grow_pins

Coding Interview Preparation Guide

Basic Level Questions

Arrays and Strings

1. Find the largest and smallest elements in an array.
    - **Hint:** Use a single loop with conditional checks.
    - **Time Complexity:** O(n)
    - **Asked By:** TCS, Infosys

2. Reverse a string.
    - **Hint:** Use two-pointer technique.
    - **Time Complexity:** O(n)
    - **Asked By:** Infosys, Wipro

3. Check if a string is a palindrome.
    - **Hint:** Compare the string with its reverse.
    - **Time Complexity:** O(n)
    - **Asked By:** Accenture, Cognizant

4. Merge two sorted arrays.
    - **Hint:** Use two-pointer technique for linear merge.
    - **Time Complexity:** O(n + m)
    - **Asked By:** TCS, Capgemini

5. Count the frequency of characters in a string.
    - **Hint:** Use a hash map to store frequencies.
    - **Time Complexity:** O(n)
    - **Asked By:** Infosys, HCL

6. Find the second largest element in an array.
    - **Hint:** Traverse the array twice or use two variables to track the largest and second largest.
    - **Time Complexity:** O(n)
    - **Asked By:** Capgemini, Cognizant

grow_pins

7. Check if two strings are rotations of each other.
    - **Hint:** Concatenate one string with itself and check for the substring.
    - **Time Complexity:** O(n)
    - **Asked By:** Infosys, TCS

8. Find the missing number in an array of size n containing numbers from 1 to n+1.
    - **Hint:** Use the sum formula or XOR approach.
    - **Time Complexity:** O(n)
    - **Asked By:** Wipro, Accenture

9. Remove duplicates from a sorted array.
    - **Hint:** Use two-pointer technique.
    - **Time Complexity:** O(n)
    - **Asked By:** Infosys, Cognizant

10. Find the intersection of two arrays.
    - **Hint:** Use hash maps or sorting with two-pointer technique.
    - **Time Complexity:** O(n + m)
    - **Asked By:** TCS, Capgemini

11. Find the maximum difference between two elements in an array such that larger element appears after the smaller element.
    - **Hint:** Track the minimum value while traversing.
    - **Time Complexity:** O(n)
    - **Asked By:** Cognizant, Infosys

12. Check if an array is sorted.
    - **Hint:** Compare adjacent elements in a single loop.
    - **Time Complexity:** O(n)
    - **Asked By:** TCS, Wipro

grow_pins

Intermediate Level Questions

Data Structures

1. Implement a stack using an array.
    - **Hint:** Use a top pointer to track the current stack position.
    - **Time Complexity:** Push and Pop O(1)
    - **Asked By:** TCS, Wipro

2. Reverse a linked list.
    - **Hint:** Use three pointers (previous, current, next).
    - **Time Complexity:** O(n)
    - **Asked By:** Infosys, Cognizant

3. Check if a binary tree is a Binary Search Tree (BST).
    - **Hint:** Use an in-order traversal to check for sorted order.
    - **Time Complexity:** O(n)
    - **Asked By:** Accenture, HCL

4. Find the middle element of a linked list in one pass.
    - **Hint:** Use a slow and fast pointer technique.
    - **Time Complexity:** O(n)
    - **Asked By:** TCS, Infosys

5. Implement a queue using stacks.
    - **Hint:** Use two stacks, one for enqueue and another for dequeue operations.
    - **Time Complexity:** O(1) amortized per operation.
    - **Asked By:** Capgemini, Infosys

6. Flatten a binary tree into a linked list.
    - **Hint:** Use post-order traversal to rearrange nodes.
    - **Time Complexity:** O(n)
    - **Asked By:** Accenture, Cognizant

7. Find the depth of a binary tree.
    - **Hint:** Use a recursive function for depth-first search (DFS).
    - **Time Complexity:** O(n)
    - **Asked By:** TCS, Wipro

grow_pins

Arrays

1. Find the subarray with the maximum sum (Kadane's Algorithm).
    - **Hint:** Track the current sum and the maximum sum.
    - **Time Complexity:** O(n)
    - **Asked By:** Cognizant, Accenture

2. Rotate an array by k positions.
    - **Hint:** Reverse parts of the array.
    - **Time Complexity:** O(n)
    - **Asked By:** Capgemini, Infosys

3. Find the first non-repeating element in an array.
    - **Hint:** Use a hash map to store counts and indices.
    - **Time Complexity:** O(n)
    - **Asked By:** TCS, HCL

4. Find the pair of elements with the given sum in a sorted array.
    - **Hint:** Use the two-pointer technique.
    - **Time Complexity:** O(n)
    - **Asked By:** Infosys, Wipro

5. Find the product of the array elements except for the current index.
    - **Hint:** Use prefix and suffix product arrays.
    - **Time Complexity:** O(n)
    - **Asked By:** TCS, Cognizant

Strings

1. Find all permutations of a string.
    - **Hint:** Use recursion with backtracking.
    - **Time Complexity:** O(n!)
    - **Asked By:** Infosys, Wipro

2. Check if two strings are anagrams.
    - **Hint:** Sort both strings or use a frequency array.
    - **Time Complexity:** O(n log n) or O(n)
    - **Asked By:** Cognizant, Capgemini

grow_pins

3. Find the longest substring without repeating characters.
    - **Hint:** Use a sliding window technique with a hash set.
    - **Time Complexity:** O(n)
    - **Asked By:** Infosys, Accenture

4. Check if a string contains balanced parentheses.
    - **Hint:** Use a stack to validate the pairs.
    - **Time Complexity:** O(n)
    - **Asked By:** TCS, Capgemini

5. Find the longest palindromic substring.
    - **Hint:** Use dynamic programming or expand around center.
    - **Time Complexity:** O(n^2)
    - **Asked By:** TCS, Cognizant

grow_pins

Advanced Level Questions

Dynamic Programming

1. Find the longest increasing subsequence in an array.
    - **Hint:** Use DP with a 1D array.
    - **Time Complexity:** O(n^2) or O(n log n)
    - **Asked By:** TCS, Accenture

2. Solve the word break problem.
    - **Hint:** Use a DP array to track valid words.
    - **Time Complexity:** O(n^2)
    - **Asked By:** Infosys, Cognizant

3. Find the minimum number of coins for change.
    - **Hint:** Use bottom-up DP to minimize the count.
    - **Time Complexity:** O(n * m)
    - **Asked By:** TCS, HCL

4. Solve the 0/1 Knapsack problem.
    - **Hint:** Use DP with a 2D table.
    - **Time Complexity:** O(n * W)
    - **Asked By:** Infosys, Capgemini

5. Find the number of ways to decode a string of digits.
    - **Hint:** Use DP to count possible interpretations.
    - **Time Complexity:** O(n)
    - **Asked By:** Infosys, Cognizant

Graph Algorithms

1. Find the shortest path in a weighted graph (Dijkstra's Algorithm).
    - **Hint:** Use a priority queue to track minimum distances.
    - **Time Complexity:** O((V + E) log V)
    - **Asked By:** TCS, Accenture
    -
2. Implement Kruskal's Algorithm for Minimum Spanning Tree.
    - **Hint:** Use a union-find data structure.
    - **Time Complexity:** O(E log E)

   - **Asked By:** Infosys, Cognizant
3. Detect a cycle in a graph (directed and undirected).
   - **Hint:** Use DFS for directed graphs and union-find for undirected graphs.
   - **Time Complexity:** O(V + E)
   - **Asked By:** TCS, Accenture

4. Find the strongly connected components in a graph (Kosaraju's Algorithm).
   - **Hint:** Use two-pass DFS with graph reversal.
   - **Time Complexity:** O(V + E)
   - **Asked By:** Accenture, Capgemini

Advanced Data Structures

1. Implement a trie for dictionary operations.
   - **Hint:** Use a tree structure with nodes for each character.
   - **Time Complexity:** O(length of word)
   - **Asked By:** TCS, Capgemini

2. Design and implement an LRU cache.
   - **Hint:** Use a combination of a hash map and a doubly linked list.
   - **Time Complexity:** O(1) for get and put operations.
   - **Asked By:** Accenture, Cognizant

3. Implement a min-heap and its operations.
   - **Hint:** Use an array representation for the heap.
   - **Time Complexity:** O(log n)
   - **Asked By:** Infosys, Wipro

4. Implement a segment tree for range queries.
   - **Hint:** Build the tree recursively and use lazy propagation for updates.
   - **Time Complexity:** O(n log n)
   - **Asked By:** TCS, HCL

5. Design a data structure to find median in a stream of numbers.
   - **Hint:** Use a combination of max-heap and min-heap.
   - **Time Complexity:** O(log n) for insert and O(1) for median.
   - **Asked By:** Cognizant, Infosys

Bonus Problems

1. Solve the N-Queens problem.
    - **Hint:** Use backtracking to place queens.
    - **Time Complexity:** O(n!)
    - **Asked By:** Infosys, Cognizant

2. Implement the Rabin-Karp algorithm for string matching.
    - **Hint:** Use hashing to compare substring matches.
    - **Time Complexity:** O(n + m) average.
    - **Asked By:** TCS, HCL

3. Solve the Travelling Salesman Problem using dynamic programming.
    - **Hint:** Use bitmasking for state representation.
    - **Time Complexity:** O(n^2 * 2^n)
    - **Asked By:** Infosys, Capgemini

4. Design a snake and ladder game using OOP principles.
    - **Hint:** Use classes for board, players, and dice.
    - **Asked By:** Accenture, Cognizant

Infosys Online Test Sample : Click Here

Curious freaks Coding Sheet : Click Here
Please go through Curious freaks Coding Sheet
Was very useful to me !

Happy Learning !