

Arrays

Arrays are fundamental data structures storing elements in contiguous memory. Cognizant interviews often include basic array tasks. For example, candidates may be asked to find the largest or second-largest elements in an array ¹ or to compute the maximum element ². Sorting/counting tasks like the “Dutch National Flag” problem (sorting 0s,1s,2s) are also common. Below are representative array problems (with difficulties from easy to medium) that freshers should practice.

- **Find Second Largest Element in an Array** (Easy, GeeksforGeeks) – [Link](#)
- **Find the Missing Number** (Easy, GeeksforGeeks) – [Link](#)
- **Sort an Array of 0s, 1s, and 2s (Dutch National Flag Problem)** (Easy, GeeksforGeeks) – [Link](#)
- **Maximum Subarray Sum (Kadane's Algorithm)** (Medium, GeeksforGeeks) – [Link](#)

Strings

Strings are a key topic with many straightforward tasks. Common problems include reversing a string or sorting its characters, as well as finding unique characters or checking anagrams ¹ ³. For example, Cognizant interviewees have reported being asked to reverse a string and to sort the letters of a string ¹ ³. Below are classic string problems to practice (easy difficulty).

- **Reverse a String** (Easy, GeeksforGeeks) – [Link](#)
- **First Non-Repeating Character in a String** (Easy, GeeksforGeeks) – [Link](#)
- **Check if Two Strings are Anagrams** (Easy, GeeksforGeeks) – [Link](#)

Hashing

Hashing-based problems use maps or sets to count and track values. Typical freshers-level questions include “Two Sum” (find a pair with a given sum), subarray-sum problems (e.g. **Subarray Sum Equals K**), or grouping anagrams. These problems test using hash tables for O(1) lookups and frequency counts. Example problems (Easy/Medium) are listed below, all of which have Java/C++ solutions available.

- **Two Sum** (Easy, LeetCode) – [Link](#)
- **Group Anagrams** (Medium, LeetCode) – [Link](#)
- **Subarray Sum Equals K** (Medium, LeetCode) – [Link](#)

Recursion

Recursive problems require defining a base case and calling the function on smaller inputs. A classic example is the Tower of Hanoi puzzle ⁴. Other common tasks include computing recursive sequences like the Fibonacci numbers. Below are typical recursion problems (easy difficulty) one might see.

- **Tower of Hanoi** (Easy, GeeksforGeeks) – [Link](#)
- **Nth Fibonacci Number (Using Recursion)** (Easy, GeeksforGeeks) – [Link](#)

Sorting

Sorting and divide-and-conquer tasks are often examined. Candidates should know basic sorting algorithms (quicksort, mergesort, etc.) and application problems. For instance, interview prep sheets list sorting questions explicitly ⁵. The problems below involve sorting or selection (some medium difficulty).

- **Sort an Array** (Medium, LeetCode) – [Link](#)
- **Kth Largest Element in an Array** (Medium, LeetCode) – [Link](#)

Searching

Searching problems include implementing binary search and its variants. For example, freshers should practice binary search on a sorted array and also “search in a rotated sorted array” (a common twist) ⁵. The problems below cover basic to medium-level search tasks.

- **Binary Search** (Easy, GeeksforGeeks) – [Link](#)
- **Search in Rotated Sorted Array** (Medium, LeetCode) – [Link](#)

Linked List

Linked lists are linear structures where each node points to the next. They allow efficient insertions/deletions and underlie stacks/queues ⁶. Typical interview problems include reversing a list, merging sorted lists, and cycle detection. The list below contains standard linked list questions (easy difficulty).

- **Reverse Linked List** (Easy, LeetCode) – [Link](#)
- **Merge Two Sorted Lists** (Easy, LeetCode) – [Link](#)
- **Linked List Cycle (Detect a Cycle)** (Easy, LeetCode) – [Link](#)

Stack & Queue

Stacks (LIFO) and queues (FIFO) are fundamental. Common questions include checking for balanced parentheses (stack use), next-greater-element (using a stack), or implementing one structure with the other

(e.g. a queue via two stacks). The problems below illustrate typical stack/queue tasks (easy difficulty) that have known Java/C++ solutions.

- **Valid Parentheses** (Easy, LeetCode) – [Link](#)
- **Next Greater Element I** (Easy, LeetCode) – [Link](#)

Basic Trees

Tree problems often involve traversals and simple properties. A basic tree is a hierarchical structure of nodes connected by edges ⁷. Freshers should practice level-order (BFS) traversal and common ancestor queries. For example, level-order traversal and lowest common ancestor problems frequently appear. The problems below cover easy-to-medium tree questions.

- **Binary Tree Level Order Traversal** (Easy, LeetCode) – [Link](#)
- **Lowest Common Ancestor of a Binary Tree** (Medium, LeetCode) – [Link](#)
- **Validate Binary Search Tree** (Medium, LeetCode) – [Link](#)

Sources: The above problems and categories are based on common DSA topics and multiple Cognizant FSE/Digital Nurture interview experiences ¹ ³ ⁶ ⁵ ⁴, as well as standard practice platforms (GeeksforGeeks, LeetCode) for Java/C++ solutions.

-
- ¹ Cognizant Interview Experience | Digital Nurture 3.0 - Java FSE - GeeksforGeeks
<https://www.geeksforgeeks.org/interview-experiences/cognizant-interview-experience-digital-nurture-3-o-java-fse/>
 - ² Cognizant Digital Nurture 2.0 Interview Experience 2022 - GeeksforGeeks
<https://www.geeksforgeeks.org/interview-experiences/cognizant-digital-nurture-2-0-interview-experience-2022/>
 - ³ Cognizant Digital Nurture Interview Experience (ON-CAMPUS) - GeeksforGeeks
<https://www.geeksforgeeks.org/interview-experiences/cognizant-digital-nurture-interview-experience-on-campus/>
 - ⁴ Program for Tower of Hanoi Algorithm - GeeksforGeeks
<https://www.geeksforgeeks.org/dsa/c-program-for-tower-of-hanoi/>
 - ⁵ Cognizant SDE Sheet: Interview Questions and Answers - GeeksforGeeks
<https://www.geeksforgeeks.org/dsa/cognizant-sde-sheet-interview-questions-and-answers/>
 - ⁶ Linked List Data Structure - GeeksforGeeks
<https://www.geeksforgeeks.org/dsa/linked-list-data-structure/>
 - ⁷ Tree Data Structure - GeeksforGeeks
<https://www.geeksforgeeks.org/dsa/tree-data-structure/>