

DSA Topics for Different Salary Brackets

1. 4-6 LPA (Entry-level roles, fresher positions)

Focus: Basic understanding of DSA, commonly used algorithms, and problem-solving skills.

Topics:

1. Arrays (Easy/Medium)

- Basics: Traversal, insertion, deletion.
- Problems: Kadane's Algorithm, Two Sum.

2. Strings (Easy/Medium)

- Palindrome, Anagram, Substrings.

3. Hashing (Easy/Medium)

- HashMaps and Sets: Frequency count, Pair sum problems.

4. Sorting and Searching (Easy/Medium)

- Binary Search, Quick Sort, Merge Sort.

5. Linked Lists (Easy/Medium)

- Basics: Reversal, middle element, merging sorted lists.

6. Stack and Queue (Easy/Medium)

- Problems: Balanced Parentheses, Next Greater Element.

7. Basic Recursion (Easy/Medium)

- Factorial, Fibonacci, Subset problems.

8. Basic Trees (Easy)

- Binary Trees: Traversals (Inorder, Preorder, Postorder).

Preparation Tip: Solve easy-medium problems on platforms like LeetCode, GeeksforGeeks, or Codeforces.

2. 10-25 LPA (Mid-level roles, some experience required)

Focus: Strong grasp of fundamental DSA, medium-hard problems, and optimized solutions.

Topics:

1. Advanced Arrays (Medium)

- Sliding Window, Prefix Sums, Two Pointers.
- Problems: Maximum Subarray Sum, Minimum Window Substring.

2. Strings (Medium)

- Pattern Matching: KMP Algorithm.
- Problems: Longest Palindromic Substring.

3. Dynamic Programming (DP) (Medium)

- Problems: Longest Increasing Subsequence, Knapsack, DP on Strings.

4. Binary Search (Medium)

- Problems: Median of Two Sorted Arrays, Search in Rotated Array.

5. Trees and Binary Search Trees (BST) (Medium)

- Lowest Common Ancestor, Binary Tree Views, BST validation.

6. Graphs (Basics) (Medium)

- BFS, DFS, Shortest Path (Dijkstra's Algorithm).

7. Trie (Medium)

- Word Search, Prefix problems.

8. Bit Manipulation (Medium)

- XOR Problems, Subsets using bits.

Preparation Tip: Focus on solving a mix of medium and hard problems regularly.

3. 25+ LPA (High-paying roles, product-based companies like FAANG)

Focus: Expert-level understanding of DSA, system optimization, and problem-solving.

Topics:

1. Advanced Arrays and Strings (Medium/Hard)

- Problems: Subarrays with Target Sum, Advanced Sliding Window.

2. Graphs (Medium/Hard)

- Topological Sort, Minimum Spanning Tree (Kruskal, Prim), Floyd-Warshall.

3. Dynamic Programming (Advanced) (Medium/Hard)

- DP on Trees, DP with Bitmasking.

4. Segment Trees and Fenwick Trees (Hard)

- Range Queries, Lazy Propagation.

5. Advanced Trees (Hard)

- AVL Trees, Red-Black Trees.

6. Advanced Graph Algorithms (Hard)

- Problems: Bridges in a Graph, Articulation Points.

7. Heaps and Priority Queues (Medium)

- Kth Largest Element, Median Maintenance.

8. Mathematics and Number Theory (Hard)

- Modular Arithmetic, Prime Factorization, GCD/LCM.

9. String Algorithms (Hard)

- Z-Algorithm, Suffix Arrays, Rabin-Karp.

Preparation Tip: Regularly practice contests on Codeforces, LeetCode hard problems, and system design challenges.

Key Suggestions for All Categories:

1. Prioritize problem-solving skills over just memorizing algorithms.

2. Use resources like LeetCode, HackerRank, and Codeforces for consistent practice.

3. Start with easy problems, and gradually increase the difficulty.

4. Learn the time complexity of every algorithm you implement.

By tailoring your DSA preparation to your target job role and salary bracket, you can efficiently focus on what's most important.