List of CS Algorithms and Data Structures to Master

Data Structures:

1. Arrays: A collection of elements identified by index or key. Used for storing multiple items of the same type together.
2. Linked Lists: A linear collection of elements, where each element points to the next. Useful for dynamic memory allocation.
3. Stacks: A collection of elements with Last In First Out (LIFO) access. Used in function call management and expression evaluation.
4. Queues: A collection of elements with First In First Out (FIFO) access. Used in scheduling and buffering.
5. Hash Tables: A data structure that maps keys to values using a hash function. Used for efficient data retrieval.
6. Trees: A hierarchical data structure with nodes connected by edges. Types include Binary Trees, Binary Search Trees, AVL Trees, etc.
7. Heaps: A special tree-based structure that satisfies the heap property. Used in priority queues and heap sort.
8. Graphs: A collection of nodes connected by edges. Used in network analysis, pathfinding, and many algorithms.
9. Tries: A tree-like data structure used for storing strings in a space-efficient manner. Useful in search operations.
10. Sets: A collection of unique elements. Useful for membership testing and eliminating duplicates.

Algorithms:

1. Bubble Sort: A simple comparison-based sorting algorithm. Repeatedly steps through the list, compares adjacent elements and swaps them if they are in the wrong order.
2. Selection Sort: An in-place comparison-based sorting algorithm. Divides the input list into two parts: a sorted sublist and an unsorted sublist, and repeatedly selects the smallest element from the unsorted sublist.
3. Insertion Sort: Builds the final sorted array one item at a time. Suitable for small data sets and mostly sorted arrays.
4. Merge Sort: A divide-and-conquer algorithm that divides the array into halves, sorts them, and then merges the sorted halves.
5. Quick Sort: A divide-and-conquer algorithm that selects a pivot and partitions the array into two subarrays based on the pivot, then recursively sorts the subarrays.
6. Linear Search: A simple search algorithm that checks every element in the list until the target value is found or the list ends.
7. Binary Search: A search algorithm that finds the position of a target value within a sorted array by repeatedly dividing the search interval in half.
8. Depth-First Search (DFS): An algorithm for traversing or searching tree or graph data structures. Starts at the root and explores as far as possible along each branch before backtracking.
9. Breadth-First Search (BFS): An algorithm for traversing or searching tree or graph data structures. Starts at the root and explores all the neighboring nodes at the present depth before moving on to nodes at the next depth level.
10. Knapsack Problem: A problem in combinatorial optimization. Given a set of items, each with a weight and a value, determine the number of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.
11. Longest Common Subsequence: Finds the longest subsequence common to all sequences in a set of sequences. Used in bioinformatics and text comparison.
12. Dijkstra's Algorithm: An algorithm for finding the shortest paths between nodes in a graph. Used

in routing and navigation.

13. Kruskal's Algorithm: An algorithm for finding the minimum spanning tree for a connected weighted graph. Used in network design.

14. Prim's Algorithm: Another algorithm for finding the minimum spanning tree for a connected weighted graph. Greedy algorithm that builds the MST one edge at a time.

15. N-Queens Problem: A backtracking problem to place N chess queens on an N×N chessboard so that no two queens threaten each other.

16. Sudoku Solver: A backtracking algorithm to solve Sudoku puzzles by filling empty cells with digits without violating the rules of Sudoku.

17. Floyd-Warshall Algorithm: A graph analysis algorithm for finding shortest paths in a weighted graph with positive or negative edge weights.

18. Bellman-Ford Algorithm: An algorithm for finding shortest paths in a weighted graph. Handles graphs with negative edge weights.

19. KMP Algorithm: A string matching algorithm that searches for occurrences of a word within a main text string by employing the observation that when a mismatch occurs, the word itself embodies sufficient information to determine where the next match could begin.

20. Rabin-Karp Algorithm: A string searching algorithm that uses hashing to find any one of a set of pattern strings in a text.

21. Boyer-Moore Algorithm: An efficient string searching algorithm that skips sections of the text and reduces the number of comparisons needed.

22. Bit Manipulation: Algorithms that use bitwise operations to solve problems. Efficient for low-level programming and optimization tasks.

23. Prime Numbers: Algorithms related to finding prime numbers, such as the Sieve of Eratosthenes.

24. Greatest Common Divisor (GCD): Algorithms to find the largest number that divides two or more numbers without leaving a remainder, such as the Euclidean algorithm.

25. Least Common Multiple (LCM): Algorithms to find the smallest number that is a multiple of two or more numbers.