Ultimate DSA Cheat Sheet

Ultimate DSA Cheat Sheet

Based on Input Clues

If array is sorted

- Binary Search
- Two Pointers
- Sliding Window
- Modified Binary Search

If linked list is given

- Fast & Slow Pointers
- In-place Reversal
- Merge Sort
- Cycle Detection (Floyds Algorithm)

If tree is given

- DFS (Pre, In, Post order)
- BFS (Level Order)
- Recursion or Stack

If graph is given

- DFS / BFS
- Union-Find
- Topological Sort (DAG)
- Dijkstra (Shortest path)
- Bellman-Ford (Handles negative weights)
- Floyd-Warshall (All-pairs shortest path)

If matrix/grid is given

- DFS / BFS

Ultimate DSA Cheat Sheet

- Queue (Shortest path in grid)

If recursion is not allowed

- Stack
- Iterative DP

If must solve in-place

- Two Pointers
- Swap values directly
- Bit manipulation

If asked for all combinations / permutations / subsets

- Backtracking
- DFS
- Bitmasking
- BFS with queue

If asked for max/min subarray/subset/result

- Kadanes Algorithm
- Two Pointers
- Dynamic Programming
- Sliding Window

If asked for top K / least K elements

- Min/Max Heap
- Quickselect

If asked for common prefixes/strings

- HashMap / HashSet
- Trie

Ultimate DSA Cheat Sheet

If given a range query

- Prefix Sum
- Difference Array
- Segment Tree
- Fenwick Tree

Common Problem Patterns

Two Pointers - Sorted arrays, remove duplicates, pairs

Sliding Window - Subarrays/substrings with condition

Fast & Slow Pointers - Cycle detection, middle of list

BFS - Shortest path in graph/grid, level order

DFS - Explore tree/graph deeply

Backtracking - Permutations, combinations, Sudoku

Dynamic Programming - Substructure + Overlapping subproblems

Union-Find - Graph components, detect cycles

Greedy - Activity selection, interval scheduling

Bit Manipulation - Unique numbers, subsets

Heap - Top K elements, priority queue

Monotonic Stack - Next greater/smaller element

Trie - Prefix search, autocomplete

Segment Tree - Range min/max/sum queries

Topological Sort - Task scheduling, dependency order

Bonus Patterns

- Cyclic Sort Numbers in range 1 to N
- K-way Merge Merge K sorted lists (use Heap)
- Two Heaps Median of a stream
- Line Sweep Interval overlap, events
- Rabin-Karp / KMP Pattern matching

Ultimate DSA Cheat Sheet

- Memoization Optimize recursive problems

- Tabulation Bottom-up DP

If Youre Stuck

1. Is the input sorted?
2. Are duplicates allowed?
3. Is the structure linear or non-linear?
4. Do I need all solutions or just one?
5. Is recursion allowed?
6. Do I need to optimize space/time?