DSA Preparation Plan

🗓️ DSA Preparation Plan

✅ 1 – Core Java + Linear DS

- Java syntax & OOP (encapsulation, inheritance, polymorphism)
- Arrays, Strings, ArrayList, LinkedList
- Stack, Queue, Deque
- 10–15 easy problems on LeetCode or HackerRank

✅ 2 – Non-Linear DS + Recursion

- Binary Tree, BST, Heap
- Recursion basics and tree traversal
- Introduction to Graphs
- Solve problems: DFS, BFS, Tree traversals

✅ 3 – Algorithms

- Sorting (Merge, Quick, Counting)
- Searching (Binary Search, Search in Rotated Array)
- Greedy: Activity Selection, Huffman
- Dynamic Programming: Fibonacci, Knapsack
- Practice 10–12 problems (medium-level)

✅ 4 – Advanced + Patterns

- HashMap, HashSet, Trie, Disjoint Set
- Sliding Window, Two Pointer, Bit Manipulation
- Advanced Graphs: Dijkstra, Union-Find
- Mock interviews, timed contests, and system design basics

DSA Preparation Plan

📚 Complete List of Data Structures and Algorithms

🧱 A. Data Structures

🔹 1. Linear Data Structures

Structure	Description
Array	Fixed-size sequential collection of elements
Linked List	Nodes connected by pointers
Stack	LIFO (Last In First Out)
Queue	FIFO (First In First Out)
Deque	Double-ended queue (insert/remove at both ends)
String	Sequence of characters (often treated as DS)

🔹 2. Non-Linear Data Structures

Structure	Description
Tree	Hierarchical DS with root and children
└── Binary Tree	Each node has at most 2 children
└── BST (Search Tree)	Binary Tree with ordering property
└── AVL Tree	Self-balancing BST
└── Red-Black Tree	Balanced BST with coloring rules
└── Segment Tree	Range query optimization
└── Fenwick Tree (Binary Indexed Tree)	Efficient prefix sums
└── Trie (Prefix Tree)	Fast word prefix search

Heap	Complete binary tree with heap property
└── Min Heap / Max Heap	Min/Max value at root
Graph	Set of nodes (vertices) connected by edges
└── Directed / Undirected	Graph types
└── Weighted / Unweighted	Edge weights
└── Adjacency Matrix / List	Representation

🔹 3. **Hashing Structures**

Structure	**Description**
Hash Table / HashMap	Key-value storage with fast lookup
HashSet	Stores unique elements using hashing
Custom Hash Functions	Used in advanced problems

🔹 **4. Other Specialized Data Structures**

Structure	**Description**
Disjoint Set (Union-Find)	Detecting connected components
LRU Cache (LinkedHashMap/Deque)	Least Recently Used caching
Priority Queue	Queue with element priority
Skip List	Fast search using layered linked lists
Bloom Filter	Probabilistic data structure for set membership
Suffix Array / Tree	String pattern searching
K-D Tree / Quad Tree	For multidimensional data (2D/3D)

DSA Preparation Plan

🧠 **B. Algorithms**

🔹 1. Sorting Algorithms

Algorithm	Time Complexity (Best/Worst)
Bubble Sort	$O(n) / O(n^2)$
Selection Sort	$O(n^2) / O(n^2)$
Insertion Sort	$O(n) / O(n^2)$
Merge Sort	O(n log n) / O(n log n)
Quick Sort	$O(n \log n) / O(n^2)$
Heap Sort	O(n log n) / O(n log n)
Counting Sort	O(n + k) (for integers)
Radix Sort	O(nk) (non-comparison sort)
Bucket Sort	O(n+k)

🔹 2. Searching Algorithms

Algorithm	Description
Linear Search	O(n) – search element in array
Binary Search	O(log n) – sorted array required
Ternary Search	Division into 3 parts
Exponential Search	For unbounded sorted arrays
Jump Search / Interpolation Search	Faster in some cases

DSA Preparation Plan

🔹 3. Recursion & Backtracking

Concept	Usage Example
Basic Recursion	Factorial, Fibonacci
Backtracking	N-Queens, Sudoku, Subset Sum
Memoization	Cache previous results

🔹 4. Divide and Conquer

Algorithm	Description
Merge Sort	Divide into halves
Quick Sort	Divide based on pivot
Binary Search	Repeated halving
Strassen's Matrix Multiplication	Matrix optimization

🔹 5. Greedy Algorithms

Algorithm	Usage
Kruskal’s Algorithm	Minimum Spanning Tree (MST)
Prim’s Algorithm	MST using priority queue
Activity Selection	Scheduling problems
Huffman Coding	Data compression
Fractional Knapsack	Maximize value

DSA Preparation Plan

🔹 6. Dynamic Programming (DP)

Problem Type	Technique
0/1 Knapsack	Tabulation, memoization
Fibonacci, LCS, LIS	Overlapping subproblems
Matrix Chain Multiplication	Optimal structure
Coin Change, Rod Cutting	Resource optimization
DP on Trees / Graphs	Complex structures

🔹 7. Graph Algorithms

Algorithm	Purpose
DFS, BFS	Traversal
Dijkstra’s Algorithm	Shortest Path (Non-negative)
Bellman-Ford	Handles negative weights
Floyd-Warshall	All-pairs shortest path
Topological Sort	DAG processing
Kruskal / Prim	Minimum Spanning Tree (MST)
Union-Find	Connected components detection
Tarjan’s Algorithm	SCC / Bridges / Articulation Points

DSA Preparation Plan

🔹 8. Advanced Algorithms

Algorithm	Use Case
KMP Algorithm	Pattern matching (strings)
Rabin-Karp	Rolling hash for string match
Manacher’s Algorithm	Longest palindromic substring
*A Search**	Pathfinding in games/AI
Min-Cut/Max-Flow (Ford–Fulkerson)	Network flow
Suffix Arrays / Trees	Fast pattern searching

📦 Common Interview Patterns

Pattern	Use Case
Two Pointers	Sorting, searching, arrays
Sliding Window	Subarrays, strings
Fast & Slow Pointer	Cycles in linked lists
Merge Intervals	Calendar, scheduling
Monotonic Stack/Queue	Stock span, nearest greater/smaller
Bit Manipulation	XOR tricks, toggling bits
Trie	Dictionary matching, autocomplete

DSA Preparation Plan

📅 Suggested Learning Schedule

Sr No	Focus	Practice Tool
1	Arrays, Strings, Recursion	LeetCode Easy
2	Linked List, Stack, Queue	GFG/LeetCode
3	Trees, HashMap, Searching/Sorting	Coding Ninjas
4	DP, Graphs, Backtracking	LeetCode Medium/Scaler

📚 GitHub Repositories

- 🔗 The Algorithms - Java: All major DSA implementations
- 🔗 mission-peace/interview: Java-centric DSA explanations and problems

---

🚀 Top Platforms to Learn DSA in Java (Free + Paid)

✅ 1. GeeksforGeeks DSA in Java Track

- **Level**: Beginner → Advanced
- **What's Good**: Strong focus on Java-based implementations with quizzes & practice problems.
- **Includes**: Arrays, Strings, Trees, Graphs, DP, Recursion, and more.
- **Price**: Some free content, full course is paid (~₹500–1000).

✅ 2. Coding Ninjas – DSA in Java

- **Level**: Beginner-friendly with mentorship
- **What's Good**: Structured course + assignments + placement prep
- **Java Focus**: Fully Java-focused curriculum
- **Price**: Paid (~₹5000–₹9000); offers EMI and scholarships

---

DSA Preparation Plan

✅ 3. Scaler Academy (by InterviewBit)

- **Level**: Intermediate to Advanced (for serious aspirants)
- **Includes**: Live classes, mock interviews, Java-based system design
- **What's Good**: Top mentors, structured job-driven DSA curriculum
- **Price**: Paid (Premium course, EMI available)

✅ 4. freeCodeCamp YouTube – DSA in Java

- **Level**: Beginner
- **What's Good**: Free, beginner-friendly, well-explained visuals
- **Covers**: Recursion, Sorting, Trees, Graphs with Java code

---

✅ 5. LeetCode – Java Tag + Explore Cards

- **Practice-oriented platform**
- Filter problems by Java
- Explore cards for topics like Arrays, Linked List, Binary Search
- Great for **company-specific DSA prep**

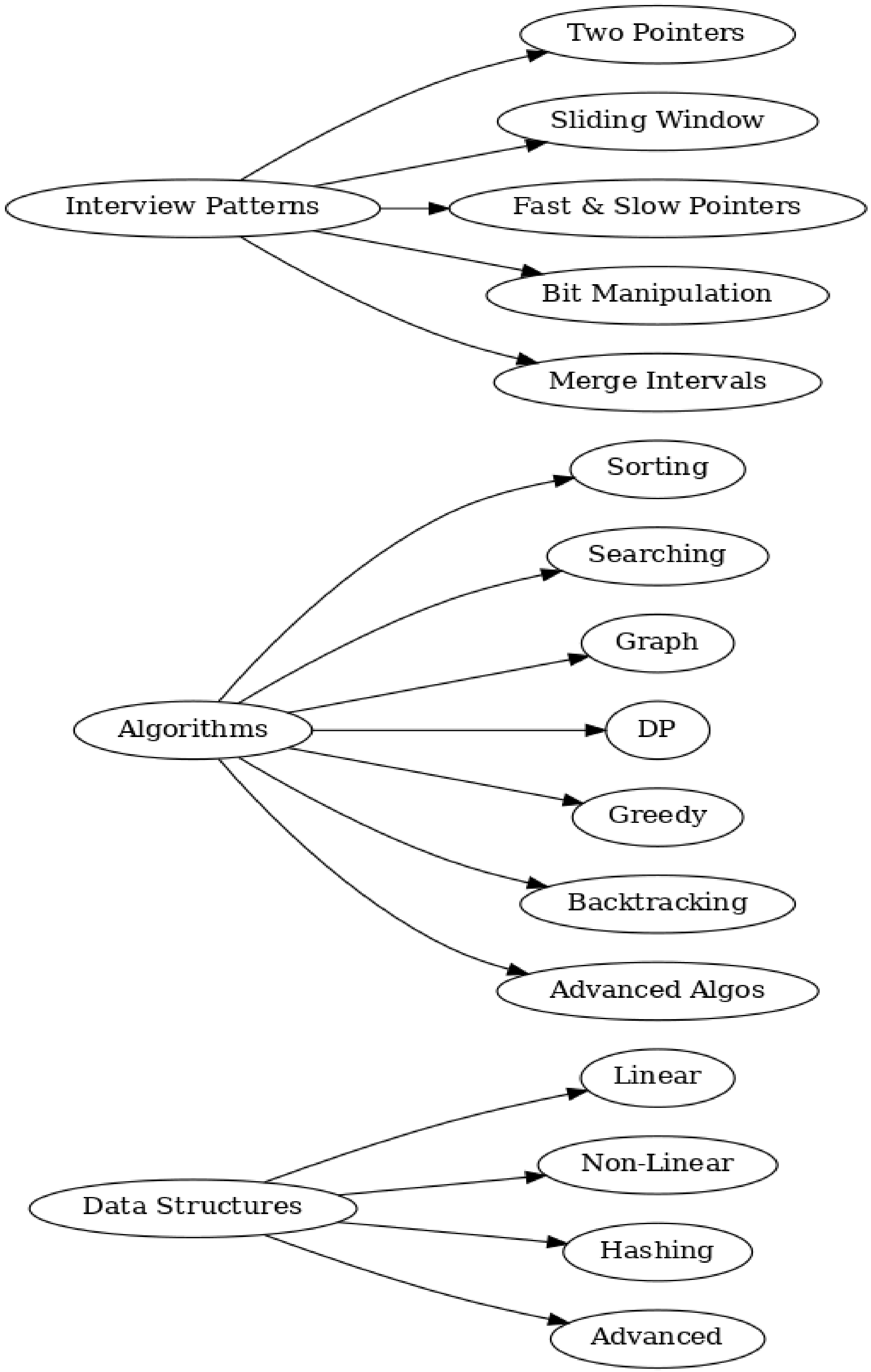
✅ 6. Udemy – Master DSA using Java

- **Instructor**: Tim Buchalka (highly rated)
- **Covers**: Core DSA in Java with clean visual explanations
- **Price**: Often discounted to ₹499–₹799

✅ 7. Coursera – Java DSA by UC San Diego

- University-backed course
- Taught in Java
- Includes assignments and certificates

DSA Preparation Plan



Interview Patterns
Two Pointers
Sliding Window
Fast & Slow Pointers
Bit Manipulation
Merge Intervals
Algorithms
Sorting
Searching
Graph
DP
Greedy
Backtracking
Advanced Algos
Data Structures
Linear
Non-Linear
Hashing
Advanced