



ALGORITHMS & DATA STRUCTURES PREPARATION STRATEGIES

Create profiles in the following platforms:

- i) HackerRank (<https://hackerrank.com>)
- ii) CodeChef (<https://www.codechef.com>)
- iii) Interview Bit (<https://www.interviewbit.com/>)
- iv) Sphere Online Judge (SPOJ) (<http://www.spoj.com/>)
- v) Top Coder (<https://www.topcoder.com/>)
- vi) CodeForces (<http://codeforces.com/>)
- vii) Leet Code (<https://leetcode.com/>)

Try to practice as much as you can. Make sure cumulative of all the scores in above platforms will be around 10,000 to 15,000 in the next 3 to 4 months. It's more than enough to get into any top product based corporates/startups.

Refer your entire initial class notes

Solving more problems from the above platforms like interviewbit, codechef, codeforces, spoj etc

Preparing other topics like DB, OS, CN, Puzzles, Language Specific Questions, Design patterns, Web technologies

****(Optional for non-technical profiles)**

TOPICS:

PHASE 1

Interpreters Vs Compilers (Refer our initial classes notes)

Time and Space Limits for Programs

Process Memory Hierarchy (Refer our initial classes notes)

Bits/Bytes

32bit vs 64bit architecture

One's Complement and Two's Complement for positive and negative numbers (Refer our initial classes notes)

Javascript and ES6 concepts (Array Helper Methods)

Bit Manipulation Programs (Refer our initial classes notes) + <https://www.geeksforgeeks.org/bitwise-algorithms/>

Concept of Time Complexities and Space Complexities

1. Asymptotic Analysis

2. Worst, Average And Best Cases

3. Asymptotic Notations

4. Analysis Of Loops

5. Analysis Of Recursion: Solving Recurrences (Masters Theorem) (Refer our initial classes notes/Slack group)

6. Amortized Analysis

7. Space Complexity

(Refer our Slack group for more resources)

<https://www.interviewbit.com/courses/programming/topics/time-complexity/>

Recursion Problems and Debugging Recursions

Fibonacci series problems

Solving Tower of Hanoi

Interleavings problems

(Refer our initial classes notes)

PHASE 2

Sorting Algorithms

1. Bubble Sort
2. Insertion Sort
3. Selection Sort
4. Merge Sort
5. Quick Sort
6. Radix Sort
7. Heap Sort
8. Counting Sort

Refer all the above sorting techniques in Geeks For Geeks and other online resources. Learn sortings along with their Time and Space Complexities.

Ex: Median of two sorted array

Solution:

- a. <http://www.geeksforgeeks.org/median-of-two-sorted-arrays/>
- b. <http://www.geeksforgeeks.org/median-of-two-sorted-arrays-of-different-sizes/>

Searching Algorithms

1. Linear Search
2. Binary Search
3. Ternary Search

<https://www.geeksforgeeks.org/searching-algorithms/>

<https://www.topcoder.com/community/data-science/data-science-tutorials/binary-search/>

<https://www.codechef.com/wiki/tutorial-binary-search>

<http://www.geeksforgeeks.org/search-in-row-wise-and-column-wise-sorted-matrix/>

<http://www.spoj.com/problems/TEST/>

<http://www.spoj.com/problems/AGGRCOW/>

<http://www.spoj.com/problems/SUMFOUR>

(Refer all the above searching techniques in Geeks For Geeks and other online resources.)

Hashing

1. Hash Tables,
2. Hash Functions,
3. Complexity
4. Collision Resolution*****

<https://www.geeksforgeeks.org/hashing-data-structure/>

(Refer all the above hashing techniques in Geeks For Geeks and other online resources.)

PHASE 3

Linked Lists

Singly Linked List

Doubly Linked List(Refer our class Notes/Slack)

Circular Linked List

Advantages and Disadvantages of LL

Applications of LL

(Refer all the above LL concepts in Geeks For Geeks and other online resources.)

Stacks

Implementing Stack using Array (Refer our class Notes/Slack)

Implementing Stack using Linked List

(Refer all the above Stack concepts in Geeks For Geeks and other online resources.)

Queues

Implementing Queues using Array

Implementing Queues using Linked List

(Refer all the above Stack concepts in Geeks For Geeks and other online resources.)

PHASE 4(Advanced)

Trees

General Tree

Binary Trees

Binary Search Trees

Threaded Binary Tree

(Refer all the above Trees concepts in Geeks For Geeks and other online resources.)

Priority Queues And Heaps

Implementations of Priority Queue

Applications of Priority Queue

- Priority scheduling of processes in OS
- Dijkstra's Algorithm for shortest path
- Prim's Algorithm for minimum spanning tree
- Heapsort

Implementation of Heaps

(Refer all the above PQ&Heaps concepts in Geeks For Geeks and other online resources.)

Trie*

(Trie is an efficient information retrieval data structure)

Implementation of Trie

(Refer Trie concepts in Geeks For Geeks and other online resources.)

Greedy Algorithms

1. Prim's MST Algorithm
2. Dijkstra's Algorithm
3. Kruskal's MST Algorithm
4. Huffman Coding

References:

<https://www.topcoder.com/community/data-science/data-science-tutorials/greedy-is-good/>

<http://staff.ustc.edu.cn/~csli/graduate/algorithms/book6/chap17.htm>

(Refer all the above Greedy Algorithms concepts in Geeks For Geeks and other online resources.)

Dynamic Programming

(Refer Dynamic Programming PDF that is uploaded in our Slack group) +

<http://people.csail.mit.edu/bdean/6.046/dp/>

<https://www.topcoder.com/community/data-science/data-science-tutorials/dynamic-programming-from-novice-to-advanced/>

<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-videos/lecture-19-dynamic-programming-i-fibonacci-shortest-paths/>

<http://www.geeksforgeeks.org/fundamentals-of-algorithms/#DynamicProgramming>

Graph Theory

Graph Representation

Graph Traversals

Topological Sort

Shortest Path Algorithms

Minimum Spanning Tree

Disjoint Set Data Structure

Shortest Path Problems:

<http://www.spoj.com/problems/MICEMAZE/>

<http://www.spoj.com/problems/SHPATH/>

<http://www.spoj.com/problems/HIGHWAYS/>

<https://www.codechef.com/problems/HOMDEL>

<http://www.spoj.com/problems/BUGLIFE/> - Bipartite

(Refer all the above Graph Theory concepts in Geeks For Geeks and other online resources.)

HAPPY LEARNING