



Data Structures & Algorithms – Progressive Assessment Syllabus

This program is designed to build **strong problem-solving skills** through a **3-level assessment model — Easy, Medium, and Hard** — aligned with real interview and placement standards.

Assessment Structure

- **Easy** → Concept clarity & basic logic
- **Medium** → Pattern recognition & optimized thinking
- **Hard** → Advanced logic, constraints, and interview-level problems

Each day focuses on **one core topic**, gradually increasing in complexity.

Complete Topic-Wise Syllabus

Recursion & Backtracking

- Divide & Conquer principles
- Recurrence relations & recursion trees
- Backtracking decision trees
- Subsets, subsequences, combinations
- N-Queens problem (constraint satisfaction)

Searching Techniques

- Binary Search fundamentals
- Boundary conditions & infinite loop traps
- Binary Search on Answer (decision-based search)
- Optimization problems (allocation, feasibility checks)

Sorting Algorithms

- Bubble, Insertion, Selection sorting
- Custom comparator & lambda logic
- Interval sorting & merging
- Greedy scheduling using sorting
- Heap-based sorting (Heap Sort)

Linked List

- Reverse Linked List (iterative & recursive)
- Cycle detection (slow & fast pointers)
- Merging sorted linked lists
- Intersection point detection
- Palindrome linked list check



- LRU Cache design using DLL + HashMap

Stack

- Stack operations & expression evaluation
- Infix → Postfix conversion
- Monotonic stack patterns
- Next Greater Element
- Largest Rectangle in Histogram

Queue

- Queue basics & BFS introduction
- Circular Queue implementation
- Monotonic Queue
- Sliding Window Maximum / Minimum

Priority Queue / Heap

- Min Heap & Max Heap fundamentals
- Heapify & insertion logic
- Top-K problems
- Kth largest / smallest element
- Frequency-based sorting using heaps

Trees & BST

- Tree traversals (Preorder, Inorder, Postorder)
- Recursive & iterative traversal techniques
- Binary Search Tree operations
- BST validation using boundary logic

HashMap Techniques

- Frequency counting
- Prefix Sum technique
- Subarray sum equals K
- Grouping data (Anagrams, pair counting)

Graph Algorithms

- BFS traversal using adjacency lists
- Shortest path using Dijkstra's Algorithm



- Topological sorting (DFS & BFS approaches)
- DAG dependency resolution

Dynamic Programming

- Fibonacci with memoization
- DP vs recursion
- Longest Common Subsequence (2D DP)
- Grid problems (Unique Paths, Minimum Path Sum)

Greedy Algorithms

- Activity Selection problem
- Interval scheduling
- Job sequencing for maximum profit
- Greedy + DP hybrid problems