Problem Solving & DSA Training Program

# Days 4-14 Agenda (Remaining 11 Days)

## Day 4: 17/07/2025 - Advanced Recursion & Introduction to Dynamic Programming

### Morning Session:

 Recursion optimization techniques (memoization)  Solve Coin Change problem using memoization

 Fibonacci with and without memoization  Tower of Hanoi problem

### Afternoon Session:

 Introduction to Dynamic Programming concepts  DP vs Recursion vs Memoization

 Climbing Stairs problem  House Robber problem

 Practice problems on HackerRank/LeetCode

## Day 5: 18/07/2025 - Dynamic Programming Fundamentals

### Morning Session:

 1D DP problems:

 Longest Increasing Subsequence

 Maximum Subarray Sum (Kadane's revisited with DP)  Jump Game problem

 Decode Ways

### Afternoon Session:

 2D DP introduction:

 Grid path problems  Minimum Path Sum  Unique Paths

 Edit Distance (introduction)

## Day 6: 19/07/2025 - Advanced Dynamic Programming

### Morning Session:

Classic DP problems:

 Longest Common Subsequence  0/1 Knapsack problem

 Edit Distance (detailed implementation)  Palindrome problems

### Afternoon Session:

 DP optimization techniques  Space optimization in DP

 Practice session with medium-level DP problems  Mini contest on DP problems

## Day 7: 20/07/2025 - Stacks and Queues

### Morning Session:

 Stack implementation and applications:  Valid Parentheses

 Next Greater Element

 Largest Rectangle in Histogram  Evaluate Reverse Polish Notation

### Afternoon Session:

 Queue implementation and applications:  Circular Queue implementation

 Stack using Queue and Queue using Stack  Sliding Window Maximum

 First non-repeating character in stream

## Day 8: 21/07/2025 - Linked Lists

### Morning Session:

 Linked List fundamentals:

 Singly Linked List implementation  Reverse Linked List

 Merge Two Sorted Lists

 Remove Nth Node from End

### Afternoon Session:

Advanced Linked List problems:

Detect Cycle in Linked List (Floyd's algorithm) Find Intersection of Two Linked Lists

Add Two Numbers (represented as linked lists) Copy List with Random Pointer

## Day 9: 22/07/2025 - Trees - Fundamentals

### Morning Session:

 Binary Tree basics:

 Tree traversals (Inorder, Preorder, Postorder)  Level Order Traversal

 Maximum Depth of Binary Tree  Symmetric Tree

### Afternoon Session:

 Binary Search Tree:

 BST implementation

 Search, Insert, Delete in BST  Validate BST

 Lowest Common Ancestor in BST  Convert Sorted Array to BST

## Day 10: 23/07/2025 - Advanced Trees

### Morning Session:

 Tree algorithms:

 Diameter of Binary Tree  Path Sum problems

 Binary Tree Maximum Path Sum

 Serialize and Deserialize Binary Tree

### Afternoon Session:

 Heap/Priority Queue:

 Min Heap and Max Heap implementation  Kth Largest Element

 Merge k Sorted Lists

 Top K Frequent Elements

## Day 11: 24/07/2025 - Graphs - Introduction

### Morning Session:

 Graph representations:

 Adjacency Matrix vs Adjacency List  Graph implementation in Python

 Depth First Search (DFS)  Number of Islands

### Afternoon Session:

 Breadth First Search (BFS):  BFS implementation

 Shortest Path in Unweighted Graph  Rotting Oranges

 Word Ladder

## Day 12: 25/07/2025 - Advanced Graph Algorithms

### Morning Session:

 Graph algorithms:

 Topological Sort

 Detect Cycle in Directed Graph  Course Schedule problem

 Clone Graph

### Afternoon Session:

 Shortest Path algorithms:  Dijkstra's Algorithm

 Network Delay Time

 Cheapest Flights Within K Stops  Introduction to Union-Find

## Day 13: 26/07/2025 - Specialized Topics & Optimization

### Morning Session:

 Bit Manipulation:

 Basic bitwise operations

Single Number problem

 Counting Bits  Power of Two

### Afternoon Session:

 Two Pointers & Sliding Window (advanced):  3Sum problem

 Container With Most Water

 Longest Substring Without Repeating Characters  Minimum Window Substring

## Day 14: 27/07/2025 - Integration & Final Assessment

### Morning Session:

 Trie (Prefix Tree):

 Trie implementation  Word Search II

 Implement Trie (Prefix Tree)

### Afternoon Session:

 **Final Contest/Assessment:**

 Mixed problems from all topics covered  Time-bound problem solving

 Code review and optimization discussion  Course wrap-up and next steps

# Daily Structure (Each Day)

## Morning Session (3 hours)

 **Theory & Concept Introduction** (45 minutes)  **Live Coding & Implementation** (90 minutes)  **Problem Solving Practice** (45 minutes)

## Afternoon Session (3 hours)

 **Advanced Problems** (90 minutes)

 **Hands-on Practice** (60 minutes)

 **Contest/Assessment** (30 minutes)

**Assessment Strategy**

## Daily Assessments:

 **Mini Contests** (Days 4, 6, 8, 10, 12) - 20 minutes each

 **Coding Challenges** - At least 2 problems per day

 **Peer Code Reviews** - 15 minutes daily

## Major Assessments:

 **Mid-term Contest** (Day 9) - 1 hour

 **Final Contest** (Day 14) - 2 hours

# Problem Sets by Day

## Day 4 Problems:

 Coin Change (with memoization)  Fibonacci (optimized)

 Climbing Stairs  House Robber

## Day 5 Problems:

 Longest Increasing Subsequence  Maximum Product Subarray

 Jump Game  Unique Paths

## Day 6 Problems:

 Longest Common Subsequence  0/1 Knapsack

 Edit Distance

 Palindromic Substrings

## Day 7 Problems:

 Valid Parentheses

 Next Greater Element

 Largest Rectangle in Histogram  Sliding Window Maximum

## Day 8 Problems:

 Reverse Linked List

 Merge Two Sorted Lists  Linked List Cycle

 Remove Nth Node from End

## Day 9 Problems:

 Binary Tree Inorder Traversal

 Maximum Depth of Binary Tree  Symmetric Tree

 Validate BST

## Day 10 Problems:

 Diameter of Binary Tree  Path Sum II

 Kth Largest Element  Merge k Sorted Lists

## Day 11 Problems:

 Number of Islands  Clone Graph

 Word Ladder

 Rotting Oranges

## Day 12 Problems:

 Course Schedule

 Network Delay Time

 Cheapest Flights Within K Stops  Find if Path Exists in Graph

## Day 13 Problems:

 Single Number  3Sum

 Container With Most Water

 Longest Substring Without Repeating Characters

## Day 14 Problems:

 Mixed problems from all topics  Time-bound contest problems

# Resources & Tools

## Online Platforms:

 LeetCode for daily practice  HackerRank for contests

 GeeksforGeeks for theory

## Python Libraries:

(deque, defaultdict, Counter) for priority queues

collections

heapq

bisect

for binary search

## Development Environment:

 Python 3.8+

 VS Code or PyCharm

 Jupyter Notebooks for visualization

# Expected Outcomes

By Day 14, participants will have:

 Solved 100+ coding problems

 Mastered major data structures and algorithms  Developed problem-solving intuition

 Prepared for technical interviews

 Built a strong foundation for competitive programming