DSA

1. Basics
    - Basic of any Language
        - □ Function
        - □ Conditionals
        - □ Loops
        - □ Patterns
        - □ Pointers
        - □ Bitwise Operators
        - □ OOP
        - □ Basic Problem Solving In Language
    - Time and Space Complexity
    - Mathematics
        - □ GCD, LCM
        - □ Check for Prime
        - □ Prime Factors
        - □ Sieve of Eratosthenes
        - □ Computing Power

---

2. Arrays & Strings (most Important)
    - Basic Understanding
        - □ Fixed and Dynamic
        - □ Operations
    - Algorithms
        - □ Kadane Algo
        - □ Dutch National Flag Algo
        - □ Sliding Window
        - □ Two Pointers
    - String Algorithms
        - □ Rabin Karp Algorithm
        - □ KMP Algorithm

3. Multidimensional Array

- □ Traversal based Prob
- □ Spiral Order of Array etc
- □ Rotate, Transpose

---

4. Matrix

- □ Introduction to Matrix in C++ and Java
- □ Multidimensional Matrix
- □ Pass Matrix as Argument
- □ Printing matrix in a snake pattern
- □ Transposing a matrix
- □ Rotating a Matrix
- □ Check if the element is present in a row and column-wise sorted matrix.
- □ Boundary Traversal
- □ Spiral Traversal
- □ Matrix Multiplication
- □ Search in row-wise and column-wise
- □ Sorted Matrix

---

5. Recursion & Backtracking
(2nd most Important)

- □ Basic recursion
- □ Factorial ,Fibbonacci etc

- □ ***Divide & Conquer

---

6. Sorting Algorithms

- □ Insertion Sort
- □ Selection Sort
- □ Bubble Sort
- □ Merge Sort
- □ Quick Sort
- □ Heap Sort
- □ Cycle Sort
- □ Counting Sort
- □ Radix Sort
- □ Bucket Sort

- ○ Partitions
  - □ Naive
  - □ Lomuto
  - □ Hoare

---

7. Searching Algorithms

- ○ Binary Search
  - □ Iterative & Recursive
  - □ Applications
  - □ BS on array
  - □ BS on matrix -row column etc
- ○ *Other Search Algorithms*

** MAIN DSA**

---

8. Linked List

- □ Reversal Problems
- □ Sorting Problems & previously learnt Sorting Techniques
- □ Slow & Fast pointers
- □ Modifying Linked List

- ○ Doubly Linked List
- ○ Circular Linked List
- ○ Loop Problems
  - □ Detecting Loops
  - □ Detecting loops using Floyd Cycle
  - □ Detecting and Removing Loops in Linked List

---

9. Stacks & Queues

- □ Stack Implementation using array, linked list etc.
- □ Infix , Prefix, Postfix Implementation
- □ Priority Queue
- □ Deque (only understanding)

10. Binary Trees

- ❑ Construction of tree
- ○ Standard Problems
  - ❑ Remove
  - ❑ Insert
- ○ Tree Traversal
  - ❑ Inorder, Preorder, Postorder Traversal
  - ❑ Level Order, Spiral form traversal etc
- ○ View
  - ❑ Tree Views
  - ❑ Top View
  - ❑ Bottom View

---

11. BST

- ❑ Construction
- ❑ Insertion, Deletion
- ❑ Floor in BST
- ❑ Self Balancing BST
- ❑ Conversion based problem
- ❑ Modification in BST
- ❑ Standard Problems
- ○ All similar topics to study same as BINARY TREE
- ○ *Other trees AVL. Red Black etc*

---

12. Priority Queues

- ❑ Implementation based problem
- ❑ Convert heap to BST, Binary Tree, Linked List
- ❑ K based problems
- ❑ Priority Queues

13. Graphs
(FAANG)

- □ BFS DFS
- □ Min Spanning Tree
- □ Djikstra Path Algo
- □ Belman ford Algo
- □ Kruskal Algo
- □ Topological Sort
- □ Graphs in Matrix

14. Dynamic
Programming

- □ Is an advanced version of Recursion
- □ Memoization, Tabulation
- □ DP with arrays
- □ DP with strings
- □ DP with Maths
- □ DP with Trees
- □ Breaking & Partition Problems
- □ Counting Based Problems
- □ Variety problems

15. Hard Questions
Recursion &
Backtracking

- □ Print permutation
- □ Print subset
- □ Rat in maze problem
- □ N queen problem
- □ Sudoku problem

16. Other Topics

- □ Dequeue (Most Imp)
- □ Hashmaps (implementation imp)
- □ Tries
- □ Bit Manipulation
- □ **Greedy Algo**
- □ Circular Queues

- ❑ Doubly / Circular LL
- ❑ String Algo like KMP and Z Algo
- ❑ Segment tree
- ❑ Disjoint set

- ➢ Time:
  - Weekdays -- 3-4 hrs
  - Weekends -- 7-8 hrs
  - Total 6 months possible
- ➢ 20 Questions on each topic
  - Refer GeeksForGeeks Syllabus file for question types
  - Debugging
    - ❑ Backward analysis
    - ❑ Forward analysis
    - ❑ Blockwise analysis
    - ❑ aditya verma video yt
- ➢ Contests (not Competitive Coding)
  - Start after learning array and strings
  - Hacker Earth
  - Leetcode
  - Read unsolved questions try and see soln
- ➢ Do 100 Random Questions
  - Where you dont know data Structure application is present here.
  - IMP if you're not doing Competitive Coding
  - Makes master in DSA

- Consistency after DSA
  - Remember, Revise
  - Leetcode everyday question
  - GFG everyday question
  - DSA Question sheet

- Competitive Coding
  - Only 100-150 quest needed for placement
  - Dont to if time isnt there
  - Only for regular revision
  - Number Theory