# 8-Week DSA Timetable for Beginners

This 8-week DSA timetable is designed for beginners to build a strong foundation in Data Structures and Algorithms (DSA). Dedicate 2–3 hours daily to cover topics, solve problems, and revise for optimal learning.

## Week 1: Arrays and Basic Mathematics

|  |  |  |
| --- | --- | --- |
| Day | Topics / Questions to Solve | Time (Hours) |
| Day 1 | Array traversal, finding max/min in an array | 2 |
| Day 2 | Reverse an array, check if the array is sorted | 2 |
| Day 3 | Kth smallest/largest element, Sieve of Eratosthenes | 3 |
| Day 4 | Practice questions on arrays from LeetCode (Easy-Medium) | 3 |
| Day 5 | Solve GCD/LCM problems, check for prime numbers | 2 |
| Day 6 | Revise all concepts and practice more problems | 3 |
| Day 7 | Weekly review with a mix of Easy and Medium problems | 3 |

## Week 2: Strings and Hashing

|  |  |  |
| --- | --- | --- |
| Day | Topics / Questions to Solve | Time (Hours) |
| Day 1 | String basics, reverse a string, check if it’s a palindrome | 2 |
| Day 2 | Count character frequencies, anagram checks | 2 |
| Day 3 | Longest substring without repeating characters | 3 |
| Day 4 | Practice Easy-Medium string problems on LeetCode | 3 |
| Day 5 | Introduction to Hashmaps: frequency-based problems | 3 |
| Day 6 | Revise all concepts and solve new problems | 3 |
| Day 7 | Weekly review with mixed questions (mock test) | 3 |

## Week 3: Recursion and Backtracking

|  |  |  |
| --- | --- | --- |
| Day | Topics / Questions to Solve | Time (Hours) |
| Day 1 | Basics of recursion: factorial, Fibonacci | 2 |
| Day 2 | Generate subsets of an array | 3 |
| Day 3 | Permutations of a string | 3 |
| Day 4 | N-Queens problem | 3 |
| Day 5 | Solve backtracking problems from LeetCode (Medium-Hard) | 3 |
| Day 6 | Revise recursion/backtracking concepts | 3 |
| Day 7 | Weekly review with mixed problems | 3 |

## Week 4: Searching and Sorting

|  |  |  |
| --- | --- | --- |
| Day | Topics / Questions to Solve | Time (Hours) |
| Day 1 | Binary search on sorted arrays | 2 |
| Day 2 | Implement Bubble Sort and Selection Sort | 2 |
| Day 3 | Solve problems on Search in Rotated Sorted Arrays | 3 |
| Day 4 | Learn and implement Merge Sort | 3 |
| Day 5 | Count inversions in an array | 3 |
| Day 6 | Revise sorting techniques and solve related problems | 3 |
| Day 7 | Weekly review with Medium/Hard problems | 3 |

## Week 5: Linked Lists

|  |  |  |
| --- | --- | --- |
| Day | Topics / Questions to Solve | Time (Hours) |
| Day 1 | Basics of linked lists: Insertion and Deletion | 2 |
| Day 2 | Reverse a linked list | 2 |
| Day 3 | Merge two sorted linked lists | 3 |
| Day 4 | Detect a loop in a linked list | 3 |
| Day 5 | Solve problems like adding two numbers represented by linked lists | 3 |
| Day 6 | Revise and practice new problems | 3 |
| Day 7 | Weekly review (mix of Medium/Hard problems) | 3 |

## Week 6: Stacks and Queues

|  |  |  |
| --- | --- | --- |
| Day | Topics / Questions to Solve | Time (Hours) |
| Day 1 | Implement Stack and Queue using arrays | 2 |
| Day 2 | Balanced Parentheses problem | 2 |
| Day 3 | Implement Queue using two Stacks | 3 |
| Day 4 | Solve Largest Rectangle in Histogram | 3 |
| Day 5 | Sliding Window Maximum | 3 |
| Day 6 | Revise all concepts and solve problems | 3 |
| Day 7 | Weekly review with mixed problems | 3 |

## Week 7: Binary Trees

|  |  |  |
| --- | --- | --- |
| Day | Topics / Questions to Solve | Time (Hours) |
| Day 1 | Binary tree traversals: Preorder, Inorder, Postorder | 2 |
| Day 2 | Check if two trees are identical | 2 |
| Day 3 | Find the height of a binary tree | 3 |
| Day 4 | Solve Diameter of a Binary Tree problem | 3 |
| Day 5 | Lowest Common Ancestor problem | 3 |
| Day 6 | Revise and practice tree-based problems | 3 |
| Day 7 | Weekly review (Medium/Hard problems) | 3 |

## Week 8: Graphs

|  |  |  |
| --- | --- | --- |
| Day | Topics / Questions to Solve | Time (Hours) |
| Day 1 | Learn BFS traversal | 2 |
| Day 2 | Learn DFS traversal | 2 |
| Day 3 | Solve problems on cycle detection in undirected graphs | 3 |
| Day 4 | Solve Dijkstra's shortest path algorithm | 3 |
| Day 5 | Implement and solve Topological Sort | 3 |
| Day 6 | Revise all graph concepts | 3 |
| Day 7 | Final weekly review (mock problems) | 3 |