DSA COMPLETE GUIDE (ZOHO PREPARATION)

---

1. WHY DSA IS IMPORTANT FOR ZOHO

Zoho interviews focus on **problem-solving skills** and **clarity of logic**. DSA is the backbone of coding interviews because it teaches: - Efficient problem solving - Structured thinking - Time and space optimization - Ability to scale solutions

In Zoho, they look for **clean logic**, not just complex code.

---

2. COMPLETE DSA TOPIC LIST (WHAT TO COVER)

2.1 Basics (Must have fingertip clarity)

- Arrays
- Strings
- Linked Lists
- Stacks
- Queues
- Hashing (HashMap/HashSet)

2.2 Intermediate (High priority)

- Trees (Binary Tree, BST)
- Recursion
- Sorting algorithms
- Searching algorithms
- Backtracking (basic)

2.3 Advanced (Good to know for higher rounds)

- Graphs
- Dynamic Programming
- Trie
- Segment Tree
- Heap / Priority Queue

---

3. DSA TOPICS WITH DETAILED NOTES

3.1 ARRAYS (COMPLETE NOTES)

What is Array?

An array is a collection of elements stored in contiguous memory locations.

Key Operations

- Access by index
- Insert (costly)
- Delete (costly)

Important Problems

- Find max/min
- Two-sum
- Subarray sum
- Sliding window
- Rotate array
- Move zeroes

Concepts to be strong in

- Prefix sum
- Two pointers
- Sliding window
- Sorting-based approach

Example: Two Sum

Problem: Find two numbers whose sum equals target.

Approach: Use HashMap. - Iterate array - Check if (target - current) exists - If yes, return indices

Complexity: O(n)

3.2 STRINGS (COMPLETE NOTES)

What is String?

A string is a sequence of characters.

Key Operations

- Concatenation
- Substring
- Reverse
- Compare

Important Problems

- Palindrome
- Anagram
- First non-repeating character
- Pattern matching
- Longest substring without repeating characters

Concepts to be strong in

- Sliding window
- Hashing
- Two pointers

3.3 LINKED LISTS (COMPLETE NOTES)

What is Linked List?

A linear data structure where elements are stored in nodes and linked using pointers.

Types

- Singly Linked List
- Doubly Linked List
- Circular Linked List

Important Problems

- Reverse linked list
- Detect cycle
- Merge two sorted lists
- Find middle node
- Remove nth node from end

Concepts to be strong in

- Slow-fast pointers
- Dummy node
- Pointer manipulation

3.4 STACKS & QUEUES (COMPLETE NOTES)

Stack

LIFO (Last In First Out)

Uses: - Parenthesis matching - Reverse a string - DFS (graph)

Queue

FIFO (First In First Out)

Uses: - BFS (graph) - Level order traversal

Important Problems

- Balanced parentheses
- Next greater element
- Queue using stack

3.5 HASHING (COMPLETE NOTES)

What is Hashing?

A technique to store key-value pairs for O(1) average time operations.

Data structures

- HashMap
- HashSet

Important Problems

- Count frequency
- Two sum
- Subarray sum
- Anagram grouping

Concepts to be strong in

- Collision handling
- Hash functions

3.6 TREES (COMPLETE NOTES)

What is Tree?

A hierarchical data structure with nodes connected by edges.

Types

- Binary Tree
- Binary Search Tree (BST)

Traversals

- Inorder
- Preorder
- Postorder
- Level order

Important Problems

- Lowest Common Ancestor
- Height of tree
- Balanced tree
- BST validation

Concepts to be strong in

- Recursion
- Tree traversal
- BFS/DFS

3.7 RECURSION (COMPLETE NOTES)

What is Recursion?

A function calling itself.

Key Concepts

- Base case
- Recursive case
- Stack memory

Important Problems

- Factorial
- Fibonacci

- Tree traversal
- Backtracking

3.8 SORTING & SEARCHING (COMPLETE NOTES)

Sorting Algorithms

- Bubble Sort (O(n^2))
- Selection Sort (O(n^2))
- Insertion Sort (O(n^2))
- Merge Sort (O(n log n))
- Quick Sort (O(n log n))

Searching Algorithms

- Linear Search
- Binary Search (requires sorted array)

Important Concepts

- Time complexity
- Stable vs unstable sorting

3.9 BACKTRACKING (BASIC)

What is Backtracking?

A method of solving problems by trying possibilities and removing wrong paths.

Important Problems

- N-Queens
- Sudoku
- Subset generation

3.10 GRAPHS (BASIC)

What is Graph?

A set of nodes connected by edges.

Types

- Directed vs Undirected
- Weighted vs Unweighted

Traversals

- BFS (queue)
- DFS (stack/recursion)

Important Problems

- Shortest path (BFS for unweighted)
- Cycle detection
- Topological sort

3.11 DYNAMIC PROGRAMMING (BASIC)

What is DP?

A technique to solve problems by storing previous results.

Key Concepts

- Memoization
- Tabulation

Important Problems

- Fibonacci with DP
- Knapsack
- Longest Increasing Subsequence

3.12 HEAP / PRIORITY QUEUE (BASIC)

What is Heap?

A complete binary tree with special order property.

Types

- Max heap
- Min heap

Uses

- Top K elements
- Median maintenance
- Priority scheduling

---

4. WHAT ZOHO EXPECTS (DSA LEVEL)

Must be fingertip ready

- Arrays + Strings
- Linked list basics
- Stacks & queues
- Hashing
- Basic recursion

Must be strong in

- Trees (BST)
- Sorting & searching
- Two pointers
- Sliding window

Good to know

- Graphs (basic)
- DP (basic)

---

5. BEST PRACTICE STRATEGY (FOR ZOHO)

Step 1: Basics

- Learn each data structure
- Understand time complexity

Step 2: Solve 30 problems per topic

- Start with easy
- Move to medium

Step 3: Mock tests

- 2 coding tests per week

Step 4: Explain your logic

- Write clean code
- Handle edge cases

---

6. REFERENCE LINKS (FOR DEEPER STUDY)

(Use these only for reference and deeper understanding)

1. GeeksforGeeks – DSA topics (complete)
2. LeetCode – Practice problems
3. CodeChef – DSA practice
4. HackerRank – DSA fundamentals

END OF DSA GUIDE