25 Must-Know DSA Patterns

1. Sliding Window

Used for problems involving subarrays or substrings, often with max/min/average.

2. Two Pointers

Involves two pointers moving through the array, usually for sorted arrays or linked lists.

3. Fast and Slow Pointers

Detects cycles in linked lists or arrays.

4. Merge Intervals

Used to solve overlapping interval problems.

5. Cyclic Sort

Great for problems involving numbers in a range.

6. In-place Reversal of Linked List

For reversing lists or parts of them.

7. Breadth-First Search (BFS)

Level-order traversal, shortest path in unweighted graphs.

8. Depth-First Search (DFS)

Used for exploring all paths or connected components.

9. Two Heaps

Solves problems involving medians or K largest/smallest elements.

10. Subsets

Generates all subsets, permutations, or combinations.

11. Modified Binary Search

Find elements in a sorted array with modifications.

12. Bitwise XOR

Used for problems involving finding the missing number or duplicates.

13. Top K Elements

Frequently uses heaps or quickselect to find the most/least frequent.

14. K-way Merge

Merges multiple sorted arrays using a min-heap.

15. 0/1 Knapsack

Classic DP pattern involving choices with capacity constraints.

16. Topological Sort

Used for ordering tasks based on prerequisites.

17. Trie-based Problems

Great for prefix search or dictionary matching.

18. Backtracking

Try all possibilities recursively and backtrack on failure.

19. Dynamic Programming (DP)

Break problem into overlapping subproblems and cache results.

20. Greedy

Make local optimal choices aiming for a global solution.

21. Union-Find (Disjoint Set)

Used in Kruskals algorithm and connected components.

22. Segment Tree

Solves range query and update problems efficiently.

23. Monotonic Stack

Used for problems involving the next/previous greater/smaller element.

24. Graph - Dijkstras

Shortest path in weighted graphs.

25. Graph - Floyd Warshall / Bellman Ford

All pairs shortest path / negative cycles.