

LEVEL 1: Basics & Fundamentals

Goal: Understand foundational concepts, complexity, and simple data handling.

Topics (60 MCQs):

1. Introduction to DSA
 2. Time and Space Complexity
 - Big O, Big Θ , Big Ω notation
 3. Recursion basics
 4. Arrays
 - Declaration, traversal, insertion, deletion
 - Searching (linear search)
 5. Strings
 - Basic operations
 6. Mathematical algorithms
 - Factorials, GCD, LCM, primes
 7. Input/Output handling
-

LEVEL 2: Basic Data Structures

Goal: Learn fundamental data structures and their operations.

Topics (60 MCQs):

1. Linked List
 - Singly linked list, doubly linked list, circular linked list
 - Traversal, insertion, deletion
 2. Stack
 - Array & linked list implementation
 - Push, pop, peek, applications
 3. Queue
 - Simple queue, circular queue, priority queue
 - Enqueue, dequeue, front, rear
 4. Basic problems
 - Reverse linked list, balanced parentheses, stack/queue challenges
-

LEVEL 3: Searching & Sorting

Goal: Understand searching/sorting algorithms and their efficiency.

Topics (60 MCQs):

1. Searching Algorithms
 - Linear search, binary search
 - Ternary search
 - Search in sorted & rotated arrays
2. Sorting Algorithms
 - Bubble sort, selection sort, insertion sort

- Merge sort, quick sort, heap sort
 - Counting sort, radix sort
 - 3. Complexity analysis of algorithms
 - 4. Practice problems
 - Find min/max, kth smallest/largest, sorted order check
-

LEVEL 4: Advanced Data Structures

Goal: Handle complex data efficiently using advanced data structures.

Topics (60 MCQs):

1. Trees
 - Binary tree, binary search tree (BST)
 - Tree traversals (inorder, preorder, postorder)
 - Height, depth, level order
 2. Heap
 - Min-heap, max-heap
 - Heapify, insert, delete
 3. Hashing
 - Hash tables, hash functions, collision resolution
 4. Graph Basics
 - Representation (adjacency matrix/list), BFS, DFS
 5. Practical problems
 - Lowest common ancestor, path finding
-

LEVEL 5: Graphs & Advanced Algorithms

Goal: Solve complex problems using graphs and algorithmic techniques.

Topics (60 MCQs):

1. Graph Algorithms
 - BFS, DFS, Dijkstra's, Bellman-Ford
 - Floyd-Warshall, Prim's, Kruskal's
 - Topological sorting
 2. Dynamic Programming (DP)
 - Fibonacci, Knapsack, Coin Change, Longest Common Subsequence (LCS)
 3. Greedy Algorithms
 - Interval scheduling, Huffman coding
 4. Backtracking
 - N-Queens, Rat in Maze, Sudoku solver
 5. Complexity and optimization
-

LEVEL 6: Expert Level & Advanced Problem Solving

Goal: Master competitive programming and advanced problem-solving.

Topics (60 MCQs):

1. Advanced Trees
 - AVL tree, Red-Black tree, Segment tree, Fenwick tree (BIT)
2. Advanced Graphs
 - Shortest path optimizations, Network flow (Ford-Fulkerson)
 - Strongly connected components, Tarjan's algorithm
3. Advanced DP
 - Bitmask DP, DP on trees
4. String Algorithms
 - KMP, Rabin-Karp, Trie, Suffix Tree/Array
5. Complexity Analysis
 - Amortized analysis, NP problems, Approximation algorithms
6. Problem-solving patterns
 - Sliding window, two pointers, divide & conquer, greedy + DP combinations