**🚀 Important Topics to Focus on First**

Here’s a structured and logical order for these topics to facilitate better understanding:

### ****1. Introduction to Data Structures and Algorithms (DSA)****

* Definition and Importance of DSA
* Role of DSA in Programming and Problem Solving
* Differences Between Data Structures and Algorithms

### ****2. Types of Data Structures****

* Linear vs. Non-Linear Data Structures
* Primitive vs. Non-Primitive Data Structures
* Hierarchical Data Structures

### ****3. Concept of Complexity Analysis****

* Why Complexity Analysis is Important
* Time Complexity vs. Space Complexity
* Understanding Algorithm Efficiency

### ****4. Asymptotic Analysis****

* Big-O, Big-Theta, and Big-Omega Notations
* Best, Worst, and Average Case Analysis
* Time Complexity of Common Algorithms

### ****5. Concepts of Arrays****

* Single-Dimensional and Multi-Dimensional Arrays
* Jagged Arrays and Sparse Arrays
* Operations on Arrays (Insertion, Deletion, Traversal)

### ****6. Concept of Strings****

* How Strings are Stored in Memory
* String Operations (Reversal, Concatenation, Searching)
* Common String Algorithms

### ****7. Concept of Linked Lists****

* Singly, Doubly, and Circular Linked Lists
* Applications and Advantages of Linked Lists
* Common Linked List Problems

### ****8. Linear Search and Binary Search****

* Difference Between Linear and Binary Search
* Time Complexity of Both Algorithms
* Recursive and Iterative Implementations
* Applications in Real-World Scenarios

### ****9. Concepts of Recursion****

* What is Recursion?
* Direct and Indirect Recursion
* Applications of Recursion in Problem Solving

### ****10. Applications of All Data Structures****

* Where and Why to Use Linked Lists, Arrays, Trees, Graphs, etc.
* Real-World Use Cases

### ****11. Memory Allocation and Memory Leak****

* Static vs. Dynamic Memory Allocation
* How Memory Leaks Happen and How to Prevent Them
* Stack vs. Heap Memory

### ****Data Structures & Algorithms****

#### **1. Linked Lists**

* Singly Linked List
* Doubly Linked List
* Circular Linked List
* Circular Doubly Linked List
* Reverse a Linked List
* Remove Duplicates from a Linked List
* Detect Cycle in a Linked List (Floyd’s Algorithm)
* Remove Middle Element from a Linked List in O(n) time
* Remove Nth Node from the End of the List (Fast & Slow Pointer)
* Merge Two Sorted Linked Lists
* Delete Function for Doubly Linked List
* Insert a Node After a Node (Check All Conditions)
* Get Middle Element in a Linked List
* Remove Odd Element Nodes from a Linked List
* Convert an Array to a Linked List
* Print Linked List in Order and Reverse Order
* Applications of Linked Lists
* Drawbacks of Linked Lists

#### **2. Arrays**

* Find Second Largest Element in an Array (Handle Negative Numbers)
* Find Kth Largest Element in an Array
* Third Largest Element in an Array
* Sum of an Array Using Recursion
* Reverse an Array
* Convert an Array to a Linked List
* Find Minimum in a Sorted Rotated Array
* Longest Substring Without Repeating Characters
* Product Except Itself
* Subarray with Maximum Elements in Increasing Order
* Remove a Specific Element from an Array
* Flatten a Multidimensional Array
* Jagged Arrays
* Sparse Arrays
* Array Operations Complexity

#### **3. Recursion**

* Binary Recursion
* Tail Recursion vs. Head Recursion
* Direct vs. Indirect Recursion
* Applications of Recursion
* Fibonacci Series Using Recursion
* Factorial Using Recursion
* Remove a Character from a String Using Recursion
* Recursively Remove All Occurrences of a Character from a String
* Recursive Binary Search
* Sum of an Array Using Recursion
* Print First N Elements of Fibonacci Series Using Recursion
* Recursion that Recurses Only 5 Times
* Advantages and Disadvantages of Recursion
* Space Complexity of Recursion

#### **4. Searching & Sorting**

* Binary Search (Time, Logic, Debug)
* Binary Search Using Recursion
* Time Complexity of Binary Search
* Linear Search vs. Binary Search
* Find Minimum in a Sorted Rotated Array
* Merge Two Sorted Linked Lists

#### **5. Strings**

* Reverse Each Word in a String
* Remove a Character from a String Using Recursion
* Remove All Occurrences of a Specific Character from a String
* Longest Substring Without Repeating Characters
* Largest Substring Without Vowels
* Check if Two Strings are Anagrams
* Extract Numbers from Strings
* Ensure Sentence Begins with Uppercase and Ends with a Period

#### **6. Other Problems**

* Two Sum (LeetCode)
* Valid Parentheses
* Palindrome Code
* Remove Duplicates from a Linked List
* Detect Cycle in a Linked List
* Find Combination of Two Numbers Given Sum as 4

### ****Memory Management****

#### **1. Memory Allocation**

* Static vs. Dynamic Memory Allocation
* Advantages of Static Memory Allocation
* Disadvantages of Dynamic Memory Allocation
* Memory Leak
* How to Prevent Memory Leak
* Memory Pool
* Stack vs. Heap Memory
* Garbage Collection & Working

#### **2. Virtual Memory**

* Concepts of Virtual Memory
* Relation with Data Structures

### ****Complexity Analysis****

#### **1. Asymptotic Notations**

* Big-O Notation
* Big-Theta Notation
* Big-Omega Notation
* Differences Between n log n and log n
* Time Complexity of Binary Search
* Space Complexity of Recursion
* Complexity of Algorithms
* Best Case in Asymptotic Analysis
* Asymptotic Analysis for Quadratic Equations

#### **2. Time & Space Complexity**

* Time Complexity of Accessing an Element from a Linked List
* Complexity of Array Operations
* Time Complexity of Recursion

### ****Miscellaneous****

#### **1. Data Structure Concepts**

* Linear vs. Non-Linear Data Structures
* Hierarchical Data Structures
* Primitive vs. Non-Primitive Data Types
* Static vs. Dynamic Typing
* Multidimensional Arrays
* Jagged Arrays
* Sparse Arrays
* Contiguous vs. Non-Contiguous Memory Allocation

#### **2. Algorithms**

* Why Need Algorithms?
* Recursion vs. Loop
* Advantages and Disadvantages of Recursion
* Applications of Recursion
* Real-World Applications of Recursion

#### **3. Practice Platforms**

* LeetCode (Blind 75, NeetCode.io)
* HackerRank
* GeeksforGeeks

### ****Pending Tasks****

#### **1. Linked Lists**

* Implement Linked List (Singly, Doubly, Circular)
* Reverse a Doubly Linked List
* Remove Middle Element from a Linked List
* Delete Function for Doubly Linked List
* Detect Cycle in a Linked List
* Merge Two Sorted Linked Lists

#### **2. Arrays**

* Fix Logic for Finding Second Largest Element in an Array
* Find Kth Largest Element in an Array
* Longest Substring Without Repeating Characters
* Subarray with Maximum Elements in Increasing Order

#### **3. Recursion**

* Recursive Binary Search
* Fibonacci Series Using Recursion
* Remove a Character from a String Using Recursion
* Recursively Remove All Occurrences of a Character from a String

#### **4. Complexity Analysis**

* Learn Asymptotic Analysis (Theta Notation)
* Understand Big-O, Big-Theta, and Big-Omega
* Analyze Time and Space Complexity of Algorithms

#### **5. Practice Problems**

* Two Sum (LeetCode)
* Valid Parentheses
* Palindrome Code
* Remove Duplicates from a Linked List
* Detect Cycle in a Linked List

### ****Timeline****

* **High Priority (Feb 2025)**
  + Virtual Memory
  + Memory Pool
  + Binary Recursion
  + Applications of Recursion
  + Third Largest Element in an Array
  + Sum of an Array Using Recursion
  + Insert to a Doubly Linked List
* **Medium Priority (Jan 2025)**
  + Linked List Implementation
  + Binary Search Implementation
  + Merge Two Sorted Linked Lists
  + Remove Nth Node from the End of the List
  + Detect Cycle in a Linked List
  + Find Minimum in a Sorted Rotated Array
* **Low Priority (Dec 2024 & Earlier)**
  + Recursion Workouts
  + Reverse a Doubly Linked List
  + Memory Allocation Concepts
  + Asymptotic Notations
  + Practice Problems from LeetCode, HackerRank, GeeksforGeeks