

TCS DSA Previous Year Questions

Easy

- Write the code for this: Find if a number is prime.
(Decide whether a given integer greater than 1 has any divisors other than 1 and itself.)
- Write the code for this: Swap two numbers without using a third variable.
(Swap the values of two variables using only arithmetic or bitwise operations, no extra storage.)
- Write the code for this: Find the sum of all elements in an array.
(Compute the total sum of every number in the given array.)
- Write the code for this: Count the occurrences of a character in a string.
(Given a string and a character, count how many times that character appears.)
- Write the code for this: Remove all white spaces from a string.
(Return the string with spaces, tabs, and other whitespace characters removed.)
- Write the code for this: Sort an array using bubble sort.
(Implement the bubble sort algorithm to sort an array in ascending order.)
- Write the code for this: Find the common elements between two arrays.
(Return the elements that appear in both arrays.)
- Write the code for this: Implement a linear search algorithm.
(Search for a value by scanning each element until found or the end is reached.)
- Write the code for this: Implement a binary search algorithm on a sorted array.
(Efficiently locate a target value in a sorted array using divide-and-conquer.)

- Write the code for this: Check if a linked list is a palindrome.
(Determine whether the sequence of node values reads the same forward and backward.)
 - Write the code for this: Insert a node at the beginning of a linked list.
(Given the head of a list and a value, add a new node as the new head.)
 - Write the code for this: Delete a node from a linked list given a key.
(Remove the first node whose value equals the given key and return the new head.)
 - Write the code for this: Find the sum of digits of a number.
(Compute the sum of all decimal digits of the given non-negative integer.)
 - Write the code for this: Print the Fibonacci series up to N terms.
(Generate the first N numbers of the Fibonacci sequence.)
 - Write the code for this: Check for an Armstrong number.
(Determine whether a number equals the sum of its digits each raised to the number of digits.)
 - Write the code for this: Find the frequency of each element in an array.
(Count how many times each distinct value appears in the array.)
 - Write the code for this: Reverse the words in a given sentence.
(Given a sentence, return a string with the word order reversed.)
 - Write the code for this: Find the diameter of a binary tree.
(Return the length (number of nodes or edges, specify) of the longest path between any two nodes.)
 - Write the code for this: Perform in-order, pre-order, and post-order traversals of a binary tree.
(Implement recursive or iterative traversals that visit nodes in the three standard orders.)
 - Write the code for this: Check if two trees are identical.
(Decide whether two binary trees have the same structure and node values.)
-

Medium

- Write the code for this: Find the leader elements in an array (elements greater than all elements to their right).
(Return elements that are strictly greater than every element to their right.)
- Write the code for this: Rearrange an array in alternating positive and negative items.
(Reorder elements so they alternate sign: positive, negative, positive, ..., preserving relative order if required.)
- Write the code for this: Find the equilibrium index of an array.
(Return an index where the sum of elements to its left equals the sum to its right.)
- Write the code for this: Find the majority element in an array (appears more than $N/2$ times).
(Return the element that occurs more than half the time, if any.)
- Write the code for this: Find the stock buy and sell problem to maximize profit (one transaction).
(Given daily prices, find the max profit achievable by one buy and one sell.)
- Write the code for this: Find the starting point of the loop in a cyclic linked list.
(Given a linked list with a cycle, return the node where the cycle begins.)
- Write the code for this: Rotate a linked list by K nodes.
(Move the last k nodes to the front (or rotate right by k) and return the new head.)
- Write the code for this: Flatten a multilevel linked list.
(Convert a list where nodes may have child lists into a single-level linked list in order.)

- Write the code for this: Print the left view of a binary tree.
(Return the nodes visible when the tree is viewed from the left side—one per level.)
- Write the code for this: Convert a binary tree to its mirror tree.
(Transform the tree so left and right children are swapped at every node.)
- Write the code for this: Find the minimum element in a rotated sorted array.
(Given a sorted array rotated at some pivot, find its minimum value.)
- Write the code for this: Implement a queue using two stacks.
(Use two stacks to simulate enqueue and dequeue operations of a FIFO queue.)
- Write the code for this: Find all permutations of a string.
(Generate every ordering (permutation) of the string's characters.)
- Write the code for this: Find if there is a path between two vertices in a graph (using BFS or DFS).
(Return true if you can reach vertex B from vertex A following edges.)
- Write the code for this: Find the minimum number of coins required to make a given value (Coin Change problem).
(Given coin denominations and an amount, return the minimum coin count to make that amount.)
- Write the code for this: Find the longest common prefix among an array of strings.
(Return the longest starting substring common to all strings.)
- Write the code for this: Group anagrams together from a list of words.
(Partition words into groups where each group's words are anagrams of each other.)
- Write the code for this: Implement an algorithm to print all valid combinations of n-pairs of parentheses.
(Generate all well-formed parentheses strings that use n opening and n closing brackets.)
- Write the code for this: Find the number of ways to reach the top of a staircase (climbing stairs problem).
(Count distinct sequences of 1- or 2-step moves to reach stair N.)

- Write the code for this: Check if a graph is bipartite.
(Determine whether vertices can be colored using two colors so no edge has same-color ends.)
-

Hard

- Write the code for this: Find the largest subarray with a sum of 0.
(Return the longest contiguous subarray whose elements sum to zero.)
- Write the code for this: Solve the 0/1 Knapsack problem using dynamic programming.
(Given item weights and values and capacity W , maximize value by choosing items without splitting.)
- Write the code for this: Find the maximum path sum from any node to any node in a binary tree.
(Return the largest sum obtainable by any path in the tree (path may start and end anywhere).)
- Write the code for this: Find the length of the longest increasing subsequence.
(Return the size of the largest strictly increasing subsequence in the array.)
- Write the code for this: Implement a trie (prefix tree) and its insert and search operations.
(Build a prefix tree supporting insertion of words, full-word search, and prefix search.)
- Write the code for this: Find all possible palindromic partitions of a string.
(Partition the string into substrings so every substring is a palindrome; return all such partitions.)
- Write the code for this: Find the minimum number of jumps to reach the end of an array.

(Given an array where each element is max jump length, return the minimum jumps to reach last index.)

- Write the code for this: Connect nodes at the same level in a binary tree.
(Set each node's next pointer to its immediate neighbor to the right on the same level.)
 - Write the code for this: Find the longest consecutive sequence in an unsorted array.
(Return the length of the longest run of consecutive integers, order-agnostic.)
 - Write the code for this: Solve Sudoku using backtracking.
(Fill the 9×9 board so every row, column, and 3×3 box contains digits 1–9.)
 - Write the code for this: Find the celebrity in a party.
(Given a matrix of “knows” relations, find the person everyone knows but who knows no one.)
 - Write the code for this: Design a data structure that supports insert, delete, and getRandom in $O(1)$ time.
(Implement a structure that can add and remove values and return a random element, all in average constant time.)
-