Code Library

Subscribe to the channel = https://bit.ly/3fBvYkf

DSA CheatSheet

1. **Learn a Language**--
   C++/Java/Python

   Resources--
   C++ :
   R1 = https://bit.ly/3uzxmbr (will be completed soon)
   R2 = http://bit.ly/3nOdZZD
   R3 = http://bit.ly/38FifE6
   Java :
   R1 = http://bit.ly/3heJQA8
   R2 = http://bit.ly/3mQ7IuX

2. **Data Structures**--
   1. Arrays
   2. String
   3. Time & Space Complexity
   4. Searching (Linear/Binary)
   5. Sorting (Selection/Bubble/Insertion/Merge/Quick/Heap Sort)
   6. Stack
   7. Queue
   8. Linked List (Single/Doubly)
   9. Hashing
   10. Recursion
   11. Backtracking
   12. STL for C++ or Java collections for Java
   13. Tree & Binary Search Tree
   14. Heap/ priority queue
   15. Graph
   16. Dynamic programming

   Resources--
   R1 = http://bit.ly/3hhe4m1

3. **A) C++ STL**--
   Topics--
   1) Vector
   2) Stack
   3) Set
   4) Map
   5) unordered_set
   6) unordered_map
   7) pair
   8) queue
   9) deque
   10) list
   11) Binary Search/lower_bound/upper_bound
   11) Custom Comparator

   Resources--
   R1 = http://bit.ly/3aICELu
   R2 = http://bit.ly/3mVoiKc
   R3 = https://bit.ly/2JpGmOQ

   B) Java Collections--
   R1 = http://bit.ly/3hi1Utd

4. **Algorithms--**

   1) **Number Theory**--
      a) Fibonacci Series/Number
      b) Prime
      c) Sieve of Eratosthenes
      d) Segmented Seive

e) GCD & Euclid's Algorithm
f) Fast Modulo Exponentiation
g) multiplicative modulo inverse
h) fermat's little theorem

2) **Sorting Algorithms**--
a) Selection Sort
b) Bubble Sort
c) Insertion Sort
d) Quick Sort
e) Merge Sort
f) Heap Sort

3) **Searching--**
a) Linear Search
b) Binary Search

4) **Recursion & Backtracking**--
a) Basic Question
b) Fibonacci Recursion
c) Tower of Hanoi
d) Generate Brackets Recursion
e) Knapsack Recursion
f) Phone Keypad Problem
g) Rat in a maze
h) N-Queen Problem
i) Sudoku Problem

5) **Greedy**

6) **Graph Algorithms**--
a) BFS
b) DFS
c) Directed Graph
d) Undirected Graph
e) Disjoint Set Union
f) Minimum Spanning Tree (kruskal's Algo, Prim's Algo)
g) Shortest Path (Dijkstra's Algo, Bellman Ford, Floyd-Warshall)
h) Cycle Detection
i) Topological Sort / DAG
j) Kosaraju's Algo
k) Connected components / Strongly Connected Comp
l) Eular Tour
m) Articulation Point and Bridge
n) LCA

7) **DP**--
R1 = http://bit.ly/3rs78XV

Algorithm Resources--
R1 = http://bit.ly/3aGKGUV
R2 = http://bit.ly/3hgkGkF

5. Problem Solving Skills--

1. LeetCode = https://leetcode.com/
2. GFG Practice Site = http://bit.ly/2KEp2WJ
3. A2OJ = http://bit.ly/38yRgua
4. Hackerrank = http://bit.ly/3rvG0XQ