To learn Data Structures and Algorithms (DSA) from basics to advance in a structured way, here's a roadmap divided into levels:

**Level 1: Basics of Programming and Core Concepts**

1. **Getting Started with Java**
   * Variables, Data Types, and Input/Output
   * Conditionals and Loops
   * Functions and Recursion
2. **Mathematics for DSA**
   * Prime Numbers and Sieve of Eratosthenes
   * Greatest Common Divisor (GCD) and Least Common Multiple (LCM)
   * Modular Arithmetic
   * Bit Manipulation Basics
3. **Arrays**
   * Basics and Traversals
   * Searching and Sorting (Binary Search, Bubble Sort, Quick Sort)
   * Prefix Sum and Sliding Window Techniques
4. **Strings**
   * String Manipulation
   * Pattern Matching (Brute Force, KMP, Rabin-Karp)

**Level 2: Fundamental Data Structures**

1. **Linked List**
   * Singly and Doubly Linked List
   * Basic Operations (Insertion, Deletion, Reversal)
   * Cycle Detection (Floyd’s Algorithm)
2. **Stacks and Queues**
   * Stack and Queue Implementation (Array/Linked List)
   * Applications (Parenthesis Matching, LRU Cache)
   * Deque and Priority Queue
3. **Hashing**
   * Hash Maps and Hash Sets
   * Applications (Frequency Counting, Two-Sum Problem)
   * Collision Handling (Chaining, Open Addressing)
4. **Binary Trees**
   * Tree Traversals (Inorder, Preorder, Postorder, Level Order)
   * Binary Search Tree (BST) Basics
   * Lowest Common Ancestor (LCA), Diameter of Tree

**Level 3: Intermediate Algorithms**

1. **Divide and Conquer**
   * Merge Sort and Quick Sort
   * Binary Search Variants
   * Median of Two Sorted Arrays
2. **Dynamic Programming (DP) Basics**
   * Fibonacci Sequence (Recursive and Iterative)
   * Longest Common Subsequence (LCS)
   * 0/1 Knapsack Problem
3. **Greedy Algorithms**
   * Activity Selection
   * Huffman Coding
   * Minimum Spanning Tree (Kruskal’s/Prim’s Algorithm)
4. **Graphs**
   * Representation (Adjacency Matrix, List)
   * BFS and DFS
   * Shortest Paths (Dijkstra, Bellman-Ford)
   * Topological Sort

**Level 4: Advanced Topics**

1. **Advanced Trees**
   * Segment Trees
   * Fenwick Tree (Binary Indexed Tree)
   * Trie Data Structure
2. **Advanced Graph Algorithms**
   * Union-Find (Disjoint Set Union)
   * Tarjan’s Algorithm (Strongly Connected Components)
   * Floyd-Warshall Algorithm
3. **Dynamic Programming (Advanced)**
   * DP on Trees
   * DP with Bitmasking
   * Advanced Matrix Chain Multiplication
4. **Backtracking**
   * N-Queens Problem
   * Sudoku Solver
   * Word Search in a Matrix
5. **Mathematics (Advanced)**
   * Combinatorics
   * Modular Exponentiation
   * Fast Fourier Transform (FFT)

**Level 5: Competitive Programming and Problem Solving**

* Mixed problem-solving on platforms like:
  + **LeetCode**: Arrays, Strings, Binary Search, DP
  + **Codeforces**/**CodeChef**: Graphs, Math, Advanced DP
  + **HackerRank**/**GeeksforGeeks**: Implementation and practice