|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Focus | Core Topics | Goals & Milestones |
| Phase 0 (Prep ‑ 1 week) | Foundations | • Big‑O/Ω/Θ 🔎 • Math for DS (mod, log₂, combinatorics) | ‑ Know how to eyeball time/space quickly ‑ Install IDE, set up LeetCode/HackerRank accounts |
| Phase 1 (Essentials ‑ 4 weeks) | Linear DS & Easy Algos | **Arrays & Strings** ➜ sliding window, prefix‑sum, two‑pointer  **Linked Lists** ➜ fast/slow pointer, reversal  **Stacks/Queues** ➜ monotonic, BFS intro  **Hash Maps/Sets** ➜ frequency tables | ‑ Solve 50 easy LC questions ‑ Implement your own vector/linked‑list in Java |
| Phase 2 (Core Mastery ‑ 6 weeks) | Classic DS/Algos | **Sorting** ➜ quick, merge, heap‑sort; counting/radix for XXL data  **Binary Search** (on value & answer‑space)  **Recursion / Divide‑&‑Conquer** **Trees** ➜ BST, AVL, heap, segment, Fenwick  **Greedy Patterns** ➜ intervals, activity select, Huffman | ‑ Build a min‑heap from scratch ‑ Reach 150 LC solves (mix easy/medium) |
| Phase 3 (Problem‑Solving Beast ‑ 8 weeks) | Advanced & Competitive | **Dynamic Programming** ➜ 1‑D, 2‑D, bitmask, DP‑on‑trees  **Graph Algorithms** ➜ BFS/DFS variations, Dijkstra, Bellman–Ford, Topo sort, MST (Kruskal/Prim), Union‑Find, shortest path on DAG **Backtracking + Bit Tricks** ➜ subsets, permutations, N‑Queens **Tries & String Algos** ➜ KMP, Z‑algo, Rabin‑Karp  **Geometry, Math, Prob** basics | ‑ Tackle ≥20 hard LC problems ‑ Compete in 3‑4 weekly contests & note patterns |
| Phase 4 (Polish & Patterns ‑ ongoing) | Interview Mode | • Systematic pattern catalogue (below) • Mock interviews, whiteboard practice | ‑ Can verbalize trade‑offs on any DS ‑ 1‑shot whiteboard of fav hard problem |

**Ultimate DSA Conqueror Road Map**

**Go‑To Patterns**

|  |  |
| --- | --- |
| Problem Vibe | 🔑 Algorithm / Data Structure |
| Fast lookup / frequency | HashMap / HashSet |
| Ordered unique items | TreeSet / TreeMap |
| “Find kth / medians / top‑K” | Heap / Quick‑select |
| Window / substring stats | Sliding‑window + HashMap |
| “Min # of … while …” | Greedy + Sorting / Heap |
| Paths in un‑weighted graph | BFS |
| Shortest path weighted | Dijkstra (positive) / Bellman‑Ford (neg) |
| Connectivity / cycles | Union‑Find (DSU) |
| “All possibilities / combos” | Backtracking + pruning |
| “Min cost to reach state” | Dynamic Programming |
| Range query + updates | SegmentTree / Fenwick |
| Search in monotonic space | Binary Search‑on‑answer |
| Lexicographic string search | Trie / KMP / Z‑algo |
| Scheduling / intervals | Sort + Greedy sweep‑line |
| 2‑sum / 3‑sum / k‑sum | Two‑pointer + HashSet |
| “Counting pairs with X property” | Prefix‑sum / Hash Freq |
| Large integer math | Bitwise DP / BigInteger |
| Flow / matching | Dinic / Kuhn (advanced) |