### ****Step-by-Step Algorithm Learning Path****

## ****Step 1: Sorting Algorithms (Fundamentals First)****

1. **Bubble Sort**
2. **Selection Sort**
3. **Insertion Sort**
4. **Merge Sort** (Divide & Conquer)
5. **Quick Sort** (Divide & Conquer)
6. **Heap Sort**
7. **Counting Sort**
8. **Radix Sort**
9. **Bucket Sort**

## ****Step 2: Searching Algorithms (Finding Data Efficiently)****

1. **Linear Search** (Brute-force search)
2. **Binary Search** (Divide & Conquer, requires sorted data)
3. **Binary Search Variations (Lower Bound, Upper Bound, etc.)**
4. **Jump Search**
5. **Interpolation Search**
6. **Exponential Search**

## ****Step 3: Recursion & Divide and Conquer****

1. **Binary Search (Recursive)**
2. **Merge Sort (Recursive)**
3. **Quick Sort (Recursive)**
4. **Tower of Hanoi**
5. **Backtracking (N-Queens, Sudoku Solver, etc.)**

## ****Step 4: Graph Algorithms (BFS, DFS, Shortest Path, MST)****

1. **Breadth-First Search (BFS)**
2. **Depth-First Search (DFS)**
3. **Dijkstra’s Algorithm (Shortest Path in Weighted Graphs)**
4. **Bellman-Ford Algorithm**
5. **Floyd-Warshall Algorithm**
6. A Algorithm\*
7. **Topological Sorting (Kahn's Algorithm, DFS-Based)**
8. **Kruskal’s Algorithm (Minimum Spanning Tree - MST)**
9. **Prim’s Algorithm (MST)**

## ****Step 5: Dynamic Programming (Optimizing Recursive Solutions)****

1. **Fibonacci (Recursion vs. Memoization vs. Bottom-Up DP)**
2. **0/1 Knapsack Problem**
3. **Longest Common Subsequence (LCS)**
4. **Longest Increasing Subsequence (LIS)**
5. **Coin Change Problem**
6. **Rod Cutting Problem**
7. **Edit Distance Algorithm**

## ****Step 6: Greedy Algorithms****

1. **Activity Selection Problem**
2. **Huffman Encoding**
3. **Dijkstra’s Algorithm (Also under Graphs, but Greedy)**
4. **Prim’s Algorithm (Also under MST, but Greedy)**
5. **Job Scheduling Problem**

## ****Step 7: Bit Manipulation****

1. **Check if a Number is Power of 2**
2. **Count Set Bits in an Integer (Brian Kernighan’s Algorithm)**
3. **Find the Single Non-Repeating Element (XOR trick)**
4. **Swap Two Numbers Without Using a Temporary Variable**

## ****Step 8: Mathematical Algorithms****

1. **Greatest Common Divisor (GCD) – Euclidean Algorithm**
2. **Least Common Multiple (LCM)**
3. **Sieve of Eratosthenes (Prime Numbers up to N)**
4. **Modular Exponentiation**
5. **Fast Exponentiation (Binary Exponentiation)**

## ****Step 9: String Algorithms****

1. **KMP Algorithm (Pattern Matching)**
2. **Rabin-Karp Algorithm (Pattern Matching)**
3. **Z-Algorithm (Pattern Matching)**
4. **Trie (Prefix Tree) Data Structure**
5. **Aho-Corasick Algorithm (Multiple Pattern Matching)**
6. **Manacher’s Algorithm (Longest Palindromic Substring)**

## ****Step 10: Advanced Graph Algorithms****

1. **Tarjan’s Algorithm (Strongly Connected Components - SCCs)**
2. **Kosaraju’s Algorithm (SCCs)**
3. **Johnson’s Algorithm (All-Pairs Shortest Path in Sparse Graphs)**
4. **Borůvka’s Algorithm (Minimum Spanning Tree - Alternative to Kruskal/Prim)**

## ****Step 11: Advanced Dynamic Programming & Combinatorics****

1. **Kadane’s Algorithm (Maximum Subarray Sum)**
2. **Matrix Chain Multiplication**
3. **Egg Dropping Problem**
4. **Bellman-Held-Karp Algorithm (Traveling Salesman Problem - TSP, DP approach)**

## ****Step 12: Miscellaneous Algorithms (Final Set)****

1. **Floyd’s Cycle Detection Algorithm (Tortoise and Hare for Linked Lists)**
2. **Reservoir Sampling**
3. **Fisher-Yates Shuffle Algorithm**
4. **Union-Find (Disjoint Set Union - DSU) with Path Compression**
5. **Mo’s Algorithm (Answering Range Queries Efficiently)**
6. **Suffix Array & Suffix Tree Construction**