



Aptitude - Indiabix

DSA Preparation - Striver A-Z DSA sheet.

DSA questions - Leetcode, geeksforgeeks
(alternative for LC premium)

Short revision notes - InterviewBit

Puzzles - InterviewBit, geeksforgeeks

HR, Managerial Round questions - Indeed/
InterviewBit

Add a comment...



Browse



Create



Send



More



1. Sliding Window

Description: Used for problems that involve finding a subarray or substring with specific properties within a given array or string.

Use Cases: Maximum sum subarray of size k , longest substring without repeating characters.



patterns 12 October
From Create Mode

3. Fast and Slow Pointers (Tortoise and Hare)

Description: Typically used in cyclic linked lists to detect cycles.

Use Cases: Detecting cycles in a linked list, finding the middle of a linked list.



4. Merge Intervals

Description: Used to merge overlapping intervals.

Use Cases: Merging meeting times, merging overlapping intervals.



5. Cyclic Sort

Description: Used to sort numbers when the range is known (e.g., 1 to N).

Use Cases: Finding missing numbers, finding duplicate numbers.



6. In-place Reversal of a Linked List

Description: Used to reverse a linked list or part of it.

Use Cases: Reversing a linked list, reversing part of a linked list.



7. Breadth-First Search (BFS)

Description: Used for traversing or searching tree/graph structures level by level.

Use Cases: Shortest path in an unweighted graph, level-order traversal of a tree.



8. Depth-First Search (DFS)

Description: Used for traversing or searching tree/graph structures by going as deep as possible before backtracking.

Use Cases: Detecting cycles in a graph, path finding in a maze.

9. Dynamic Programming (DP)

Description: Used for solving problems by breaking them down into simpler subproblems and storing the results to avoid redundant computations.

Use Cases: Fibonacci sequence, knapsack problem, longest common subsequence.

10. Greedy Algorithm

Description: Makes the locally optimal choice at each stage with the hope of finding the global optimum.

Use Cases: Activity selection, coin change problem.

11. Backtracking

Description: Used for solving problems by trying out all possible solutions and backtracking when a solution fails.

Use Cases: N-Queens problem, Sudoku solver.

13. Binary Search

Description: Efficiently finds an element in a sorted array by repeatedly dividing the search interval in half.

Use Cases: Finding an element in a sorted array, searching for the first/last occurrence of a target, finding the peak element in a mountain array.

12. Union-Find

Description: Used to detect cycles in a graph and for disjoint-set operations.

Use Cases: Kruskal's algorithm for Minimum Spanning Tree, network connectivity.

14. Topological Sort

Description: Linear ordering of vertices in a directed acyclic graph (DAG) such that for every directed edge $u \rightarrow v$, vertex u comes before v .

Use Cases: Course schedule, task scheduling.



15. Kadane's Algorithm

Description: Used to find the maximum sum subarray in an array.

Use Cases: Maximum sum subarray, finding the largest sum contiguous subarray.



patterns 12 October
From Create Mode

16. Prefix Sum

Description: Preprocesses an array to quickly answer range sum queries.

Use Cases: Range sum query, finding subarrays with a given sum.

Add a comment...



patterns 12 October
From Create Mode

17. Divide and Conquer

Description: Divides a problem into smaller subproblems, solves them independently, and combines their solutions.

Use Cases: Merge sort, quick sort, binary search.

Add a comment...



patterns 12 October
From Create Mode

19. Monotonic Stack

Description: Used for problems that involve finding the next/previous greater or smaller element in an array.

Use Cases: Next greater element, largest rectangle in histogram.

Add a comment...



patterns 12 October
From Create Mode

20. Bit Manipulation

Description: Uses bitwise operators to solve problems efficiently.

Use Cases: Checking if a number is a power of two, counting the number of 1s in binary representation.

Add a comment...

21. Trie

Description: A tree-like data structure that stores a dynamic set of strings, typically used for searching for prefixes.

Use Cases: Auto-complete, spell checker.

LEARN FROM PEERS & EARN WHILE LEARN

Inbox



Registrar KIIT 12:13

to Director, dean.civil, Dean, Dr.Aru...



Dear Students,

This is for the information of B. Tech students (2021 & 2022 Admission batch) that the **"LEARN FROM PEERS"** programme will be conducted for the academic session 2024-25, which will provide a platform for both senior and junior students to learn together.

The emphasis of the programme is mostly on

- a. application of knowledge of concerned subjects learnt from the classroom teaching and
- b. learning through discussion mode.

The Flip Educator (students of 2021 & 2022 admitted batch only) who are interested to teach, may apply in the link given below by **25 October, 2024** :

Link: <https://forms.gle/kR7am9q6RKtxmmTP7>