

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

Week 7

Week 8

Build a Strong Foundation in Data Structures and Algorithms 

(

1. Daily routine (weekdays)

- Warm-up (15 min): Flashcards of yesterday's key ideas.
- Concept block (45-60 min): Read/watch, take concise notes.
- Coding block (120 min): Solve 3-5 problems; first unguided, then compare with discuss/solutions.
- Reflection (15-30 min): Summarise patterns, add to cheatsheet.

2. Weekend deep work

- Morning block (3-4 h): Build/implement something from scratch.
- Afternoon block (3-4 h): Large problem set or timed contest.
- Evening block (2 h): Writeups, visualisations, or lightweight reading.

3. Tools & hygiene

- Spaced repetition: Anki cards for formulas, templates, common pitfalls.
- Code journal: Keep a Git repo; each solved LeetCode problem gets its own markdown explaining intuitions & complexities.

- Weekly retrospective (Sun night): What clicked? What needs extra reps? Tweak next week's plan accordingly.

)

Week Focus (Step #) Mon – Fri (4 h each) Sat (9 h) Sun (9 h) Key resources (pick one or two to start, keep the rest for reference)

1 Foundations ①Arrays & Strings + Big-O 1 h concept reading/notes 2h LeetCode ► Easy arrays/strings (3–5 q.). 1 h code review & spaced-repetition flashcards 3 h deep-dive: sliding-window & two-pointer patterns. 3 h solve 8–10 new problems. 3 h reflect → write “lesson learned” summaries 3 h mock white-board: re-implement top 5 array problems from scratch 3 h timed drills (15 min/problem). 5 h rest + light reading (e.g., CLRS §2) • LeetCode Explore → Array & String card (leetcode.com) • GeeksforGeeks “Getting Started with Arrays” article (geeksforgeeks.org)

2 Foundations ①Linked Lists, Stacks, Queues, Hash Tables Same structure, topics rotate each day 4 h mini-project: build your own singly & doubly linked list classes + tests. 5 h LeetCode mixed set (10–12 q.) 4 h hashmap/stack challenge set (parentheses, LRU cache, monotonic stack). 5 h review & annotate solutions • GeeksforGeeks Window Sliding Technique guide (geeksforgeeks.org) • Medium deep-dive Mastering Sliding Window (medium.com)

Foundations ①Sorting/Searching + Trees (BST, traversal) Concept → practice loop as above 4 h sort implementations from memory (quick, merge, heap) + complexity proofs. 5 h tree DFS/BFS problems 4 h binary search variations sprint. 5 h mixed review quiz + spaced-rep • LeetCode Explore → Linked List card (leetcode.com)

4 Financial Algos ②Stock I & II + basic DP 1 h read editorial + annotate recurrence. 2 h code & debug problems 1 h compare top submissions 4 h “Stock I & II” re-implement + add profit-curve plot in Python. 5 h build Jupyter notebook to back-test your solutions on real OHLCV data 4 h timed contest with only stock-family questions. 5 h reflect: write blog-style post explaining max-profit DP • LeetCode Top-Interview “Array & Strings” section (great two-pointer drills) (leetcode.com)

5 Financial Algos ②Stock III, IV & Cool-down Same cadence; focus on state-machine DP 4 h derive state diagrams for each variant. 5 h implement + unit tests 4 h param-sweep experiments (vary k, fee, cooldown) and chart results 5 h create summary cheatsheet for future reference • GfG LRU Cache – Complete Tutorial (geeksforgeeks.org) • takeUforward LRU implementation walkthrough (takeuforward.org)

6 Advanced Topics ③Advanced DP + BIT/Fenwick & Segment Trees Mon-Wed: knapsack, LIS, DP optimisation Thu-Fri: BIT intro + range queries 4 h code BIT & seg-tree from scratch + visualize operations. 5h medium/hard LeetCode using these structures 4 h contest simulation (rated hard set) 5 h performance review & refactor • GfG Merge Sort article with code & proofs (geeksforgeeks.org)

7 Advanced Topics ③Graphs (Dijkstra, Bellman-Ford, Floyd, Toposort) Concept → practice loop; use weighted graph problems 4 h implement each algo from scratch + unit tests. 5 h “hard” graph set (sp shortest path, course schedule, MST) 4 h create Python network-analysis

notebook; apply Dijkstra to real crypto order-book snapshot. 5 h write summary & mind-map •
GfG Array & Strings interview drills page (binary-search variants) (leetcode.com)
8 Advanced Topics ③String Matching + Wrap-Up Integration Mon: KMP Tue: Rabin-Karp. Wed:
Z-algo. Thu-Fri: Tries & autocomplete project 4 h build a trie-based ticker-symbol autocompleter.
5 h solve remaining hard string problems 4 h cumulative mock interview (mix of all topics, 8 q.).
5 h consolidate notes → final “Algorithmic-Trading Prep Handbook” PDF • GfG BST Traversals
– Inorder / Pre / Post ([geeksforgeeks.org](https://www.geeksforgeeks.org))