problems on above topic - Hackerearth/Codeforce)
- 9: Bridge in Graph
- 10: Articulation point in graph
- 11: Minimum spanning tree & kruskal algo
- 12: Prim's Algo
- 13: 0/1 BFS in linear time (cpalgo)
- 14: Finding bridges online (Solve prob)

10.Frequent Topic on CP-----

1. LCA
2. Centroid Decomposition
3. Heavy Light Decomposition
4. Persistence
5. Simplex Algorithm (you should try this, its okay if you don't, because nobody prepares problem for this)
6. Dynamic programming optimizations
7. Link-cut trees

Of course finishing practice on algorithms of CLRS, themselves, may take months. The ones which I feel are frequently used and must be practiced are-

Binary search

1. Dijkstra's Algorithm
2. Minimum Spanning Tree
3. Dfs and bfs
3. Number theoretic algorithms like primality testing and heavy modular mathematics.
4. Disjoint set union
5. KMP string matching
6. Segment trees (and Fenwick trees not in CLRS)
7. Dynamic programming
8. Convex hull computation
9. FFT

Other helpful topics

1. Bitmasking

11.Dynamic Programming-----

1: Start with Recursion & Memoization with strong knowledge. – AND MEMORIZE SOLUTION

2: Knapsack and LCS prob solve

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

4: MUST Solve problem from SPOJ(specially), then Codeforces.

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

6: Read DP with bitmasks and solve on hackerearth

7: DP in trees (Rachit jain video)

8: SOS DP - CF

Practice More(NOT EASY)

12.More-----

- 1: Disjoint Set(Using all optimizations)
- 2: Offline Queries using Disjoint Set
- 3: Kruskal's Algo
- 4: Sparse Table (Not Imp)
- 5: Fenwick Tree (Read Update Trick also) - refer video tuf
- 6: Binary Lifting on fenwick tree (More Solve prob)

13.And More-----

- 1: Matrix Exponentiation
- 2: Sqrt Decomposition -gfg
- 3: Update and query operations
- 4: Mo's Algo (Codeforce blog must)
- 5: Mo's Algo on Trees
- 6: Segment Tree (Most Imp topic - Range queries and point updates)
- 7: Lazy propagation in segment tree

This help you to E- level problems on Codeforces as least.

14.At Last-----

- 1: Sprague-Grundy Theorem -Gaurav Sen video
- 2: Flows and related prob
- 3: Heavy light decomposition - refer Anudeep blog at google
- 4: Convex Hull Algo - blog on CF
- 5: FFT/NTT

Learn all basic algos on Hackerearth.