**Data Structures**

1. **Arrays**
   - Sliding Window Technique
   - Two Pointers
   - Prefix Sum
   - Kadane’s Algorithm (Maximum Subarray Sum)
2. **Strings**
   - String Manipulation
   - Rabin-Karp Algorithm (Pattern Matching)
   - KMP Algorithm (String Matching)
   - Palindrome Checkers
   - Longest Common Subsequence (LCS)
3. **Linked Lists**
   - Singly and Doubly Linked Lists
   - Cycle Detection (Floyd’s Tortoise and Hare Algorithm)
   - Merge Two Sorted Lists
   - Reverse a Linked List (Iterative and Recursive)
   - Remove N-th Node from End of List
4. **Stacks and Queues**
   - Next Greater Element
   - Valid Parentheses
   - Min/Max Stack
   - Circular Queue
   - Sliding Window Maximum
5. **Trees**
   - Binary Tree Traversals (Preorder, Inorder, Postorder)
   - Lowest Common Ancestor (LCA)
   - Binary Search Tree (BST) Operations
   - Serialize and Deserialize Binary Trees
   - Tree Diameter
6. **Graphs**
   - BFS and DFS (Iterative and Recursive)
   - Topological Sort
   - Dijkstra’s Algorithm
   - Bellman-Ford Algorithm
   - Floyd-Warshall Algorithm
   - Minimum Spanning Tree (Prim’s and Kruskal’s Algorithms)
7. **Hashing**
   - HashMap/HashSet Usage
   - Collision Resolution Techniques
   - Two Sum Problem
   - Subarray Sum Equals K
   - Longest Consecutive Sequence
8. **Heaps and Priority Queues**
   - Min-Heap and Max-Heap
   - Kth Largest Element
   - Median Finder
   - Merge K Sorted Lists

9. **Tries**
    - Word Search
    - Prefix Matching
    - Auto-complete System
10. **Disjoint Set Union (Union-Find)**
    - Path Compression and Union by Rank
    - Detect Cycles in Graphs
    - Kruskal’s Algorithm
11. **Segment Trees and Fenwick Trees**
    - Range Queries (Sum, Minimum, Maximum)
    - Point Updates
    - Lazy Propagation
12. **Matrix**
    - Spiral Order Traversal
    - Rotate Matrix
    - Search in a Sorted Matrix
    - Island Problems (Connected Components)

---

## Algorithms

1. **Sorting Algorithms**
    - Quick Sort, Merge Sort, Heap Sort
    - Counting Sort, Radix Sort
    - Bucket Sort
2. **Searching Algorithms**
    - Binary Search and Variants
    - Exponential Search
    - Ternary Search
3. **Dynamic Programming**
    - 0/1 Knapsack Problem
    - Longest Increasing Subsequence
    - Matrix Chain Multiplication
    - Edit Distance
    - Coin Change
    - Rod Cutting Problem
4. **Greedy Algorithms**
    - Activity Selection Problem
    - Huffman Encoding
    - Job Scheduling Problem
    - Interval Partitioning
5. **Backtracking**
    - N-Queens Problem
    - Sudoku Solver
    - Word Search
    - Permutations and Combinations
6. **Divide and Conquer**
    - Merge Sort
    - Quick Sort
    - Closest Pair of Points

7. **Bit Manipulation**
    - Single Number
    - Power of Two
    - Count Set Bits (Hamming Weight)
    - XOR Queries
8. **Mathematical Algorithms**
    - Greatest Common Divisor (GCD)
    - Sieve of Eratosthenes (Prime Numbers)
    - Modular Arithmetic
    - Fast Exponentiation
9. **Graph Algorithms**
    - Shortest Path Algorithms
    - Network Flow (Edmonds-Karp Algorithm)
    - Eulerian and Hamiltonian Paths
    - Articulation Points and Bridges

---

**Practice Tips**

- Focus on problem-solving using **LeetCode**, **HackerRank**, **Codeforces**, **GeeksforGeeks**, and **CodeChef**.
- Practice system design for senior roles (Scalability, Databases, Caching, APIs).
- Use mock interviews to assess your readiness.

Would you like me to prioritize any specific topic or provide resources for them?