



# Six-Month DSA Study Plan

---

## Overview

This structured six-month plan gradually introduces core DSA concepts and patterns, with 8-10 carefully selected practice questions each week. By the end of this plan, you'll have covered foundational topics and advanced concepts needed for top-tier technical interviews.

---

## Special Resources

1. [20 Patterns to Crack MAANG](#)
  2. [14 Tricks to Identify Patterns](#)
- 

## Month 1: Arrays and Strings

### Week 1: Arrays Basics & Simple Patterns

- Topics: Two-pointer, Sliding Window, HashMap basics
- Patterns: Prefix Sum, Running Sum

### Practice

1. [Easy] [Two Sum](#)
  2. [Easy] [Contains Duplicate](#)
  3. [Easy] [Maximum Subarray \(Kadane's Algorithm\)](#)
  4. [Medium] [Best Time to Buy and Sell Stock](#)
  5. [Medium] [Product of Array Except Self](#)
  6. [Medium] [Subarray Sum Equals K](#)
  7. [Medium] [Longest Subarray with Sum K](#)
  8. [Medium] [Minimum Size Subarray Sum](#)
- 

### Week 2: Advanced Array Manipulations

- Topics: Sorting, Interval Overlap, Cyclic Sort

- **Patterns:** Sorting + Two-pointer, Merge Intervals

#### Practice

1. [Easy] [Sort Colors \(Dutch National Flag Problem\)](#)
  2. [Medium] [Merge Intervals](#)
  3. [Medium] [Set Matrix Zeros](#)
  4. [Medium] [Next Permutation](#)
  5. [Medium] [Rotate Image](#)
  6. [Medium] [Find the Duplicate Number](#)
  7. [Medium] [Kth Largest Element in an Array](#)
  8. [Medium] [First Missing Positive](#)
- 

#### Week 3: Strings - Basics and Manipulations

- **Topics:** HashMaps for Strings, Substrings
- **Patterns:** Sliding Window, Hashing

#### Practice

1. [Easy] [Longest Common Prefix](#)
  2. [Easy] [Valid Anagram](#)
  3. [Medium] [Longest Substring Without Repeating Characters](#)
  4. [Medium] [Minimum Window Substring](#)
  5. [Medium] [Find All Anagrams in a String](#)
  6. [Medium] [Group Anagrams](#)
  7. [Medium] [Longest Palindromic Substring](#)
  8. [Medium] [Decode Ways](#)
- 

#### Week 4: Advanced String Manipulations

- **Topics:** Substrings and Subsequences

- Patterns: Two-pointer, Dynamic Programming (DP) for Strings

#### Practice

1. [Easy] [Implement strStr\(\)](#)
  2. [Medium] [Longest Common Subsequence](#)
  3. [Medium] [Edit Distance](#)
  4. [Medium] [Palindromic Substrings](#)
  5. [Medium] [Partition Labels](#)
  6. [Hard] [Minimum Window Substring](#)
  7. [Hard] [Substring with Concatenation of All Words](#)
  8. [Hard] [Wildcard Matching](#)
- 

## Month 2: Linked Lists and Binary Trees

### Week 1: Linked Lists Basics

- Topics: Singly and Doubly Linked Lists, Basic Operations
- Patterns: Two-pointer, Fast & Slow Pointers

#### Practice

1. [Easy] [Reverse Linked List](#)
  2. [Easy] [Middle of the Linked List](#)
  3. [Medium] [Linked List Cycle](#)
  4. [Medium] [Intersection of Two Linked Lists](#)
  5. [Medium] [Add Two Numbers](#)
  6. [Medium] [Remove N-th Node from End](#)
  7. [Medium] [Flatten a Multilevel Doubly Linked List](#)
  8. [Hard] [Merge K Sorted Lists](#)
- 

### Week 2: Advanced Linked Lists

- Topics: Complex List Manipulations, Reversal in Groups
- Patterns: Fast & Slow Pointers, Recursive Linked List

## Practice

1. [Medium] [Reverse Nodes in k-Group](#)
  2. [Medium] [Rotate List](#)
  3. [Medium] [Copy List with Random Pointer](#)
  4. [Medium] [Partition List](#)
  5. [Medium] [LRU Cache](#)
  6. [Hard] [LRU Cache](#)
  7. [Hard] [Reverse Nodes in k-Group](#)
  8. [Hard] [Insert into a Cyclic Sorted List](#)
- 

## Week 3: Binary Trees Basics

- Topics: Binary Tree Traversals (Inorder, Preorder, Postorder)
- Patterns: DFS and BFS Traversals

## Practice

1. [Easy] [Binary Tree Inorder Traversal](#)
2. [Easy] [Binary Tree Preorder Traversal](#)
3. [Easy] [Binary Tree Postorder Traversal](#)
4. [Medium] [Binary Tree Level Order Traversal](#)
5. [Medium] [Binary Tree Zigzag Level Order Traversal](#)
6. [Medium] [Binary Tree Right Side View](#)
7. [Medium] [Binary Tree Left Side View](#)
8. [Medium] [Binary Tree Maximum Path Sum](#)

## Week 4: Advanced Trees

- Topics: Tree Properties, Recursive & Iterative Solutions
- Patterns: Depth-First Search, Breadth-First Search
- Practice:

1. [Easy] [Symmetric Tree](#)
2. [Medium] [Lowest Common Ancestor of a Binary Tree](#)
3. [Medium] [Binary Tree Level Order Traversal II](#)

4. [Medium] [Serialize and Deserialize Binary Tree](#)
  5. [Medium] [Populating Next Right Pointers in Each Node](#)
  6. [Hard] [Recover Binary Search Tree](#)
  7. [Hard] [Flatten Binary Tree to Linked List](#)
  8. [Hard] [Binary Tree Cameras](#)
- 

## Month 3: Graphs and Recursion

### Week 1: Graphs Basics

- **Topics:** Graph Representations, BFS, DFS
  - **Patterns:** Breadth-First Search, Depth-First Search
  - **Practice:**
    1. [Easy] [Flood Fill](#)
    2. [Medium] [Number of Islands](#)
    3. [Medium] [Surrounded Regions](#)
    4. [Medium] [Rotting Oranges](#)
    5. [Medium] [Word Ladder](#)
    6. [Medium] [Clone Graph](#)
    7. [Medium] [Course Schedule](#)
    8. [Medium] [Detect Cycle in Undirected Graph](#)
    9. [Hard] [Minimum Number of Swaps for Sorting](#)
- 

### Week 2: Advanced Graph Algorithms

- **Topics:** Topological Sort, Connected Components
  - **Patterns:** Topological Sort, Union-Find
  - **Practice:**
    1. [Medium] [Alien Dictionary](#)
    2. [Medium] [Course Schedule II](#)
    3. [Medium] [Reconstruct Itinerary](#)
-

4. [Medium] [Connected Components in Graph](#)
  5. [Medium] [Accounts Merge](#)
  6. [Hard] [Find Critical and Pseudo-Critical Edges](#)
  7. [Hard] [Redundant Connection](#)
  8. [Hard] [Graph Valid Tree](#)
- 

### Week 3: Recursion and Backtracking

- **Topics:** Basics of Recursion, Permutations, Combinations
- **Patterns:** Backtracking, DFS
- **Practice:**
  1. [Easy] [Subsets](#)
  2. [Medium] [Combination Sum](#)
  3. [Medium] [Permutations](#)
  4. [Medium] [Palindrome Partitioning](#)
  5. [Hard] [N-Queens](#)
  6. [Hard] [Sudoku Solver](#)
  7. [Hard] [Word Search II](#)
  8. [Hard] [K-th Permutation Sequence](#)

### Week 4: Advanced Backtracking Problems

- **Topics:** Constraint Satisfaction, Optimizations
- **Patterns:** Recursion + Memoization
- **Practice:**
  1. [Medium] [Generate Parentheses](#)
  2. [Medium] [Letter Combinations of a Phone Number](#)
  3. [Medium] [Partition to K Equal Sum Subsets](#)
  4. [Hard] [Restore IP Addresses](#)
  5. [Hard] [Expression Add Operators](#)
  6. [Hard] [Remove Invalid Parentheses](#)

7. [Hard] [Count of Unique BSTs](#)

8. [Hard] [Word Search](#)

---

## Month 4: Dynamic Programming (DP) - Part 1

### Week 1: Introduction to Dynamic Programming

- Topics: Fundamentals of DP, 1D DP
- Patterns: Memoization, Tabulation
- Practice:
  1. [Easy] [Climbing Stairs](#)
  2. [Medium] [House Robber](#)
  3. [Medium] [House Robber II](#)
  4. [Medium] [Decode Ways](#)
  5. [Medium] [Maximum Subarray](#)
  6. [Medium] [Jump Game](#)
  7. [Medium] [Longest Increasing Subsequence](#)
  8. [Hard] [Longest Substring with K Distinct Characters](#)

### Week 2: Intermediate DP

- Topics: 0/1 Knapsack and Variants
- Patterns: Knapsack, Subset Sum
- Practice:
  1. [Medium] [Partition Equal Subset Sum](#)
  2. [Medium] [Combination Sum IV](#)
  3. [Medium] [Coin Change](#)
  4. [Medium] [Target Sum](#)
  5. [Medium] [Ones and Zeros](#)
  6. [Medium] [Minimum Subset Sum Difference](#)
  7. [Hard] [Word Break](#)

## Week 3: Advanced DP - Sequence Alignment

- Topics: Longest Common Subsequence and Variants
- Patterns: Substring and Subsequence Problems
- Practice:
  1. [Medium] [Longest Common Subsequence](#)
  2. [Medium] [Longest Common Substring](#)
  3. [Medium] [Minimum Insertions/Deletions](#)
  4. [Medium] [Edit Distance](#)
  5. [Medium] [Longest Palindromic Subsequence](#)
  6. [Medium] [Sequence Pattern Matching](#)
  7. [Hard] [Palindrome Partitioning II](#)
  8. [Hard] [Distinct Subsequences](#)

---

## Week 4: Advanced DP - Matrix and Grid Problems

- Topics: 2D DP, Grid Path Problems
- Patterns: Dynamic Programming on Grids
- Practice:
  1. [Easy] [Unique Paths](#)
  2. [Medium] [Unique Paths II](#)
  3. [Medium] [Minimum Path Sum](#)
  4. [Medium] [Dungeon Game](#)
  5. [Medium] [Cherry Pickup](#)
  6. [Hard] [Interleaving String](#)
  7. [Hard] [Maximal Rectangle](#)
  8. [Hard] [Largest Rectangle in Histogram](#)

Here's the formatted version for easy copying into a Word document:

---

## Month 5: Greedy Algorithms, Advanced Dynamic Programming, and Heaps

---

## Week 1: Greedy Algorithms - Intermediate Problems

- **Topics:** Interval Scheduling, Partitioning, Activity Selection
- **Patterns:** Interval Problems, Greedy Choice Property
- **Practice:**
  1. [Easy] [Best Time to Buy and Sell Stock II](#)
  2. [Medium] [Jump Game](#)
  3. [Medium] [Jump Game II](#)
  4. [Medium] [Gas Station](#)
  5. [Medium] [N Meetings in One Room](#)
  6. [Medium] [Partition Labels](#)
  7. [Medium] [Task Scheduler](#)
  8. [Hard] [Minimum Number of Refueling Stops](#)

---

## Week 2: Heaps - Basics and Applications

- **Topics:** Heap Basics, Priority Queue, Top-K Problems
- **Patterns:** Min-Heap, Max-Heap
- **Practice:**
  1. [Easy] [Kth Largest Element in a Stream](#)
  2. [Medium] [Kth Largest Element in an Array](#)
  3. [Medium] [Top K Frequent Elements](#)
  4. [Medium] [K Closest Points to Origin](#)
  5. [Medium] [Sort Characters By Frequency](#)
  6. [Hard] [Find Median from Data Stream](#)
  7. [Hard] [Merge K Sorted Lists](#)
  8. [Hard] [Sliding Window Maximum](#)

## Week 3: Advanced Dynamic Programming - Optimization Problems

- **Topics:** DP on Trees, Knapsack Variants
- **Patterns:** 0/1 Knapsack, DP with Trees
- **Practice:**

1. [Medium] [Partition Equal Subset Sum](#)
  2. [Medium] [Ones and Zeroes](#)
  3. [Medium] [Coin Change](#)
  4. [Medium] [Target Sum](#)
  5. [Medium] [Tree Diameter](#)
  6. [Hard] [Longest Increasing Path in a Matrix](#)
  7. [Hard] [Maximal Square](#)
  8. [Hard] [Burst Balloons](#)
- 

#### Week 4: Advanced Greedy Algorithms & Special Topics

- **Topics:** Advanced Greedy Techniques, Miscellaneous
- **Patterns:** Greedy + Dynamic Programming, Hybrid Patterns
- **Practice:**
  1. [Medium] [Minimum Number of Arrows to Burst Balloons](#)
  2. [Medium] [Reduce Array Size to The Half](#)
  3. [Medium] [Divide Array in Sets of K Consecutive Numbers](#)
  4. [Hard] [Minimum Cost to Hire K Workers](#)
  5. [Hard] [IPO](#)
  6. [Hard] [Maximum Profit in Job Scheduling](#)
  7. [Hard] [Candy](#)

#### Month 6: Tries, Advanced Data Structures, and Comprehensive Revision

---

#### Week 1: Tries and Prefix Trees

- **Topics:** Trie Basics, String Manipulations with Trie
- **Patterns:** Trie Structure, Prefix Matching
- **Practice:**
  1. [Easy] [Implement Trie \(Prefix Tree\)](#)
  2. [Medium] [Add and Search Word - Data Structure Design](#)
  3. [Medium] [Word Search II](#)

4. [Medium] [Replace Words](#)
  5. [Medium] [Design Search Autocomplete System](#)
  6. [Hard] [Palindrome Pairs](#)
  7. [Hard] [Concatenated Words](#)
  8. [Hard] [Maximum XOR of Two Numbers in an Array](#)
- 

## Week 2: Segment Trees and Binary Indexed Trees

- Topics: Range Queries, Lazy Propagation
- Patterns: Segment Tree, Fenwick Tree
- Practice:
  1. [Medium] [Range Sum Query - Mutable](#)
  2. [Medium] [Range Sum Query 2D - Mutable](#)
  3. [Medium] [Count of Smaller Numbers After Self](#)
  4. [Medium] [Kth Largest Element in an Array](#)
  5. [Hard] [Reverse Pairs](#)
  6. [Hard] [The Skyline Problem](#)
  7. [Hard] [Maximum Sum of Rectangle No Larger Than K](#)

## Week 3: Final Revision - Arrays, Strings, Linked Lists

- Topics: Comprehensive Review and Practice
  - Practice:
    1. Review top 3-5 questions from Arrays and Strings sections
    2. Review top 3-5 questions from Linked Lists
    3. Practice Mock Interviews with a random mix of problems
    4. Focus on any weak areas or topics that need additional review
- 

## Week 4: Final Revision - Trees, Graphs, DP, and Advanced Topics

- Topics: Comprehensive Review and Practice
- Practice:
  1. Review top 3-5 questions from Trees and Graphs

2. Review top 3-5 questions from Dynamic Programming
3. Practice Mock Interviews with a random mix of problems
4. Focus on polishing solutions for harder problems or any weak areas

 Connect with Me

GitHub: <https://github.com/iankushsingh>