
1. Foundational Concepts (Big O + Complexity + Basics)

- What is **time and space complexity**
- **Asymptotic notation** — Big-O, Big-Ω, Big-Θ
- What is **quadratic time complexity**
- **Complexity of recursion**
- **Complexity of binary search, bubble sort, BST insertion**
- **Downsides of dynamic memory allocation**
- **Static vs dynamic memory allocation**
- **Divide and Conquer** technique
- **Advantages & disadvantages of recursion**

2. Arrays and Variants

- What is a **linear data structure**
- **Jagged array** (irregular multidimensional array)
- **Sparse array**
- **Convert string to Title Case** (string manipulation)
- **Remove duplicates from string using hash table**
- **Remove duplicate elements from linked list** (extension)

3. Linked Lists

- What is a **linked list**
- **Hierarchical linked list**
- **Applications of singly & doubly linked list**
- **Remove odd-value nodes** (practical problem)

4. Stack and Queue

- What is a **stack and queue**
 - **Applications of stack and queue**
 - **Priority Queue** — how it works and use cases
-

5. Trees

- What is a **Tree**
 - **Degree of a node**
 - **Full tree, Perfect tree, Complete tree**
 - **Balanced tree**
 - **AVL tree**
 - **Red-Black tree**
 - **Segment tree**
 - **B-Tree** (used in databases & indexing)
 - **Min node in BST** — code example
 - **Height of BST** — concept and code
 - **Create sorted array from BST**
 - **Diffing algorithm** (used in React, related to trees)
-

6. Graphs

- **Graph representation:** adjacency list vs adjacency matrix
- **Degree of a node** (in graph)
- **Complete graph**
- **BFS and DFS — traversal algorithms**
 - Time complexity of BFS
 - Time complexity of DFS
- **Shortest Path Algorithms**
 - Dijkstra's algorithm
 - Bellman-Ford algorithm
 - A* algorithm
 - Tarjan's algorithm (strongly connected components)
- **Minimum Spanning Tree**
 - Kruskal's algorithm
 - Prim's algorithm
 - Applications of MST
- **Graph indexing** (how databases index graph structures)

7. Hashing & Hash Tables

- **Hash table** — concept and applications
 - **Load factor**
 - **Collision resolution techniques:**
 - Separate chaining
 - Linear probing
 - Quadratic probing
 - Double hashing
 - Rehashing
 - **Applications of hash tables**
 - **Array vs Hash Table**
 - **Remove duplicates using Hash Table**
-

8. Complexity & Performance Analysis

- **Big-O recap** with examples ($O(1)$, $O(n)$, $O(\log n)$, $O(n^2)$)
 - **Best / Average / Worst case** (e.g., QuickSort, Binary Search)
 - **Complexity of BST insertion**
 - **Recursion complexity patterns** ($T(n) = 2T(n/2) + n$)
 - **Tradeoffs between recursion, iteration, and dynamic programming**
-

9. Advanced DSA Topics

- **Heap / Priority Queue** (Min Heap, Max Heap)
 - **Spanning Tree and Minimum Spanning Tree** (used in networking & graphs)
 - **Backtracking**
 - **Monotonic Stack**
 - **Collision handling** (rehashing, probing methods)
 - **Full vs Perfect vs Complete binary trees** (comparisons)
 - **Hierarchical data structures** (trees, tries, etc.)
-

10. Data Structures in Databases (Cross Concept with MongoDB/NodeJS)

- Which **data structure is used behind indexing** → B-Tree, B+Tree, Hash Index
 - **Graph indexing**
 - **Trie / Prefix Tree** — used in autocomplete & search
 - **Referential equality** — used in comparing complex structures
-

✅ **Summary Path to Follow:**

1. Complexity →
 2. Arrays →
 3. Linked List →
 4. Stack & Queue →
 5. Trees →
 6. Graphs →
 7. Hash Tables →
 8. Algorithms (Recursion, Divide & Conquer, Backtracking, DP)
-