

Ram Ram Coders!

SUGGESTIONS FOR OPTIMAL USE

- ➔ Follow the questions in a serial wise manner for optimal results.
- ➔ You can follow the playlists as you wish, but our suggestion for optimal results is the manner as given below.
- ➔ For doubts(academic/non academic), our link to social media handles are given as follows:

[Telegram](#) , [Instagram](#) , [YouTube](#)

Hope you achieve great heights

Achhe Placement ka sapna bhul nhi jaana re

Maths

1. [Find factors check for prime.](#)
2. [Prime Sum](#)
3. [Missing Number in an array](#)
4. [Trailing zeroes in a factorial.](#)
5. [Next similar number.](#)
6. [Missing two numbers.](#)
7. [nCr.](#)

8. [Unique Paths.](#)
9. [LCM and GCD.](#)
10. [Smallest positive number that cannot be represented as a sum.](#)

Arrays

1. [Left And Right Rotate](#)
2. [2-Sum](#)
3. [Remove Duplicates From Sorted Array.](#)
4. [Wave Array.](#)
5. [Possible Triangle Triplet.](#)
6. [Find Pivot Index](#)
7. [3 Sum Closest](#)
8. [Merge Sorted Array](#)
9. [Sort Colors](#)
10. [Binary search Introduction](#)
11. [Forest Search](#)
12. [Painters Partitioning](#)
13. [Allocate Minimum Number of Pages](#)
14. [Find Missing in 2D Array](#)
15. [Sort An Array According To Other](#)
16. [Largest Sum Contiguous Subarray](#)
17. [Stock Buy and Sell](#)
18. [Sliding Window Technique](#)
19. [Maximum Number of Vowels in a Substring](#)
20. [Sliding Window Maximum](#)

21. [Jump Game](#)
22. [Search in Rotated Sorted Array](#)
23. [Lower Bound And Upper Bound - STL](#)
24. [Rotate Image](#)
25. [Matrix Multiplication](#)
26. [Sort The Matrix Diagonally](#)
27. [Search in a row column Sorted matrix](#)
28. [Is Sudoku Valid](#)
29. [Row With Max 1's](#)
30. [Set Matrix Zeroes](#)
31. [Trapping Rain Water](#)
32. [Spiral Matrix](#)

Hashing

1. [Intro to hash-maps.](#)
2. [Longest sub-array with zero sum.](#)
3. [Longest sub-array with zero sum-2.](#)
4. [Check if a string is isogram](#)
5. [Anagrams.](#)
6. [Longest substring without repeating characters.](#)
7. [Array subset of another array.](#)
8. [Longest consecutive subsequence.](#)
9. [Sum equals to sum.](#)
10. [Find all pairs with a given sum.](#)
11. [Zero sum sub-arrays.](#)
12. [Longest sub-array with at most k distinct elements.](#)

13. [Top k frequent elements.](#)
14. [Sub-arrays with k distinct integers.](#)
15. [Find largest d in the array such that \$a+b+c=d\$.](#)

Strings

1. [Reverse Words in a given Strings.](#)
2. [Add Strings](#)
3. [Integer To Roman](#)
4. [Recursively Remove All Adjacent Duplicates](#)
5. [Implement Atoi](#)
6. [Remove All Adjacent Duplicates in String](#)
7. [Longest Palindrome Substring](#)
8. [Most Important String str.find\(\) explained.](#)

Stack

1. [Parenthesis Checker](#)
2. [Evaluation Of Postfix Expression](#)
3. [Infix To Postfix](#)
4. [Next Greater Element](#)
5. [Stock Span Problem](#)
6. [The Celebrity Problem](#)
7. [Min Stack](#)
8. [Largest Rectangle in histogram](#)

9. [Remove K Digits](#)
10. [Maximum Rectangle](#)

Queue + Priority_queue + Heap

1. [Reverse First k Elements of Queue.](#)
2. [Stack Using Queue](#)
3. [Queue Using Two Stacks.](#)
4. [First Non-Repeating Character in a Stream.](#)
5. [Rotten Oranges.](#)
6. [Steps By knight](#)
7. [Minimum Cost Of Ropes](#)
8. [Kth Largest Element in a stream](#)
9. [Find Medium in A Stream](#)

Greedy

1. [Maximum product of 3 numbers.](#)
2. [Maximum number of toys.](#)
3. [Minimum operations.](#)
4. [Majority element.](#)
5. [Fractional knapsack.](#)
6. [Minimum platforms.](#)
7. [Best time to buy and sell stocks.](#)
8. [Jump game 2.](#)
9. [Policeman catch thieves.](#)

10. [Merge intervals.](#)
11. [Minimize the heights 2.](#)
12. [Disjoint intervals.](#)
13. [Gas station.](#)
14. [Maximum sum of increasing order elements from n arrays.](#)
15. [Job sequencing problem.](#)
16. [N-meetings in one room.](#)
17. [Coin piles.](#)
18. [Water the plants.](#)

Recursion.

1. [Recursion beginner's video.](#)
2. [Geeco-nacci number.](#)
3. [Print 1 to N without using loop.](#)
4. [Pascal's triangle](#)
5. [Delete middle element of stack.](#)
6. [Sort a stack.](#)
7. [Power of numbers.](#)
8. [Find kth bit in nth binary string](#)
9. [Josephus problem.](#)
10. [Print N-bit binary numbers having more 1s than 0s.](#)
11. [Subset sums.](#)
12. [Count good numbers.](#)
13. [Generate parentheses.](#)
14. [Permutation with spaces.](#)
15. [Predict the winner.](#)

16. [Tower of Hanoi.](#)

Linked List

1. [Reverse Linked List](#)
2. [Delete Node in Linked List](#)
3. [Middle Of the Linked List](#)
4. [Linked List Cycle](#)
5. [Linked List Cycle 2](#)
6. [Swap Nodes in pairs](#)
7. [Add Two Numbers](#)
8. [Palindrome Linked List](#)
9. [Remove Nth Node from End of List.](#)
10. [Intersection of two Linked List](#)
11. [Partition List.](#)
12. [Rearrange A Linked List](#)
13. [Merge two Sorted Lists.](#)
14. [Rotate List.](#)
15. [Reverse Nodes in k-Group.](#)
16. [Sort List.](#)
17. [Copy List With Random Pointer](#)
18. [Sorted Insert For Circular Linked List.](#)
19. [Flattening a Linked List.](#)

Binary Tree

1. [Binary Tree – Introduction.](#)
2. [Inorder Traversal.](#)
3. [Preorder & Postorder Traversal.](#)
4. [Maximum Depth Of Binary Tree.](#)
5. [Binary Tree level order Traversal.](#)
6. [Level Order Traversal in Spiral Form.](#)
7. [Vertical Traversal Of Binary Tree](#)
8. [Left And Right View of Binary tree.](#)
9. [Boundary Traversal of Binary Tree.](#)
10. [Symmetric Tree.](#)
11. [Populating Next Right Pointers in Each Node.](#)
12. [Sum Tree.](#)
13. [Balanced Binary Tree.](#)
14. [Lowest Common Ancestor.](#)
15. [Inorder Traversal.](#)
16. [Postorder Traversal \(Iterative\).](#)
17. [Preorder Traversal \(Iterative\).](#)
18. [Binary Tree Paths](#)
19. [Burning Tree.](#)
20. [All Nodes Distance k in Binary Tree](#)
21. [Diameter Of Binary Tree.](#)
22. [Path Sum.](#)
23. [Path Sum 2.](#)
24. [Contructing Binary Tree from Post/Pre/In-Order](#)

25. [Construct Binary Tree from Preorder & inorder.](#)
26. [Binary Tree From Preorder & Postorder.](#)

Binary Search Tree

1. [Search in Binary Search Tree.](#)
2. [Validate Binary Search Tree.](#)
3. [Insert Into Binary Search Tree.](#)
4. [Delete Node In BST.](#)
5. [Convert Sorted List to Binary Search Tree.](#)
6. [Balance A Binary Search Tree.](#)
7. [Construct Binary Search Tree From Preorder Traversal.](#)
8. [Construct Binary Search Tree From Postorder Traversal.](#)
9. [Delete Nodes Greater Than K.](#)

Backtracking.

1. [Backtracking for beginners.](#)
2. [Flood fill algorithm.](#)
3. [Rat in a maze.](#)
4. [Permutations.](#)
5. [Possible words from phone digits.](#)
6. [Path with maximum gold.](#)
7. [Word search.](#)

8. [N-queens.](#)
9. [Sudoku.](#)
10. [Palindrome partitioning.](#)

Graph

1. [Graph Representation \(Introduction \).](#)
2. [Graph Traversal \(Adjacency List And Adjacency Matrix\).](#)
3. [BFS \(Adjacency List\).](#)
4. [DFS \(Adjacency List\).](#)
5. [Number Of Island.](#)
6. [Find Whether Path Exist.](#)
7. [Snake and Ladder.](#)
8. [Covid Spread.](#)
9. [Detect Cycle in Undirected Graph \(BFS\).](#)
10. [Detect Cycle in Undirected Graph \(DFS\)](#)
11. [Bipartite Graph.](#)
12. [Bipartite Graph \(BFS\)](#)
13. [Number Of Connected Components in Undirected Graph](#)
14. [Topological Sort.](#)
15. [Kahn's Algorithm](#)
16. [Detect Cycle in undirected Graph \(DFS\).](#)
17. [Detect Cycle in Undirected Graph \(BFS\).](#)
18. [Dijkstra's Shortest path Algorithm.](#)
19. [Dijkstra's Shortest Path Algorithm \(Code\)](#)

20. [Introduction Of Minimum Spanning Tree](#)
21. [Introduction To Prim's Algorithm.](#)
22. [Prim's Algo Code](#)
23. [Union and Find](#)
24. [Kruskal's Algo](#)
25. [Kruskal's algo-code](#)
26. [Bellman Ford algo](#)
27. [Bellman Ford Algo\(code\)](#)
28. [Strongly Connected Components\(Kosaraju's Algo\)](#)

Dynamic Programming.

1. [DP's beginner's video.](#)
2. [Count ways to reach nth stair.](#)
3. [Reach a given score.](#)
4. [Coin change.](#)
5. [Coin change 2.](#)
6. [Counting bits.](#)
7. [Number of unique paths.](#)
8. [Player with max score.](#)
9. [Stickler thief.](#)
10. [0-1 knapsack problem.](#)
11. [Knapsack with duplicate items.](#)
12. [Maximum sum problem.](#)
13. [Longest common subsequence.](#)
14. [Maximum number of insertions and deletions.](#)

15. [Longest Palindromic subsequence.](#)
16. [Minimum cost to make two strings identical.](#)
17. [Buying vegetables.](#)
18. [Longest increasing subsequence.](#)
19. [Minimum number of deletions to a sorted sequence.](#)
20. [Distinct subsequences.](#)
21. [Repeating subsequences.](#)
22. [Regular expression matching.](#)
23. [Regular expression matching 2.](#)
24. [Shortest common subsequence.](#)
25. [Max length chain.](#)
26. [Interleaving string.](#)
27. [Scramble strings.](#)
28. [Egg dropping puzzle.](#)
29. [Intro to tabular DP.](#)
30. [Nth Fibonacci number.](#)
31. [Count ways to reach nth stairs.](#)
32. [A DP guide.](#)
33. [Reach a given score.](#)
34. [Coin change.](#)
35. [Coin change 2.](#)
36. [0-1 Knapsack problem.](#)
37. [Knapsack with duplicate items.](#)
38. [Subset sum problem](#)
39. [Minimum sum partition.](#)
40. [Longest Common subsequence.](#)
41. [Longest increasing subsequence.](#)

42. [Longest increasing subsequence 2.](#)
43. [Longest common substring.](#)
44. [Regular expression match.](#)
45. [Number of unique paths.](#)
46. [Box stacking.](#)
47. [Predict the winner.](#)
48. [Regular expression match.](#)
49. [Matrix Chain Multiplication\(Recursive\)](#)
50. [Matrix Chain Multiplication\(Tabular\)](#)
51. [Count Palindromic Subsequences](#)

BITS MANIPULATION

1. [Bit manipulation all you need to know.](#)
2. [Number of 1 Bits.](#)
3. [Sum of numbers without using mathematical operators.](#)
4. [Reverse Bits.](#)
5. [Single Number.](#)
6. [Find Missing and Repeating.](#)
7. [Single Number 2.](#)
8. [Sub array with given XOR.](#)
9. [XORing the Sub arrays.](#)
10. [Bit Difference.](#)

For queries, contact us Telegram , Instagram ,YouTube

© **Bro Coders**