**Arrays**

1. **Two Sum Problem** - Find two numbers in an array that add up to a target value
2. **Maximum Subarray** - Find the contiguous subarray with the largest sum (Kadane's Algorithm)
3. **Array Rotation** - Rotate an array to the right by k steps
4. **Find Missing Number** - Find the missing number in an array containing 1 to n
5. **Merge Sorted Arrays** - Merge two sorted arrays into one sorted array
6. **Product of Array Except Self** - Calculate product of all elements except the element at current index
7. **Search in Rotated Sorted Array** - Find an element in a rotated sorted array

**Strings**

1. **String Reversal** - Reverse a string without using built-in reverse methods
2. **Valid Anagram** - Check if two strings are anagrams of each other
3. **Longest Palindromic Substring** - Find the longest palindromic substring in a given string
4. **String to Integer (atoi)** - Convert a string to an integer without using Integer.parseInt()
5. **Valid Parentheses** - Check if a string of brackets is valid/balanced

**Collections: List Classes (ArrayList, LinkedList)**

1. **Linked List Cycle Detection** - Detect if a linked list has a cycle (Floyd's Cycle-Finding Algorithm)
2. **Reverse a Linked List** - Reverse a singly linked list
3. **Remove Nth Node from End** - Remove the nth node from the end of a linked list
4. **Merge K Sorted Lists** - Merge k sorted linked lists into one sorted list
5. **Implement LRU Cache** - Implement a Least Recently Used (LRU) cache using LinkedHashMap

**Collections: Set Classes (HashSet, TreeSet)**

1. **Longest Consecutive Sequence** - Find the length of the longest consecutive elements sequence in an unsorted array using HashSet
2. **Remove Duplicates** - Remove duplicates from an array using HashSet
3. **Implement a HashSet** - Implement a HashSet without using built-in hash set libraries

**Collections: Map Classes (HashMap, TreeMap)**

1. **Word Frequency Counter** - Count the frequency of words in a text using HashMap
2. **First Non-Repeating Character** - Find the first non-repeating character in a string using HashMap
3. **Design a Data Structure with O(1) Operations** - Design a data structure that supports insert, delete, getRandom in O(1) time

**Collections: Queue Classes (PriorityQueue, ArrayDeque)**

1. **Sliding Window Maximum** - Find the maximum element in each sliding window of size k using a Deque
2. **Implement Stack using Queues** - Implement a stack using queues

**Bonus Problems (More Advanced)**

1. **Merge Intervals** - Merge overlapping intervals in an array of interval pairs
2. **Top K Frequent Elements** - Find the k most frequent elements in an array using a PriorityQueue
3. **Implement a Thread Pool** - Design a thread pool with a fixed number of threads and a task queue
4. **LFU Cache Implementation** - Implement a Least Frequently Used (LFU) cache
5. **Find Median from Data Stream** - Implement a data structure that supports finding the median of a stream of numbers