

PART-2

ROADMAP TO LEARN DATA STRUCTURES & ALGORITHMS STEP-BY-STEP

Curated By-



HIMANSHU KUMAR(LINKEDIN) <http://www.linkedin.com/in/himanshukumarmahuri>

Telegram channel- https://t.me/the_rising_engineers

TOPICS COVERED -

- Basic Data Structures: Arrays, Strings, Stacks, Queues
- Asymptotic analysis (Big-O notation)
- Basic math operations (addition, subtraction, multiplication, division, exponentiation)
- Sqrt(n) primality testing
- Euclid's GCD Algorithm
- Basic Recursion
- Greedy Algorithms
- Basic Dynamic Programming
- Naive string searching
- $O(n \log n)$ Sorting
- Binary Searching

PART-2

Basic Data Structures:

Arrays, Strings, Stacks, Queues

Arrays

Resources:-

- I. [codechef.com - Data Structure Tutorial: Array](https://www.codechef.com/tutorial/array)
- II. [cs.cmu.edu - Arrays](https://cs.cmu.edu/15-122/lectures/arrays/)
- III. [geeksforgeeks.org - Arrays Data Structure](https://www.geeksforgeeks.org/arrays-data-structure/)

Practice Problems:-

- I. [codechef.com - LECANDY](https://www.codechef.com/LECANDY), [editorial](#)
- II. [codechef.com - CNOTE](https://www.codechef.com/CNOTE), [editorial](#) ;
- III. [codechef.com - SALARY](https://www.codechef.com/SALARY), [editorial](#)
- IV. [codechef.com - CHN15A](https://www.codechef.com/CHN15A), [editorial](#)
- V. [codechef.com - RAINBOWA](https://www.codechef.com/RAINBOWA), [editorial](#)
- VI. [codechef.com - FRGTNLNG](https://www.codechef.com/FRGTNLNG), [editorial](#)
- VII. [codechef.com - COPS](https://www.codechef.com/COPS), [editorial](#)

Strings

Resources

- I. [tutorialspoint.com - C++ strings](https://www.tutorialspoint.com/cplusplus/cpp_strings.htm)
- II. [guru99.com - Java strings](https://www.guru99.com/java-strings.html)
- III. [docs.python.org - Python strings](https://docs.python.org/3/library/string.html)
- IV. [tutorialspoint.com - Python strings](https://www.tutorialspoint.com/python/python_strings.htm)
- V. [geeksforgeeks.org - Many string questions](https://www.geeksforgeeks.org/string-questions/)

Practice Problems

PART-2

- I. codechef.com - [CSUB](#), [editorial](#)
- II. codechef.com - [LAPIN](#), [editorial](#)

Stack and Queue

Resources

- I. geeksforgeeks.org – [Stack Data Structure](#)
- II. geeksforgeeks.org – [Introduction and Array Implementation](#)
- III. tutorialspoint.com – [Data Structures Algorithms](#)
- IV. cs.cmu.edu – [Stacks](#)
- V. iterbi-web.usc.edu – [Stacks and Queues](#)
- VI. cs.cmu.edu – [Stacks and Queues](#)

Practice Problems

- I. spoj.com – [JNEXT](#)
- II. spoj.com – [STPAR](#)
- III. spoj.com – [ONP](#)
- IV. codechef.com – [COMPILER](#)
- V. spoj.com – [MMASS](#)
- VI. spoj.com – [HISTOGRAM](#)
- VII. codeforces.com – [D. Maximum XOR Secondary](#)
- VIII. spoj.com – [ANARC09A](#)
- IX. codeforces.com – [C. Minimal string](#)
- X. codeforces.com – [B. Alternating Current](#)
- XI. codeforces.com – [C. Longest Regular Bracket Sequence](#)

Curated By-



HIMANSHU KUMAR(LINKEDIN) <http://www.linkedin.com/in/himanshukumarmahuri>

Telegram channel- https://t.me/the_rising_engineers

PART-2

Asymptotic analysis (Big-O notation)

Basic

1. youtube.com - [Time complexity of a computer program](#)
2. youtube.com - [Big-O notation in 5 minutes - The basics](#)
3. youtube.com - [Definition Of Big O Notation - Intro to Theoretical Computer Science](#)
4. youtube.com - [Algorithms Lecture 1 -- Introduction to asymptotic notations](#)
5. iarcs.org.in - [Measuring the efficiency of algorithms](#)
6. runestone.academy - [Particularly for Big-O notation](#)

Advanced

1. rob-bell.net - [A beginner's guide to Big O notation](#)
2. youtube.com - [Big O Notation, Gayle Laakman McDowell](#)
3. web.mit.edu - [Big O notation](#)
4. youtube.com - [Time and space complexity analysis of recursive programs - using factorial](#)
5. [A very nice tutorial with examples](#)

Practice Problems

1. Check some MCQs on space and time complexity [here](#).
2. You can see some problems with solutions here: [Time complexity of an algorithm](#)

Basic math operations (addition, subtraction, multiplication, division, exponentiation)

- codechef.com - [A tutorial on Fast Modulo Multiplication](#)

PART-2

Euclid's GCD Algorithm

Resources

1. youtube.com - [Mycodeschool video](#)
2. khanacademy.org - [The Euclidean Algorithm](#)
3. geeksforgeeks.org - [Example program to find gcd in c++:](#)

Prime Numbers, divisibility of numbers

Resources:

1. Only $O(\sqrt{n})$ algorithm for finding whether a number is a prime, factorization of a number.
2. [Finding prime factors by taking the square root](#)

Practice Problems:

1. community.topcoder.com - [DivisorInc](#)
2. community.topcoder.com - [Prime Polynom](#)
3. community.topcoder.com - [Prime Anagrams](#)
4. community.topcoder.com - [Refactoring](#)

Basic Recursion

Resources:

1. topcoder.com - [An Introduction to Recursion, Part 1](#)
2. topcoder.com - [An Introduction to Recursion: Part 2](#)
3. geeksforgeeks.org - [Recursion](#) ;(along with questions)
4. web.mit.edu - [Recursion](#)
5. csee.umbc.edu - [Recursion](#) ;(Examples with exercises)
6. loveforprogramming.quora.com - [Backtracking, Memoization & Dynamic Programming](#)
7. byte-by-byte - [Recursion for Coding Interviews](#)

PART-2

Practice Problems:

1. codechef.com - [NOKIA](#), [editorial](#)
2. codechef.com - [TRISQ](#), [editorial](#)
3. codechef.com - [LFSTACK](#), [editorial](#)
4. codechef.com - [FICE](#), [editorial](#)

Greedy Algorithms

Resources

1. iarcs.org.in - [Greedy Algorithms](#)
2. iarcs.org.in - [Greedy Algorithms](#)
3. topcoder.com - [Greedy Algorithms](#)
4. [Greedy Algorithms](#)

Practice Problems

- codechef.com - [TACHSTCK](#), [editorial](#)
- codechef.com - [CIELRCPT](#), [editorial](#)
- codechef.com - [MAXDIFF](#), [editorial](#)
- codechef.com - [CHEFST](#), [editorial](#)
- codechef.com - [CAKEDOOM](#), [editorial](#)
- codechef.com - [CLETAB](#), [editorial](#)
- codechef.com - [TADELIVE](#), [editorial](#)
- codechef.com - [MANYCHEF](#), [editorial](#)
- codechef.com - [MMPROD](#), [editorial](#)
- codechef.com - [CHEFTMA](#), [editorial](#)
- codechef.com - [STICKS](#), [editorial](#)
- spoj.com - [BAISED](#)
- spoj.com - [BALIFE](#)
- spoj.com - [GCJ101BB](#)

PART-2

- codechef.com - [FGFS](#)
- codechef.com - [KNPSK](#)
- codechef.com - [LEMUSIC](#)
- spoj.com - [ARRANGE](#)
- spoj.com - [FASHION](#)

Dynamic programming (Basic DP)

Resources

- medium.freecodecamp.org - [Demystifying Dynamic Programming](#)
- iarcs.org.in - [Dynamic Programming - Tiling](#)
- topcoder.com - [Dynamic Programming – From Novice to Advanced](#)
- illinois.edu - [Dynamic Programming](#) ;(Exercises are recommended)
- codechef.com - [Dynamic Programming](#)
- geeksforgeeks.org - [Dynamic Programming](#) ;(Contains a lot of practice sessions)
- MIT OCW (Contains some Advanced topics as well)
 - [Dynamic Programming I](#)
 - [Dynamic Programming II](#)
 - [Dynamic Programming III](#)
 - [Dynamic Programming IV](#)

Curated By-

HIMANSHU KUMAR(LINKEDIN) <http://www.linkedin.com/in/himanshukumarmahuri>

Telegram channel- https://t.me/the_rising_engineers

PART-2

Practice Problems

- codechef.com - [ALTARAY](#), [editorial](#)
- codechef.com - [DELISH](#), [editorial](#)
- codechef.com - [DBOY](#), [editorial](#)
- codechef.com - [XORSUB](#), [editorial](#)
- codechef.com - [GRID](#), [editorial](#)
- codechef.com - [TADELIVE](#), [editorial](#)
- codechef.com - [FROGV](#), [editorial](#)
- codechef.com - [MATRIX2](#), [editorial](#)
- codechef.com - [AMSGAME2](#), [editorial](#)
- spoj.com - [MDOLLS](#)
- spoj.com - [MSTICK](#)
- spoj.com - [MCARDS](#)
- spoj.com - [MIXTURES](#)
- spoj.com - [SAMER08D](#)
- spoj.com - [AIBOHP](#)

Naive string searching

Resources

- i. [geeksforgeeks.org](https://www.geeksforgeeks.org/naive-pattern-searching/) - [Naive Pattern Searching](#)

PART-2

Sorting

A. [khanacademy.org](https://www.khanacademy.org)

B. visualgo.net

C. [iarcs.org.in](https://www.iarcs.org.in)

D. Merge Sort

➤ youtube.com - [Merge sort algorithm](#)

➤ Practice Problems

codechef.com - [MRGSRT](#)

E. Quick Sort

➤ youtube.com - [Quicksort algorithm](#)

➤ Practice Problems

codechef.com - [TSORT](#)

F. Counting Sort

➤ geeksforgeeks.org - [Counting Sort](#)

➤ Practice Problems

➤ codechef.com - [TACHSTCK](#), [editorial](#)

➤ codechef.com - [STICKS](#), [editorial](#)

Binary Search

Resources

1. [topcoder.com](https://www.topcoder.com) (Try solving problems of Simple and Moderate level as mentioned in the end of the link)
2. [codechef.com](https://www.codechef.com)
3. [usfca.edu](https://www.usfca.edu)
4. [khanacademy.org](https://www.khanacademy.org)

Detailed Theoretical analysis

- [cmu.edu](https://www.cmu.edu) (A theoretical analysis)

PART-2

Problems

- geeksforgeeks.org - [Binary Search](#) (Contains some solved problems)
- codechef.com - [STRSUB](#), [editorial](#)
- codechef.com - [ASHIGIFT](#), [editorial](#)
- codechef.com - [STACKS](#), [editorial](#)
- codechef.com - [DIVSET](#), [editorial](#)
- codechef.com - [LOWSUM](#), [editorial](#)
- codechef.com - [SNTEMPLE](#), [editorial](#)
- codechef.com - [SNAKEEAT](#), [editorial](#)
- codechef.com - [SCHEDULE](#), [editorial](#)
- codechef.com - [RIGHTTRI](#), [editorial](#)
- codechef.com - [FORESTGA](#), [editorial](#)
- codechef.com - [CHEFHCK2](#), [editorial](#)
- spoj.com - [ABCDEF](#)
- spoj.com - [NOTATRI](#)
- spoj.com - [SCALE](#)
- spoj.com - [SUMFOUR](#)
- spoj.com - [SUBSUMS](#)
- spoj.com - [ANARC05B](#)
- spoj.com - [RENT](#)
- spoj.com - [PIE](#)
- spoj.com - [MKUHAR](#)
- spoj.com - [SVADA](#)
- spoj.com - [SUBS](#)

Curated By-



HIMANSHU KUMAR(LINKEDIN)

<http://www.linkedin.com/in/himanshukumarmahuri>

Telegram channel- https://t.me/the_rising_engineers

ADVANCED DSA TOPICS FOR PART-2

TOPICS-

- Heaps (priority queue)
- Disjoint Set Union
- Segment Trees
- Binary Index Tree (Fenwick tree)
- Trees (traversals, tree dynamic programming)
- Finding Lowest Common Ancestors ($O(\log N)$ solution where N is number of nodes).
- Graph Algorithms:
 - Finding connected components and transitive closures.
 - Shortest-path algorithms (Dijkstra, Bellman-Ford, Floyd-Warshall)
 - Minimum spanning tree (Prim and Kruskal algorithms)
 - Biconnectivity in undirected graphs (bridges, articulation points)
 - Strongly connected components in directed graphs

PART-2

- Topological Sorting
- Euler path, tour/cycle.
- Modular arithmetic including division, inverse
- Amortized Analysis
- Divide and Conquer
- Advanced Dynamic Programming problems (excluding the dp optimizations which are added in expert level)
- Sieve of Eratosthenes

MUST JOIN THE TELEGRAM CHANNEL FOR NOT MISSING ANY FUTURES UPDATES.

Curated By-



HIMANSHU KUMAR(LINKEDIN)

<http://www.linkedin.com/in/himanshukumarmahuri>

Telegram channel- https://t.me/the_rising_engineers