

COMPLETE ROADMAP (DSA 60 DAYS)

Day 1-7: Introduction to Programming and Basic Concepts

- Learn about programming basics in Java (if needed), such as variables, data types, loops, and conditional statements.
- Familiarize yourself with the basics of algorithm analysis, time complexity, and space complexity.

Day 8-10: Arrays and Strings

- Understand array manipulation, sorting, and searching algorithms (binary search).
- Learn about string manipulation, palindrome detection, and substring searching.

Day 11-15: Recursion and Backtracking

- Study recursion and solve problems using recursive techniques.
- Learn backtracking and practice solving problems like N-Queens and Sudoku.

Day 16-20: Linked Lists and Stacks

- Study linked list data structure and solve related problems.
- Understand stacks and solve problems involving parentheses balancing and infix-postfix conversion.

Day 21-25: Queues and Dequeues

- Study queue and dequeue data structures.
- Solve problems involving breadth-first search and implement a circular queue.

Day 26-30: Trees and Binary Trees

- Learn about tree data structures (binary trees, binary search trees).

- Understand tree traversal algorithms (preorder, inorder, postorder).

Day 31-35: Graphs

- Study graph representations (adjacency matrix, adjacency list).
- Learn about graph traversal algorithms (DFS, BFS).
- Practice solving graph problems like finding connected components and shortest paths.

Day 36-40: Sorting and Searching Algorithms

- Study various sorting algorithms (bubble, selection, insertion, merge, quick sort).
- Learn about searching algorithms (linear search, binary search).
- Practice implementing and understanding their complexities.

Day 41-45: Hashing and Hash Maps

- Learn about hashing and hash map data structures.
- Solve problems involving hash maps, like finding duplicate elements.

Day 46-50: Advanced Data Structures

- Study heap data structure and solve problems like finding kth largest/smallest element.
- Understand trie data structure and solve problems like prefix searching.

Day 51-55: Dynamic Programming

- Study dynamic programming concepts.
- Solve classic DP problems, such as coin change, knapsack, and longest common subsequence.

Day 56-60: Practice