

# Common Java Coding Questions for Interviews

## ### 1. Array and String Manipulation

1. Write a program to find the largest and smallest number in an unsorted array.
2. Given an array of integers, write a method to move all zeroes to the end of the array while maintaining the relative order of the non-zero elements.
3. Write a function to rotate an array by k positions.
4. Write a Java program to find all pairs of integers in an array whose sum is equal to a given number.
5. Implement a function to find the most frequently occurring element in an array.
6. Write a program to reverse a string without using any built-in reverse functions.
7. Check if two given strings are anagrams of each other.
8. Find the first non-repeating character in a string.
9. Write a Java program to determine if a string contains only unique characters.
10. Implement a function to perform basic string compression using the counts of repeated characters.

## ### 2. Searching and Sorting

11. Implement the binary search algorithm on a sorted array.
12. Write a program to sort an array using the merge sort algorithm.
13. Implement quick sort in Java.
14. Write a program to find the kth smallest element in an unsorted array.
15. Given a sorted array, remove all duplicates in place such that each element appears only once.

## ### 3. Linked Lists

16. Write a program to reverse a linked list.
17. Find the middle node of a linked list in a single pass.
18. Detect if a linked list contains a cycle. If so, find the starting node of the cycle.

19. Merge two sorted linked lists into one sorted linked list.

20. Remove the nth node from the end of a linked list.

#### ### 4. Stacks and Queues

21. Implement a stack using an array and linked list.

22. Implement a queue using two stacks.

23. Write a program to evaluate a postfix expression using a stack.

24. Write a program to implement a min stack that supports push, pop, top, and retrieving the minimum element in constant time.

25. Implement a program to check if a given string of parentheses is balanced using a stack.

#### ### 5. Trees and Binary Search Trees

26. Write a program to perform an in-order traversal of a binary tree (iteratively and recursively).

27. Implement a function to check if a binary tree is a valid binary search tree (BST).

28. Find the lowest common ancestor (LCA) of two nodes in a binary tree.

29. Write a program to find the maximum depth of a binary tree.

30. Implement a function to convert a sorted array into a balanced binary search tree.

#### ### 6. Recursion and Backtracking

31. Write a program to generate all subsets of a given set.

32. Implement the N-Queens problem.

33. Solve the Tower of Hanoi problem.

34. Write a function to generate all permutations of a string.

35. Implement a Sudoku solver.

#### ### 7. Dynamic Programming

36. Implement a program to find the nth Fibonacci number using dynamic programming.

37. Write a function to find the longest increasing subsequence in an array.
38. Given a 2D grid, find the number of unique paths from the top-left corner to the bottom-right corner.
39. Implement a function to find the maximum sum subarray (Kadane's Algorithm).
40. Solve the coin change problem to find the minimum number of coins needed to make a given amount.

### ### 8. Bit Manipulation

41. Write a program to find the only non-repeating element in an array where every other element repeats twice.
42. Implement a function to count the number of 1 bits in an integer.
43. Swap two numbers without using a temporary variable.
44. Write a program to determine if a given integer is a power of 2.
45. Find the two non-repeating elements in an array where every other element repeats twice.

### ### 9. Object-Oriented Programming

46. Design a class to implement a simple cache (e.g., LRU Cache) using a suitable data structure.
47. Implement a class for a deck of cards. Include methods for shuffling, dealing a card, and resetting the deck.
48. Create a class to represent a bank account and implement methods for deposit, withdrawal, and balance inquiry.
49. Design a parking lot system with classes for parking slots, vehicles, and management logic.
50. Write a program to model a simple library management system with classes for books, members, and transactions.

### ### 10. Miscellaneous

51. Write a program to merge overlapping intervals.

52. Implement a function to convert a given integer into its Roman numeral representation.
53. Given a matrix, rotate it by 90 degrees.
54. Implement a function to find the maximum product of three numbers in an array.
55. Write a program to flatten a nested list of integers.