■ Data Structures & Algorithms (DSA) Handbook

1. Arrays & Strings

Arrays are contiguous memory blocks storing elements of the same type. Strings are arrays of characters. Problems include searching, substring, rotation, and manipulation.

2. Linked List

Linear data structure where elements (nodes) are linked using pointers. Types: singly, doubly, circular. Applications: dynamic memory allocation, queues, adjacency lists in graphs.

3. Stack & Queue

Stack (LIFO) allows operations at one end only. Queue (FIFO) processes elements in order. Variants include deque, priority queue, circular queue.

4. Hashing & Heap

Hashing maps keys to values in constant time. Heap is a complete binary tree for efficient min/max retrieval. Used in priority queues and heapsort.

5. Recursion & Backtracking

Recursion is a method where a function calls itself. Backtracking explores all possibilities but discards invalid paths. Examples: N-Queens, Sudoku.

6. Searching & Sorting

Searching includes linear and binary search. Sorting includes comparison-based (Quick, Merge, Heap) and non-comparison-based (Counting, Radix).

7. Mathematical Algorithms

Includes GCD, LCM, prime number sieves, modular arithmetic, and fast exponentiation. Applications in cryptography and number theory.

8. Bitwise Operations & Tricks

Using AND, OR, XOR, and shifts for optimization. Examples: odd/even check, subset generation, DP with bitmasks.

9. Greedy Algorithm

Solves problems by making locally optimal choices. Works when greedy-choice property holds. Examples: activity selection, Huffman coding.

10. Dynamic Programming

Breaks problems into overlapping subproblems and stores solutions. Classic problems: Fibonacci, Knapsack, LCS.

11. Advanced DP Techniques

Includes DP on trees, Bitmask DP, Digit DP, state compression, and optimization tricks.

12. Graph Algorithm

Basic traversals and shortest path algorithms like BFS, DFS, Dijkstra, and Bellman-Ford.

13. Advanced Graph Algorithms

Includes topological sort, strongly connected components, bridges, articulation points, Euler/Hamilton paths.

14. DSU & MST

Disjoint Set Union for efficient merging/cycle detection. Minimum Spanning Tree algorithms like Kruskal and Prim.

15. Network Flow Algorithm

Max flow problems solved with Ford-Fulkerson, Edmonds-Karp, and Dinic's algorithm. Applications in bipartite matching.

16. String Algorithms

Pattern matching (KMP, Rabin-Karp, Z-function), suffix arrays, suffix automaton. Used in text search and bioinformatics.

17. Computational Geometry

Convex hull, line sweep, closest pair of points. Applications in graphics, GIS, and robotics.

18. Segment Trees & Fenwick Tree

Segment trees handle range queries with updates. Fenwick tree is simpler for prefix sums.

19. Sparse Table & Binary Lifting

Sparse table supports static range queries. Binary lifting helps with LCA queries in trees.

20. Advanced Tree Techniques

Includes Heavy-Light decomposition, centroid decomposition, and Euler tour for efficient queries.

21. Tries & Suffix Trees

Trie is a prefix tree for fast lookup. Suffix trees handle substring queries efficiently.

22. Game Theory & Nim Game

Mathematical modeling of games. Nim game and Grundy numbers are common concepts.

23. Approximation & Randomized Algorithms

Approximation for NP-hard problems and randomized approaches like Monte Carlo, Las Vegas, and randomized QuickSort.