

Deloitte DSA Previous Year Questions

Easy

- Write the code for this: Find the missing number in an array of integers from 1 to N.
(Given an array containing numbers from 1..N with one missing, find which number is missing.)
- Write the code for this: Reverse a given string.
(Given a string, return the string read backwards.)
- Write the code for this: Check if a string is a palindrome.
(Determine whether a string reads the same forwards and backwards.)
- Write the code for this: Find the largest and smallest element in an unsorted array.
(Given an unsorted array, return its maximum and minimum values.)
- Write the code for this: Find the count of vowels and consonants in a string.
(Count how many vowels and how many consonants are present in a given string.)
- Write the code for this: Given two sorted arrays, merge them into a single sorted array.
(Combine two sorted arrays into one sorted array containing all elements.)
- Write the code for this: Implement a stack using an array.
(Build a stack data structure (push, pop, peek, isEmpty) backed by an array.)
- Write the code for this: Implement a queue using an array.
(Build a queue (enqueue, dequeue, peek, isEmpty) using an array.)
- Write the code for this: Find the second largest number in an array.
(Return the second-highest distinct value in an array.)
- Write the code for this: Remove duplicate elements from a sorted array.

(Given a sorted array, remove duplicates in-place so each element appears once.)

- Write the code for this: Find the first non-repeating character in a string.
(Return the first character in a string that does not repeat anywhere else.)
- Write the code for this: Calculate the factorial of a number using recursion.
(Implement $\text{factorial}(n)$ recursively to compute $n!$.)
- Write the code for this: Search for an element in a 2D matrix where rows and columns are sorted.
(Given a matrix sorted row-wise and column-wise, determine if a target exists.)
- Write the code for this: Reverse a linked list.
(Given the head of a singly linked list, reverse it and return the new head.)
- Write the code for this: Find the middle element of a linked list.
(Return the middle node (or element) of a singly linked list.)
- Write the code for this: Check if two strings are anagrams of each other.
(Decide whether the two strings contain the exact same characters in any order.)
- Write the code for this: Given an array of integers, move all zeros to the end.
(Reorder an array so that all zeros come after non-zero elements, preserving order.)
- Write the code for this: Find the maximum sum of a subarray of size K.
(Given an array and k , find the contiguous subarray of length k with the largest sum.)
- Write the code for this: Check if a binary tree is a valid Binary Search Tree (BST).
(Verify the tree satisfies BST ordering: $\text{left} < \text{node} < \text{right}$ for every node.)

- Write the code for this: Find the height of a binary tree.
(Return the number of levels in a binary tree — the length of the longest root→leaf path.)
-

Medium

- Write the code for this: Find the next greater element for each element in an array.
(For every element, find the first element to its right that is greater; otherwise -1.)
- Write the code for this: Rotate an array by K positions to the right.
(Shift the array right by k steps, wrapping elements around.)
- Write the code for this: Find the starting and ending positions of a target value in a sorted array.
(Given a sorted array and a target, return the first and last indices where it appears.)
- Write the code for this: Find the longest substring without repeating characters.
(Return the length of the longest substring that contains no duplicate characters.)
- Write the code for this: Detect a cycle in a linked list.
(Determine whether a linked list contains a cycle (loop) using fast/slow pointers.)
- Write the code for this: Find the intersection point of two linked lists.
(Given two singly linked lists, return the node where they merge, if any.)
- Write the code for this: Perform a level order traversal of a binary tree.
(Traverse the tree level by level and return node values per level.)
- Write the code for this: Find the lowest common ancestor (LCA) of two nodes in a BST.

(Given a BST and two nodes, return their lowest common ancestor node.)

- Write the code for this: Given a set of intervals, merge the overlapping intervals.
(Combine intervals that overlap into minimal number of non-overlapping intervals.)
- Write the code for this: Find all unique combinations in an array that sum up to a target (Combination Sum).
(Return all unique combinations of numbers that add up to target; reuse of elements allowed per problem variant.)
- Write the code for this: Find the number of islands in a 2D grid.
(Count groups of connected 1's (land) in a binary grid using DFS/BFS.)
- Write the code for this: Implement a Least Recently Used (LRU) Cache.
(Design a cache with $O(1)$ get and put that evicts the least-recently used key when full.)
- Write the code for this: Sort an array of 0s, 1s, and 2s (Dutch National Flag problem).
(Reorder array in-place so all 0s, then 1s, then 2s appear, in linear time.)
- Write the code for this: Find the Kth largest element in an array.
(Return the k^{th} largest value in an unsorted array.)
- Write the code for this: Find the maximum product subarray.
(Find the contiguous subarray that yields the largest product.)
- Write the code for this: Validate balanced parentheses in an expression.
(Check whether every opening bracket has a matching closing bracket in correct order.)
- Write the code for this: Generate all valid parentheses for a given number N.
(Return all well-formed parentheses combinations using N pairs.)
- Write the code for this: Perform a topological sort on a Directed Acyclic Graph (DAG).

(Return an ordering of nodes where for every directed edge $u \rightarrow v$, u comes before v .)

- Write the code for this: Find the length of the longest common subsequence between two strings.
(Return the length of the longest sequence appearing in order (not necessarily contiguous) in both strings.)
 - Write the code for this: Implement Dijkstra's shortest path algorithm.
(Given weighted non-negative graph and a source, compute shortest distances to all nodes.)
-

Hard

- Write the code for this: Find the median of two sorted arrays of different sizes.
(Given two sorted arrays, find the median of the combined array in logarithmic time.)
- Write the code for this: Find the trapping rain water problem.
(Given bar heights, compute how much water can be trapped between bars after raining.)
- Write the code for this: Find the longest palindromic substring.
(Return the longest substring that reads the same forwards and backwards.)
- Write the code for this: Implement a regular expression matcher.
(Implement pattern matching supporting '.' and '*' with correct semantics.)*
- Write the code for this: Solve the N-Queens problem.
(Place n queens on an $n \times n$ board so that no two threaten each other; return all solutions.)
- Write the code for this: Find the maximum sum path in a binary tree.
(Find the maximum sum of any path (not necessarily root-to-leaf) in a binary tree.)

- Write the code for this: Serialize and deserialize a binary tree.
(Design functions to convert a tree to a string and back, preserving structure.)
 - Write the code for this: Find the minimum window substring that contains all characters of another string.
(Return the smallest substring of S that contains all characters from T, including multiplicity.)
 - Write the code for this: Find the largest rectangle in a histogram.
(Given bar heights, compute the largest rectangular area that fits in the histogram.)
 - Write the code for this: Solve the word break problem using dynamic programming.
(Return whether a string can be segmented into dictionary words; optionally return all segmentations.)
 - Write the code for this: Merge K sorted linked lists.
(Given k sorted linked lists, merge them into one sorted linked list efficiently.)
 - Write the code for this: Find the shortest path in a binary maze.
(Given a grid with open/blocked cells, find the shortest path from source to target using valid moves.)
-