🚀 COMPLETE 12-MONTH PYTHON DSA ROADMAP (FINAL)

⏱️ Daily time: **2–3 hours**
🎯 End of 1 year: **Python SDE interview-ready**

🟢 MONTH 1 — Python Foundations + DSA Basics

Goal: Python interview safety + logic clarity

Python Core

- Variables, data types
- Input / Output (single & multiple inputs)
- if-else, loops
- break, continue, pass
- Functions
    - Parameters & return values
    - Pass by assignment (object reference)
    - Mutable vs immutable arguments ⭐
- List, tuple, set, dict
- enumerate, zip
- *args, **kwargs

Python Gotchas & Internals (🔥 CRITICAL)

- Mutable vs immutable ⭐
- is vs == ⭐
- Default argument pitfall ⭐
- Shallow vs deep copy ⭐
- Truthy vs falsy ⭐
- None vs False ⭐
- Reference behavior & memory basics ⭐

Python Performance Awareness

- list.append() vs insert()
- pop() vs pop(0)
- in → list vs set vs dict ⭐
- Dict hashing intuition
- Amortized complexity (list growth) ⭐
- In-place vs extra space ⭐
- Recursion depth & stack limit

DSA Basics

- What is DSA
- Time & space complexity
- Big-O: O(1), O(n), O(n^2)

Practice

- 20–25 easy problems
- Pattern + basic math problems

🎯 **Outcome:** Python-safe + strong logic foundation

🟢 **MONTH 2 — Arrays + Searching + Sorting + Math**

Goal: Index thinking + boundary discipline

Arrays

- 1D & 2D arrays
- Traversal
- Prefix sum
- Linear search
- Binary search (iterative & recursive)
- Boundary discipline ⭐
    - off-by-one errors
    - l <= r vs l < r

    - safe mid calculation

Array Patterns

- Two pointers (arrays)
- Sliding window (fixed size)
- In-place vs extra space tradeoffs ⭐

Sorting (Decision-Based)

- sort() vs sorted()
- Custom sorting (key, lambda)
- Stable vs unstable sort
- Multiple key sorting ⭐
- When to sort vs when to heap ⭐

Python Performance

- List slicing cost
- [::-1] vs reversed()
- min/max vs manual loop

Math

- GCD / LCM
- Prime check
- Sieve of Eratosthenes
- Modulo arithmetic
- Counting digits
- Power(x, n)

Practice

- 30–35 problems

🎯 **Outcome:** Array + math confidence

🟢 **MONTH 3 — Strings + Sliding Window + Prefix**

Goal: Pattern recognition speed

Strings

- String manipulation
- Character frequency
- Palindromes
- Anagrams
- Two pointers on strings ⭐

Sliding Window (Deep)

- Fixed vs variable window
- At most K distinct ⭐
- Exactly K distinct ⭐
- Longest vs shortest window ⭐

Prefix + Hashing

- Prefix sum + hashmap
- Subarray sum = K
- Count of subarrays ⭐
- Prefix vs sliding window logic

Python String Internals

- String immutability
- += vs join()
- Slicing cost

Python Tools

- Counter, defaultdict

Practice

- ~30 problems

🎯 **Outcome:** Fast string & window problem solving

🟢 **MONTH 4 — Hashing + Optimization Discipline**

Goal: Brute → optimal mindset

Hashing

- HashMap (dict)
- HashSet
- Frequency counting
- Lookup optimization

Hashing Internals

- Why average O(1)
- Collisions (high-level)
- Worst-case awareness

Hashing Patterns

- Hashing + arrays
- Hashing + strings
- Prefix sum + hashmap
- Why sliding window fails with negatives ⭐

Python Hashing Traps

- Mutable keys ❌
- Tuple keys ✅
- Dict order behavior

Interview Discipline (🔥 CRITICAL)

- Choosing correct DS ⭐
- Explaining time & space ⭐
- Space vs time tradeoffs ⭐

Fast I/O

- sys.stdin.readline
- sys.stdout.write

Practice

- ~35 problems

🎯 **Outcome:** Optimization confidence

🟢 MONTH 4.5 — OOP in Python (CRITICAL)

Goal: Clean design + interview clarity

Core OOP

- Class & object
- __init__
- Instance vs class variables
- Instance / class / static methods
- self

OOP Principles

- Encapsulation
- Inheritance
- Polymorphism
- Abstraction

Python-Specific OOP

- super()
- Multiple inheritance
- MRO
- Composition vs inheritance ⭐
- __str__, __repr__
- Operator overloading
- Abstract Base Classes (abc)

OOP Internals

- Public / protected / private
- Name mangling
- @property

- __eq__, __hash__
- Objects as dict keys

Practice

- Student / Employee system
- Bank account system
- Library management system

🎯 **Outcome:** OOP interview-ready

🟡 **MONTH 5 — Recursion + Backtracking**

Goal: Recursive clarity (controlled)

Recursion

- Call stack
- Base vs recursive case
- Recursion tree
- Time complexity
- Recursion vs iteration ⭐
- When recursion FAILS in Python ⭐

Backtracking

- Choose → Explore → Un-choose ⭐
- Subsets
- Permutations
- Combinations

Recursion → DP Bridge

- Overlapping subproblems
- Memoization idea ⭐

Practice

- 25–30 problems

🎯 **Outcome:** Controlled recursion

🟡 MONTH 6 — Linked List

Goal: Pointer confidence

- Singly & doubly linked list
- Python reference model
- Reverse list
- Fast & slow pointers
- Cycle detection
- Merge lists
- Middle of list
- Dummy / sentinel node ⭐
- Edge cases (0/1/2 nodes)

Practice: ~25 problems

🎯 **Outcome:** Linked list mastery

🟡 MONTH 7 — Stack + Queue + Bit Manipulation

Goal: Flow control + low-level logic

- Stack (via list)
- Stack vs recursion
- Valid parentheses
- Next greater element
- Queue vs stack
- collections.deque
- Sliding window maximum
- Monotonic stack (O(n) intuition)
- AND, OR, XOR
- XOR tricks

- Power of two
- Python bit behavior

Practice: ~30 problems

🎯 **Outcome:** Pattern fluency

🟡 **MONTH 8 — Trees + Heaps**

Goal: Core interview structures

- Binary tree, BST
- Tree properties
- DFS / BFS
- Recursive vs iterative ⭐
- LCA
- Tree as graph
- Parent mapping
- BFS from target
- Min / max heap
- heapq
- Top-K problems
- Heap vs sort decisions ⭐

Practice: 35–40 problems

🎯 **Outcome:** Tree + heap intuition

🔵 **MONTH 9 — Greedy + Interval Problems**

Goal: Decision optimization

- Greedy choice property ⭐
- Optimal substructure ⭐
- When greedy FAILS

- Interval sorting (start vs end) ⭐
- Interval merging
- Boundary handling ⭐
- Scheduling
- Greedy + heap

Practice: 25–30 problems

🎯 **Outcome:** Optimal decision making

🔵 **MONTH 10 — Graphs + Union-Find**

Goal: Complex relationship handling

- Adjacency list vs matrix ⭐
- Directed vs undirected graphs
- BFS / DFS
- BFS vs DFS decisions ⭐
- Cycle detection
- Topological sort (Kahn's)
- DAG logic
- Union-Find
- Path compression
- Union by rank
- When DSU applies / fails ⭐

Practice: ~30 problems

🎯 **Outcome:** Graph confidence

🔴 **MONTH 11 — Dynamic Programming (CRITICAL)**

Goal: Remove DP fear

DP Framework

- When to use DP
- State, transition, base
- DP vs recursion vs greedy ⭐
- Time & space complexity
- State explosion awareness

Techniques

- Memoization
- Tabulation
- Space optimization

Core Patterns

- 1D DP
- 2D DP
- Knapsack
- LIS
- Grid DP

Python DP Safety

- Avoid recursion TLE
- Prefer iterative DP ⭐

Practice: 25–30 problems

🎯 **Outcome:** DP-safe (not DP-panic)

🟢 **MONTH 12 — Revision + Interview Prep**

Goal: Offer-ready

Revision

- Revise by **patterns**
- Re-solve weak problems
- Maintain mistake log ⭐

Interview Practice

- Timed LeetCode
- Mock interviews (DSA + OOP)
- Thinking aloud ⭐
- Clean variable naming
- Helper functions
- Handling interruptions ⭐

Debugging

- Edge cases
- Dry runs
- Assertions
- Failure explanation

Optional (High Value)

- Python → Java (Top 30 problems)
- LRU Cache (Python)
- Simple class design

🎯 **FINAL OUTCOME** ✅

- ✅ Python SDE interview-ready
- ✅ Product-company compatible
- ✅ Strong problem-solving mindset
- ✅ AI / ML transition friendly

🟢 MONTH 1 — 30-DAY DAILY SCHEDULE

Goal: Python interview safety + logic clarity

🔹 WEEK 1 — Python Basics + Logic

🗓️ Day 1 — Variables, Data Types, I/O

Learn

- int, float, str, bool
- input(), multiple inputs
- type()

LeetCode

- 1480 — Running Sum of 1d Array
- 1920 — Build Array from Permutation

🗓️ Day 2 — if-else, Loops

Learn

- if / elif / else
- for, while

LeetCode

- 412 — Fizz Buzz
- 1342 — Number of Steps to Reduce a Number to Zero

🗓️ Day 3 — break, continue, pass

Learn

- Loop control
- pass usage

LeetCode

- 509 — Fibonacci Number
- 1281 — Subtract the Product and Sum of Digits of an Integer

---

🗓️ Day 4 — Functions Basics

Learn

- Parameters & return
- Multiple return values

LeetCode

- 1108 — Defanging an IP Address
- 1672 — Richest Customer Wealth

🗓️ Day 5 — Lists

Learn

- Indexing
- append, pop
- Iteration

LeetCode

- 217 — Contains Duplicate
- 1929 — Concatenation of Array

🗓️ Day 6 — Tuple, Set

Learn

- tuple immutability
- set uniqueness

LeetCode

- 1832 — Check if the Sentence Is Pangram
- 1684 — Count the Number of Consistent Strings

🗓️ Day 7 — dict (Basics)

Learn

- Key–value access
- dict iteration

LeetCode

- 1 — Two Sum
- 169 — Majority Element

🔹 **WEEK 2 — Python Gotchas & Internals**

🗓 **Day 8 — Mutable vs Immutable** ⭐

Learn

- list vs tuple
- function argument behavior

LeetCode

- 66 — Plus One
- 989 — Add to Array-Form of Integer

🗓 **Day 9 — is vs ==** ⭐

Learn

- identity vs equality

LeetCode

- 242 — Valid Anagram
- 205 — Isomorphic Strings

🗓 **Day 10 — Default Argument Pitfall** ⭐

Learn

- Mutable default arguments

LeetCode

- 58 — Length of Last Word
- 709 — To Lower Case

🗓️ Day 11 — Shallow vs Deep Copy ⭐

Learn

- copy module
- nested lists

LeetCode

- 867 — Transpose Matrix
- 766 — Toeplitz Matrix

🗓️ Day 12 — Truthy vs Falsy ⭐

Learn

- False values in Python

LeetCode

- 125 — Valid Palindrome
- 344 — Reverse String

🗓️ Day 13 — None vs False ⭐

Learn

- None comparisons
- return None traps

LeetCode

- 20 — Valid Parentheses
- 228 — Summary Ranges

🗓️ Day 14 — Reference Behavior ⭐

Learn

- Object references
- aliasing

LeetCode

- 283 — Move Zeroes
- 905 — Sort Array By Parity

🔹 **WEEK 3 — Python Performance Awareness**

🗓 **Day 15 — append vs insert**

Learn

- Time complexity

LeetCode

- 26 — Remove Duplicates from Sorted Array
- 27 — Remove Element

🗓 **Day 16 — pop() vs pop(0)**

Learn

- Why pop(0) is slow

LeetCode

- 977 — Squares of a Sorted Array
- 88 — Merge Sorted Array

🗓 **Day 17 — in → list vs set vs dict** ⭐

Learn

- Lookup complexity

LeetCode

- 349 — Intersection of Two Arrays
- 136 — Single Number

🗓 **Day 18 — Dict hashing intuition**

Learn

- Why dict is fast

LeetCode

- 387 — First Unique Character in a String
- 409 — Longest Palindrome

🗓 Day 19 — Amortized Complexity ⭐

Learn

- Dynamic resizing

LeetCode

- 724 — Find Pivot Index
- 303 — Range Sum Query – Immutable

🗓 Day 20 — In-place vs Extra Space ⭐

Learn

- Modify vs copy

LeetCode

- 189 — Rotate Array
- 283 — Move Zeroes (revisit, optimize)

🗓 Day 21 — Recursion Depth & Stack

Learn

- sys.setrecursionlimit
- recursion limits

LeetCode

- 206 — Reverse Linked List (recursive idea only)
- 231 — Power of Two

🔹 WEEK 4 — DSA Basics + Patterns

🗓️ Day 22 — What is DSA

Learn

- Why DSA matters

LeetCode

- 268 — Missing Number
- 441 — Arranging Coins

🗓️ Day 23 — Time & Space Complexity

Learn

- O(1), O(n), O(n²)

LeetCode

- 35 — Search Insert Position
- 69 — Sqrt(x)

🗓️ Day 24 — Big-O Thinking

Learn

- Brute vs optimized

LeetCode

- 121 — Best Time to Buy and Sell Stock
- 122 — Best Time to Buy and Sell Stock II

🗓️ Day 25 — Pattern: Counting

Learn

- Frequency logic

LeetCode

- 448 — Find All Numbers Disappeared in an Array
- 645 — Set Mismatch

🗓️ Day 26 — Pattern: Math

Learn

- Digits, sums

LeetCode

- 258 — Add Digits
- 171 — Excel Sheet Column Number

🗓️ Day 27 — Pattern: Prefix

Learn

- Cumulative logic

LeetCode

- 1480 — Running Sum (revisit)
- 1732 — Find the Highest Altitude

🗓️ Day 28 — Mixed Practice

LeetCode

- 13 — Roman to Integer
- 67 — Add Binary

🗓️ Day 29 — Revision Day

Do

- Re-solve 5 weak problems
- Revise Python gotchas ⭐

🗓️ Day 30 — Mock Day

Do

- Solve **3 problems in 60 minutes**

- Explain aloud
- Write clean code

Suggested:

- 1 — Two Sum
- 121 — Best Time to Buy and Sell Stock
- 20 — Valid Parentheses

🟢 MONTH 2 — 30-DAY DAILY SCHEDULE

Goal: Index thinking + boundary discipline

🔹 WEEK 1 — Array Basics + Traversal

🗓️ Day 1 — 1D Arrays Basics

Learn

- Indexing
- Traversal
- Length & boundaries

LeetCode

- 1920 — Build Array from Permutation
- 1929 — Concatenation of Array

🗓️ Day 2 — Array Traversal Patterns

Learn

- Forward & backward traversal

LeetCode

- 1295 — Find Numbers with Even Number of Digits
- 485 — Max Consecutive Ones

🗓️ Day 3 — 2D Arrays Basics

Learn

- Row/column traversal

LeetCode

- 867 — Transpose Matrix
- 1572 — Matrix Diagonal Sum

🗓️ **Day 4 — Linear Search**

Learn

- Sequential search
- Early exit logic

LeetCode

- 704 — Binary Search (just linear logic first)
- 1365 — How Many Numbers Are Smaller Than the Current Number

---

🗓️ **Day 5 — Prefix Sum (Basics)**

Learn

- Running sum concept

LeetCode

- 1480 — Running Sum of 1d Array
- 1732 — Find the Highest Altitude

🗓️ **Day 6 — Prefix Sum (Use Cases)**

Learn

- Prefix vs brute force

LeetCode

- 724 — Find Pivot Index
- 303 — Range Sum Query – Immutable

🗓️ **Day 7 — Weekly Revision**

Do

- Re-solve 3 weak array problems
- Focus on indexes & boundaries

🔹 **WEEK 2 — Binary Search + Boundary Discipline**

🗓️ Day 8 — Binary Search Basics

Learn

- l, r, mid
- l <= r loop

LeetCode

- 704 — Binary Search
- 35 — Search Insert Position

🗓️ Day 9 — Binary Search Variants

Learn

- First/last occurrence

LeetCode

- 34 — Find First and Last Position of Element
- 69 — Sqrt(x)

🗓️ Day 10 — Binary Search on Answer

Learn

- Search space idea

LeetCode

- 374 — Guess Number Higher or Lower
- 367 — Valid Perfect Square

🗓️ Day 11 — Recursive Binary Search

Learn

- Stack & base cases

LeetCode

- 704 — Binary Search (recursive)
- 278 — First Bad Version

🗓️ Day 12 — Boundary Discipline ⭐

Learn

- Off-by-one errors
- Safe mid calculation

LeetCode

- 852 — Peak Index in a Mountain Array
- 162 — Find Peak Element

🗓️ Day 13 — Binary Search Summary

LeetCode

- 33 — Search in Rotated Sorted Array
- 153 — Find Minimum in Rotated Sorted Array

🗓️ Day 14 — Weekly Revision

Do

- Re-solve 2 binary search problems
- Explain boundary logic aloud ⭐

🔹 WEEK 3 — Two Pointers + Sliding Window

🗓️ Day 15 — Two Pointers Basics

Learn

- Left & right pointer logic

LeetCode

- 167 — Two Sum II (Sorted Array)
- 977 — Squares of a Sorted Array

🗓️ Day 16 — Two Pointers (In-place)

Learn

- Modify array in place

LeetCode

- 283 — Move Zeroes
- 27 — Remove Element

🗓️ Day 17 — Sliding Window (Fixed Size)

Learn

- Window expand & shrink

LeetCode

- 643 — Maximum Average Subarray I
- 1343 — Number of Sub-arrays of Size K

🗓️ Day 18 — Sliding Window Practice

LeetCode

- 1052 — Grumpy Bookstore Owner
- 2461 — Maximum Sum of Distinct Subarrays With Length K

🗓️ Day 19 — In-place vs Extra Space ⭐

Learn

- Space tradeoffs

LeetCode

- 189 — Rotate Array
- 88 — Merge Sorted Array

🗓️ Day 20 — Python Performance Awareness

Learn

- Slicing vs iteration

LeetCode

- 344 — Reverse String
- 125 — Valid Palindrome

🗓 Day 21 — Weekly Revision

Do

- Re-solve sliding window problems
- Dry-run pointer movement ⭐

🔹 WEEK 4 — Sorting + Math

🗓 Day 22 — Sorting Basics

Learn

- sort() vs sorted()

LeetCode

- 912 — Sort an Array
- 976 — Largest Perimeter Triangle

🗓 Day 23 — Custom Sorting

Learn

- key, lambda

LeetCode

- 179 — Largest Number
- 973 — K Closest Points to Origin

🗓 Day 24 — Stable vs Unstable Sort

Learn

- Order preservation

LeetCode

- 242 — Valid Anagram
- 451 — Sort Characters By Frequency

🗓️ Day 25 — Sort vs Heap Decision ⭐

Learn

- When heap is better

LeetCode

- 215 — Kth Largest Element in an Array
- 347 — Top K Frequent Elements

🗓️ Day 26 — Math Basics

Learn

- GCD, LCM, digits

LeetCode

- 1281 — Subtract Product and Sum
- 171 — Excel Sheet Column Number

🗓️ Day 27 — Prime & Modulo

Learn

- Prime check
- Modulo usage

LeetCode

- 204 — Count Primes
- 509 — Fibonacci Number

🗓️ Day 28 — Power & Math Patterns

LeetCode

- 50 — Pow(x, n)
- 231 — Power of Two

🗓️ Day 29 — Full Revision Day

Do

- Re-solve 5 weakest problems
- Focus on **boundaries & indexes**

🗓️ Day 30 — Mock Day ⭐

Do (90 minutes)

- 704 — Binary Search
- 189 — Rotate Array
- 215 — Kth Largest Element

Explain:

- Approach
- Time & space
- Why this approach

🟢 MONTH 3 — 30-DAY DAILY SCHEDULE

Goal: Fast string & sliding window problem solving

🔹 WEEK 1 — String Basics + Two Pointers

🗓️ Day 1 — String Manipulation Basics

Learn

- Indexing
- Traversal
- len, slicing

LeetCode

- 344 — Reverse String
- 1108 — Defanging an IP Address

🗓️ Day 2 — Character Frequency

Learn

- Frequency dict
- Counting logic

LeetCode

- 387 — First Unique Character in a String
- 383 — Ransom Note

🗓️ Day 3 — Palindrome Basics

Learn

- Two pointers on string ⭐

LeetCode

- 125 — Valid Palindrome
- 680 — Valid Palindrome II

🗓️ Day 4 — Anagrams

Learn

- Compare frequency maps

LeetCode

- 242 — Valid Anagram
- 438 — Find All Anagrams in a String

🗓️ Day 5 — Two Pointers (Advanced)

Learn

- Left/right movement
- Skip characters

LeetCode

- 345 — Reverse Vowels of a String
- 917 — Reverse Only Letters

🗓️ Day 6 — Mixed String Practice

LeetCode

- 58 — Length of Last Word
- 67 — Add Binary

🗓️ Day 7 — Weekly Revision

Do

- Re-solve 3 weak string problems
- Explain pointer movement aloud ⭐

🔹 WEEK 2 — Sliding Window (Fixed → Variable)

🗓️ Day 8 — Sliding Window (Fixed Size)

Learn

- Window expand & slide

LeetCode

- 643 — Maximum Average Subarray I
- 1343 — Number of Sub-arrays of Size K

🗓️ Day 9 — Sliding Window (Fixed Practice)

LeetCode

- 1876 — Substrings of Size Three with Distinct Characters
- 1984 — Minimum Difference Between Highest and Lowest of K Scores

---

🗓️ Day 10 — Sliding Window (Variable Size)

Learn

- Expand & shrink logic

LeetCode

- 3 — Longest Substring Without Repeating Characters
- 159 — Longest Substring with At Most Two Distinct Characters

---

🗓️ Day 11 — At Most K Distinct ⭐

Learn

- Frequency map + window

LeetCode

- 340 — Longest Substring with At Most K Distinct Characters
- 904 — Fruit Into Baskets

🗓️ Day 12 — Exactly K Distinct ⭐

Learn

- exactly K = atMost(K) − atMost(K-1)

LeetCode

- 992 — Subarrays with K Different Integers
- 1248 — Count Number of Nice Subarrays

🗓️ Day 13 — Longest vs Shortest Window ⭐

Learn

- Objective difference

LeetCode

- 209 — Minimum Size Subarray Sum
- 1004 — Max Consecutive Ones III

🗓️ Day 14 — Weekly Revision

Do

- Re-solve:
  - 3
  - 904
  - 209
- Focus on **why window shrinks**

🔹 WEEK 3 — Prefix + Hashing Patterns

🗓️ Day 15 — Prefix Sum Basics

Learn

- Running sum logic

LeetCode

- 1480 — Running Sum of 1d Array
- 1732 — Find the Highest Altitude

🗓️ Day 16 — Prefix Sum + HashMap

Learn

- Store prefix counts

LeetCode

- 560 — Subarray Sum Equals K
- 974 — Subarray Sums Divisible by K

🗓️ Day 17 — Count of Subarrays ⭐

Learn

- Frequency of prefix sums

LeetCode

- 930 — Binary Subarrays With Sum
- 1248 — Count Number of Nice Subarrays (revisit)

🗓️ Day 18 — Prefix vs Sliding Window

Learn

- Why sliding window fails with negatives ⭐

LeetCode

- 525 — Contiguous Array
- 523 — Continuous Subarray Sum

🗓️ Day 19 — Hashing + Prefix Practice

LeetCode

- 238 — Product of Array Except Self
- 1371 — Find the Longest Substring Containing Vowels in Even Counts

---

🗓️ Day 20 — Prefix Review Day

Do

- Re-solve 560 or 974
- Explain logic step-by-step aloud ⭐

🗓️ Day 21 — Weekly Revision

Do

- Revise:
    - Sliding window templates
    - Prefix templates

🔹 WEEK 4 — Python Internals + Tools + Mocks

🗓️ Day 22 — String Immutability

Learn

- Why strings are immutable

LeetCode

- 1662 — Check If Two String Arrays are Equivalent
- 541 — Reverse String II

🗓️ Day 23 — += vs join()

Learn

- Performance difference

LeetCode

- 824 — Goat Latin
- 520 — Detect Capital

🗓️ Day 24 — Slicing Cost

Learn

- s[l:r] complexity

LeetCode

- 28 — Find the Index of the First Occurrence in a String
- 796 — Rotate String

🗓️ Day 25 — Counter & defaultdict

Learn

- Cleaner frequency handling

LeetCode

- 451 — Sort Characters By Frequency
- 692 — Top K Frequent Words

🗓️ Day 26 — Mixed Pattern Practice

LeetCode

- 567 — Permutation in String
- 1456 — Maximum Number of Vowels in a Substring of Given Length

---

🗓️ Day 27 — Advanced Mixed Practice

LeetCode

- 76 — Minimum Window Substring
- 424 — Longest Repeating Character Replacement

🗓️ Day 28 — Full Revision

Do

- Re-solve 5 weakest problems
- Identify pattern instantly ⭐

🗓️ Day 29 — Mock Interview Day

Solve in 90 minutes

- 3 — Longest Substring Without Repeating Characters
- 560 — Subarray Sum Equals K
- 76 — Minimum Window Substring

Explain:

- Pattern
- Why it works
- Time & space

🗓️ Day 30 — Reflection & Template Writing ⭐

Do

- Write templates for:
    - Sliding window
    - Prefix + hashmap
- Note mistakes & fixes

🟢 MONTH 4 — Hashing + Optimization Discipline (Days 1–20)

Goal: Brute → optimal mindset

🔹 WEEK 1 — Hashing Basics + Frequency

🗓 Day 1 — HashMap Basics

Learn

- dict operations
- lookup optimization

LeetCode

- 1 — Two Sum
- 217 — Contains Duplicate

🗓 Day 2 — HashSet Usage

Learn

- set vs list
- membership test

LeetCode

- 349 — Intersection of Two Arrays
- 202 — Happy Number

🗓 Day 3 — Frequency Counting

Learn

- count via dict

LeetCode

- 169 — Majority Element
- 387 — First Unique Character in a String

🗓️ Day 4 — Hashing + Arrays

Learn

- Use index + hash

LeetCode

- 448 — Find All Numbers Disappeared in an Array
- 645 — Set Mismatch

🗓️ Day 5 — Hashing + Strings

Learn

- char frequency

LeetCode

- 242 — Valid Anagram
- 205 — Isomorphic Strings

🗓️ Day 6 — Prefix Sum + HashMap

Learn

- cumulative sum logic

LeetCode

- 560 — Subarray Sum Equals K
- 974 — Subarray Sums Divisible by K

🗓️ Day 7 — Weekly Revision

Do

- Re-solve 2 weak hashing problems
- Explain time & space aloud ⭐

🔹 WEEK 2 — Hashing Internals + Traps

🗓️ Day 8 — Why dict is O(1)

Learn

- hashing intuition
- collisions (high-level)

LeetCode

- 136 — Single Number
- 409 — Longest Palindrome

🗓️ Day 9 — Worst-case Awareness

Learn

- when hashing degrades

LeetCode

- 454 — 4Sum II
- 525 — Contiguous Array

🗓️ Day 10 — Sliding Window Failure (Negatives) ⭐

Learn

- why prefix + hash is needed

LeetCode

- 523 — Continuous Subarray Sum
- 930 — Binary Subarrays With Sum

🗓️ Day 11 — Python Hashing Traps

Learn

- mutable keys ❌
- tuple keys ✅
- dict order

LeetCode

- 49 — Group Anagrams
- 451 — Sort Characters By Frequency

🗓 Day 12 — Choosing Correct DS ⭐

Learn

- dict vs set vs list

LeetCode

- 290 — Word Pattern
- 706 — Design HashMap

🗓 Day 13 — Time vs Space Tradeoff ⭐

Learn

- optimize consciously

LeetCode

- 238 — Product of Array Except Self
- 287 — Find the Duplicate Number

🗓 Day 14 — Weekly Revision

Do

- Re-solve 560 or 238
- Explain optimization reasoning ⭐

🔹 WEEK 3 — Optimization + Fast I/O

🗓 Day 15 — Lookup Optimization

Learn

- precompute vs on-the-fly

LeetCode

- 1002 — Find Common Characters
- 884 — Uncommon Words from Two Sentences

🗓 Day 16 — Hashing + Prefix (Counting)

Learn

- count subarrays

LeetCode

- 1248 — Count Number of Nice Subarrays
- 1371 — Find Longest Substring with Even Vowels

🗓 Day 17 — Fast I/O Awareness

Learn

- when fast I/O matters

LeetCode

- 299 — Bulls and Cows
- 500 — Keyboard Row

🗓 Day 18 — Mixed Optimization Practice

LeetCode

- 347 — Top K Frequent Elements
- 692 — Top K Frequent Words

🗓 Day 19 — Full Hashing Revision

Do

- Re-solve 3 toughest hashing problems
- Write brute → optimal flow ⭐

🗓️ Day 20 — Mock Day ⭐

Solve in 90 min

- 1 — Two Sum
- 560 — Subarray Sum Equals K
- 347 — Top K Frequent Elements

Explain:

- Why hashing?
- Time & space
- Alternatives

🟢 MONTH 4.5 — OOP in Python (Days 21–30)

Goal: Clean design + interview clarity

🔹 WEEK 4 — OOP Core + Practice

🗓️ Day 21 — Class & Object

Learn

- class, object
- __init__

Practice

- Create Student class
- Add attributes & methods

🗓️ Day 22 — Instance vs Class Variables

Learn

- static behavior

Practice

- Employee count example

🗓️ Day 23 — Methods & self

Learn

- instance / class / static methods

Practice

- BankAccount with:
    - deposit()
    - withdraw()

🗓️ Day 24 — OOP Principles

Learn

- Encapsulation
- Inheritance
- Polymorphism

Practice

- Shape → Circle / Rectangle

🗓️ Day 25 — Abstraction + ABC

Learn

- abstract base classes

Practice

- Payment system (Card / UPI)

🗓️ Day 26 — Python-Specific OOP

Learn

- super()
- MRO
- multiple inheritance

Practice

- Diamond inheritance example

🗓️ Day 27 — Composition vs Inheritance ⭐

Learn

- HAS-A vs IS-A

Practice

- Library → Book (composition)

🗓️ Day 28 — OOP Internals

Learn

- public / protected / private
- @property
- __eq__, __hash__

Practice

- Object as dict key example

🗓️ Day 29 — Design Practice

Practice

- Library Management System
- Explain design decisions ⭐

🗓️ Day 30 — OOP Mock Interview ⭐

Do

- Explain:
    - Encapsulation vs abstraction
    - Composition vs inheritance
- Code a small class live

🟡 MONTH 5 — 30-DAY DAILY SCHEDULE

Goal: Recursive clarity (controlled)

🔹 WEEK 1 — Recursion Fundamentals

🗓️ Day 1 — Recursion Basics

Learn

- What recursion is
- Call stack
- Base vs recursive case

LeetCode

- 509 — Fibonacci Number
- 231 — Power of Two

🗓️ Day 2 — Recursion Tree

Learn

- How calls branch
- Visualize recursion tree

LeetCode

- 70 — Climbing Stairs
- 1137 — N-th Tribonacci Number

🗓️ Day 3 — Recursion Time Complexity

Learn

- Why recursion becomes exponential
- Stack depth cost

LeetCode

- 344 — Reverse String (recursive)
- 206 — Reverse Linked List (recursive)

🗓️ Day 4 — Recursion vs Iteration ⭐

Learn

- Why iteration is safer in Python

LeetCode

- 50 — Pow(x, n)
- 326 — Power of Three

🗓️ Day 5 — When Recursion FAILS in Python ⭐

Learn

- Recursion limit
- Stack overflow
- TLE risk

LeetCode

- 234 — Palindrome Linked List
- 203 — Remove Linked List Elements

🗓️ Day 6 — Parameter vs Global State

Learn

- Passing state safely
- Avoid globals

LeetCode

- 104 — Maximum Depth of Binary Tree
- 111 — Minimum Depth of Binary Tree

🗓️ Day 7 — Weekly Revision

Do

- Re-solve Fibonacci & tree depth

- Explain recursion tree aloud ⭐

🔹 **WEEK 2 — Backtracking Fundamentals**

🗓️ **Day 8 — Backtracking Concept**

Learn

- Choose → Explore → Un-choose ⭐

LeetCode

- 78 — Subsets
- 401 — Binary Watch

🗓️ **Day 9 — Subsets (Deep)**

Learn

- Include / exclude logic

LeetCode

- 90 — Subsets II
- 1863 — Sum of All Subset XOR Totals

🗓️ **Day 10 — Permutations**

Learn

- Swapping vs visited array

LeetCode

- 46 — Permutations
- 47 — Permutations II

🗓️ **Day 11 — Combinations**

Learn

- Combination vs permutation difference

LeetCode

- 77 — Combinations
- 216 — Combination Sum III

🗓 Day 12 — Backtracking Constraints

Learn

- Pruning
- Avoid duplicates

LeetCode

- 39 — Combination Sum
- 40 — Combination Sum II

🗓 Day 13 — Backtracking State Safety

Learn

- When to copy list
- When to reuse list ⭐

LeetCode

- 131 — Palindrome Partitioning
- 784 — Letter Case Permutation

🗓 Day 14 — Weekly Revision

Do

- Re-solve:
  - 78
  - 46
  - 39
- Focus on un-choose step ⭐

🔹 WEEK 3 — Recursion → DP Bridge

🗓️ Day 15 — Overlapping Subproblems

Learn

- Repeated recursion calls

LeetCode

- 70 — Climbing Stairs (revisit)
- 198 — House Robber

🗓️ Day 16 — Memoization Idea ⭐

Learn

- Cache recursion results

LeetCode

- 746 — Min Cost Climbing Stairs
- 139 — Word Break

🗓️ Day 17 — Memoization vs Plain Recursion

Learn

- Time improvement

LeetCode

- 322 — Coin Change
- 62 — Unique Paths

🗓️ Day 18 — Python Memoization Awareness

Learn

- Dictionary size
- Stack + memory

LeetCode

- 494 — Target Sum

- 337 — House Robber III

🗓️ Day 19 — Iterative vs Recursive DP

Learn

- Why iterative is safer

LeetCode

- 300 — Longest Increasing Subsequence
- 53 — Maximum Subarray

🗓️ Day 20 — Recursion + DP Review

Do

- Re-solve 198 or 746
- Explain DP transition aloud ⭐

🗓️ Day 21 — Weekly Revision

Do

- Write recursion + DP templates
- Note Python pitfalls ⭐

🔹 **WEEK 4 — Mixed Practice + Mocks**

🗓️ Day 22 — Mixed Recursion

LeetCode

- 112 — Path Sum
- 129 — Sum Root to Leaf Numbers

🗓️ Day 23 — Mixed Backtracking

LeetCode

- 17 — Letter Combinations of a Phone Number
- 93 — Restore IP Addresses

🗓️ **Day 24 — Medium Backtracking**

LeetCode

- 79 — Word Search
- 698 — Partition to K Equal Sum Subsets

🗓️ **Day 25 — Recursion Safety Day** ⭐

Do

- Convert one recursion to iteration
- Analyze stack usage

🗓️ **Day 26 — Full Revision**

Do

- Re-solve 5 weakest problems
- Identify pattern instantly

🗓️ **Day 27 — Mock Interview Day** ⭐

Solve in 90 minutes

- 46 — Permutations
- 78 — Subsets
- 198 — House Robber

Explain:

- Recursion tree
- Time & space
- Why DP helps

🗓️ Day 28 — Template Writing

Do

- Write:
    - Recursion template
    - Backtracking template
    - Memoization template

🗓️ Day 29 — Light Practice

LeetCode

- 102 — Binary Tree Level Order Traversal
- 226 — Invert Binary Tree

🗓️ Day 30 — Reflection & Reset ⭐

Do

- Write mistakes list
- Decide:
    - When recursion is OK
    - When iteration is better

🟡 MONTH 6 — 30-DAY DAILY SCHEDULE

Goal: Pointer confidence + clean linked list manipulation

🔹 WEEK 1 — Linked List Basics + Python Reference Model

🗓️ Day 1 — Linked List Basics

Learn

- What is a linked list
- Node structure
- Singly linked list

LeetCode

- 876 — Middle of the Linked List
- 1290 — Convert Binary Number in a Linked List to Integer

🗓️ Day 2 — Python Reference Model ⭐

Learn

- What “pointer” means in Python
- Why node = node.next works
- Losing reference bug

LeetCode

- 203 — Remove Linked List Elements
- 237 — Delete Node in a Linked List

🗓️ Day 3 — Traversal & Edge Cases

Learn

- Empty list
- Single node
- Two nodes

LeetCode

- 206 — Reverse Linked List (iterative)
- 21 — Merge Two Sorted Lists (basic)

🗓️ Day 4 — Doubly Linked List (Concept)

Learn

- prev & next pointers
- Extra space tradeoff

LeetCode

- 707 — Design Linked List
- 1472 — Design Browser History

🗓️ Day 5 — Dummy / Sentinel Node ⭐

Learn

- Why dummy simplifies logic

LeetCode

- 83 — Remove Duplicates from Sorted List
- 82 — Remove Duplicates from Sorted List II

🗓️ Day 6 — Head & Tail Handling

Learn

- Updating head safely
- Maintaining tail

LeetCode

- 19 — Remove Nth Node From End of List
- 328 — Odd Even Linked List

🗓️ Day 7 — Weekly Revision

Do

- Re-solve 203 or 19
- Dry-run pointer movement aloud ⭐

🔹 **WEEK 2 — Fast & Slow Pointers + Core Patterns**

🗓️ **Day 8 — Fast & Slow Pointer Basics**

Learn

- Why fast moves 2x
- Meeting point logic

LeetCode

- 141 — Linked List Cycle
- 142 — Linked List Cycle II

🗓️ **Day 9 — Finding Middle**

Learn

- Even vs odd length

LeetCode

- 876 — Middle of Linked List (revisit)
- 234 — Palindrome Linked List

🗓️ **Day 10 — Reverse Linked List (Deep)**

Learn

- prev / curr / next discipline

LeetCode

- 206 — Reverse Linked List (revisit)
- 92 — Reverse Linked List II

🗓️ **Day 11 — Merge Linked Lists**

Learn

- Dummy node usage

LeetCode

- 21 — Merge Two Sorted Lists (revisit)
- 23 — Merge k Sorted Lists

🗓 Day 12 — Pointer Manipulation Safety

Learn

- Avoid losing references
- Update order

LeetCode

- 143 — Reorder List
- 61 — Rotate List

🗓 Day 13 — Time & Space Reasoning

Learn

- Why most LL ops are O(n)
- Recursion vs iteration

LeetCode

- 147 — Insertion Sort List
- 148 — Sort List

🗓 Day 14 — Weekly Revision

Do

- Re-solve:
    - 141
    - 206
    - 19

- Focus on pointer movement ⭐

🔹 **WEEK 3 — Advanced Patterns + Edge Discipline**

🗓 **Day 15 — Cycle & Loop Problems**

Learn

- Entry point logic

LeetCode

- 142 — Linked List Cycle II (revisit)
- 160 — Intersection of Two Linked Lists

🗓 **Day 16 — Intersection & Overlap**

Learn

- Pointer switching trick

LeetCode

- 160 — Intersection of Two Linked Lists
- 1669 — Merge In Between Linked Lists

🗓 **Day 17 — Partitioning Linked Lists**

Learn

- Split & merge technique

LeetCode

- 86 — Partition List
- 725 — Split Linked List in Parts

🗓 **Day 18 — Modify List In-place**

Learn

- Re-linking nodes safely

LeetCode

- 24 — Swap Nodes in Pairs
- 25 — Reverse Nodes in k-Group

🗓️ Day 19 — Mixed Pattern Practice

LeetCode

- 2 — Add Two Numbers
- 445 — Add Two Numbers II

🗓️ Day 20 — Full Linked List Revision

Do

- Re-solve 5 weakest problems
- Write pointer steps ⭐

🗓️ Day 21 — Weekly Revision

Do

- Explain dummy node usage
- Explain fast–slow logic verbally ⭐

🔹 **WEEK 4 — Mocks + Confidence Building**

🗓️ Day 22 — Medium Practice Day

LeetCode

- 109 — Convert Sorted List to Binary Search Tree
- 430 — Flatten a Multilevel Doubly Linked List

🗓️ Day 23 — Hard Pattern Exposure (Controlled)

LeetCode

- 146 — LRU Cache (Linked List + HashMap)
  (Just understand, don't worry if slow)

🗓 Day 24 — Convert Recursion to Iteration

Do

- Take one LL recursion
- Rewrite iteratively

LeetCode

- 206 — Reverse Linked List (both ways)

🗓 Day 25 — Edge Case Discipline ⭐

Do

- Test:
  - empty list
  - one node
  - two nodes
- Manually simulate

🗓 Day 26 — Full Revision

Do

- Re-solve 3 medium LL problems
- No hints, no notes

🗓 Day 27 — Mock Interview Day ⭐

Solve in 90 minutes

- 19 — Remove Nth Node
- 206 — Reverse Linked List
- 142 — Linked List Cycle II

Explain:

- Pointer movement
- Time & space
- Edge cases

🗓️ Day 28 — Template Writing

Do

- Write templates for:
    - Reverse LL
    - Merge LL
    - Fast–slow pointer
    - Dummy node usage

🗓️ Day 29 — Light Practice

LeetCode

- 83 — Remove Duplicates
- 328 — Odd Even Linked List

🗓️ Day 30 — Reflection & Confidence Check ⭐

Do

- List:
    - Common pointer bugs
    - How to avoid them
- Re-solve one problem confidently

🟡 MONTH 7 — 30-DAY DAILY SCHEDULE

Goal: Flow control + low-level logic confidence

🔹 WEEK 1 — Stack Basics + Core Patterns

🗓️ Day 1 — Stack Fundamentals

Learn

- Stack using Python list
- push / pop
- LIFO behavior

LeetCode

- 20 — Valid Parentheses
- 1047 — Remove All Adjacent Duplicates in String

🗓️ Day 2 — Stack vs Recursion ⭐

Learn

- How recursion uses stack
- When explicit stack is safer

LeetCode

- 844 — Backspace String Compare
- 682 — Baseball Game

🗓️ Day 3 — Stack for Expression-like Problems

Learn

- Matching logic
- State tracking

LeetCode

- 1544 — Make The String Great
- 1614 — Maximum Nesting Depth of the Parentheses

🗓️ Day 4 — Next Greater Element (Basics)

Learn

- Why brute force is slow
- Stack-based thinking

LeetCode

- 496 — Next Greater Element I
- 503 — Next Greater Element II

🗓️ Day 5 — Stack + Array Traversal

Learn

- Index stack

LeetCode

- 739 — Daily Temperatures
- 1475 — Final Prices With a Special Discount in a Shop

🗓️ Day 6 — Stack Cleanup & Edge Cases

Learn

- Empty stack checks
- Final stack state

LeetCode

- 71 — Simplify Path
- 1441 — Build an Array With Stack Operations

🗓️ Day 7 — Weekly Revision

Do

- Re-solve 20 or 739
- Explain why stack works ⭐

🔹 **WEEK 2 — Queue + Deque Patterns**

🗓 Day 8 — Queue Basics

Learn

- FIFO concept
- Why list.pop(0) is slow

LeetCode

- 933 — Number of Recent Calls
- 1700 — Number of Students Unable to Eat Lunch

🗓 Day 9 — collections.deque

Learn

- append / popleft O(1)

LeetCode

- 225 — Implement Stack using Queues
- 232 — Implement Queue using Stacks

🗓 Day 10 — Sliding Window Maximum

Learn

- Why deque is needed

LeetCode

- 239 — Sliding Window Maximum
- 1438 — Longest Continuous Subarray With Absolute Diff ≤ Limit

---

🗓 Day 11 — Queue vs Stack Decision

Learn

- When to use which

LeetCode

- 649 — Dota2 Senate
- 2073 — Time Needed to Buy Tickets

🗓️ **Day 12 — Circular Queue Concept**

Learn

- Wrap-around idea

LeetCode

- 622 — Design Circular Queue
- 641 — Design Circular Deque

🗓️ **Day 13 — Mixed Queue Practice**

LeetCode

- 346 — Moving Average from Data Stream
- 950 — Reveal Cards In Increasing Order

🗓️ **Day 14 — Weekly Revision**

Do

- Re-solve 239
- Dry-run deque operations ⭐

🔹 **WEEK 3 — Monotonic Stack (🔥 CRITICAL)**

🗓️ **Day 15 — Monotonic Stack Basics**

Learn

- Increasing vs decreasing stack
- Why total ops = O(n)

LeetCode

- 739 — Daily Temperatures (revisit)

- 901 — Online Stock Span

🗓 Day 16 — Monotonic Stack (Advanced)

Learn

- Previous greater / smaller element

LeetCode

- 84 — Largest Rectangle in Histogram
- 85 — Maximal Rectangle *(conceptual)*

🗓 Day 17 — Monotonic Stack Variants

Learn

- Next smaller element

LeetCode

- 907 — Sum of Subarray Minimums
- 1019 — Next Greater Node In Linked List

🗓 Day 18 — Monotonic Stack + Array

LeetCode

- 503 — Next Greater Element II (revisit)
- 1475 — Final Prices (revisit)

🗓 Day 19 — When Monotonic Stack Fails

Learn

- Recognize applicability

LeetCode

- 456 — 132 Pattern
- 316 — Remove Duplicate Letters

🗓️ Day 20 — Full Stack Revision

Do

- Re-solve 739 or 84
- Explain monotonic logic ⭐

🗓️ Day 21 — Weekly Revision

Do

- Write monotonic stack template
- Note mistakes ⭐

🔹 WEEK 4 — Bit Manipulation + Mocks

🗓️ Day 22 — Bit Basics (AND, OR, XOR)

Learn

- Truth tables
- XOR intuition

LeetCode

- 136 — Single Number
- 389 — Find the Difference

🗓️ Day 23 — XOR Tricks

Learn

- a^a = 0
- a^0 = a

LeetCode

- 260 — Single Number III
- 268 — Missing Number

🗓️ Day 24 — Power of Two

Learn

- n & (n-1) trick
- Python infinite bits

LeetCode

- 231 — Power of Two
- 342 — Power of Four

🗓️ Day 25 — Bit Manipulation Practice

LeetCode

- 191 — Number of 1 Bits
- 338 — Counting Bits

🗓️ Day 26 — Mixed Low-level Practice

LeetCode

- 405 — Convert a Number to Hexadecimal
- 461 — Hamming Distance

🗓️ Day 27 — Full Revision

Do

- Re-solve:
  - 20
  - 239
  - 136
- Identify pattern instantly ⭐

🗓️ Day 28 — Mock Interview Day ⭐

Solve in 90 minutes

- 739 — Daily Temperatures
- 239 — Sliding Window Maximum
- 136 — Single Number

Explain:

- Pattern
- Time & space
- Why this DS

🗓️ Day 29 — Template Writing

Do

- Write templates for:
    - Stack
    - Monotonic stack
    - Deque sliding window
    - XOR usage

🗓️ Day 30 — Reflection & Confidence Check ⭐

Do

- List:
    - When to use stack
    - When to use queue
    - When bit tricks apply
- Re-solve one medium confidently

🟡 MONTH 8 — 30-DAY DAILY SCHEDULE

Goal: Tree + heap intuition (interview-ready)

🔹 WEEK 1 — Tree Basics + Traversals

🗓️ Day 1 — Binary Tree Basics

Learn

- Node, left, right
- Height vs depth

LeetCode

- 104 — Maximum Depth of Binary Tree
- 111 — Minimum Depth of Binary Tree

🗓️ Day 2 — DFS Traversals (Recursive)

Learn

- Preorder, Inorder, Postorder

LeetCode

- 144 — Binary Tree Preorder Traversal
- 94 — Binary Tree Inorder Traversal

🗓️ Day 3 — DFS Traversals (Postorder)

Learn

- When postorder is useful

LeetCode

- 145 — Binary Tree Postorder Traversal
- 226 — Invert Binary Tree

🗓️ Day 4 — BFS (Level Order)

Learn

- Queue-based traversal

LeetCode

- 102 — Binary Tree Level Order Traversal
- 107 — Binary Tree Level Order Traversal II

🗓 Day 5 — Recursive vs Iterative ⭐

Learn

- Stack vs recursion
- Python recursion depth risk

LeetCode

- 144 — Preorder Traversal (iterative)
- 94 — Inorder Traversal (iterative)

🗓 Day 6 — Tree Properties

Learn

- Balanced vs unbalanced
- Complete vs full

LeetCode

- 110 — Balanced Binary Tree
- 101 — Symmetric Tree

🗓 Day 7 — Weekly Revision

Do

- Re-solve 104, 102
- Explain DFS vs BFS aloud ⭐

🔹 WEEK 2 — BST + Tree Patterns

🗓️ Day 8 — Binary Search Tree (BST)

Learn

- BST invariant (left < root < right)

LeetCode

- 700 — Search in a Binary Search Tree
- 701 — Insert into a Binary Search Tree

🗓️ Day 9 — BST Validation

Learn

- Range-based checking

LeetCode

- 98 — Validate Binary Search Tree
- 530 — Minimum Absolute Difference in BST

🗓️ Day 10 — BST Traversal Logic

Learn

- Inorder gives sorted order

LeetCode

- 230 — Kth Smallest Element in a BST
- 173 — Binary Search Tree Iterator

🗓️ Day 11 — Lowest Common Ancestor (LCA)

Learn

- Bottom-up recursion

LeetCode

- 236 — Lowest Common Ancestor of a Binary Tree
- 235 — Lowest Common Ancestor of a BST

🗓️ Day 12 — Path-Based Problems

Learn

- Root-to-leaf logic

LeetCode

- 112 — Path Sum
- 129 — Sum Root to Leaf Numbers

🗓️ Day 13 — Tree Diameter & Depth

Learn

- Postorder usage

LeetCode

- 543 — Diameter of Binary Tree
- 687 — Longest Univalue Path

🗓️ Day 14 — Weekly Revision

Do

- Re-solve 236 or 543
- Explain recursion flow ⭐

🔹 WEEK 3 — Tree as Graph (🔥 IMPORTANT)

🗓️ Day 15 — Tree as Graph Concept

Learn

- Parent mapping
- Visited set

LeetCode

- 863 — All Nodes Distance K in Binary Tree
- 742 — Closest Leaf in a Binary Tree

🗓️ Day 16 — BFS from Target

Learn

- BFS with visited logic ⭐

LeetCode

- 863 — All Nodes Distance K (revisit)
- 1110 — Delete Nodes And Return Forest

🗓️ Day 17 — Convert Tree to Graph

Learn

- Undirected edges

LeetCode

- 1372 — Longest ZigZag Path in a Binary Tree
- 1161 — Maximum Level Sum of a Binary Tree

🗓️ Day 18 — DFS vs BFS Decision ⭐

Learn

- Shortest path vs traversal

LeetCode

- 199 — Binary Tree Right Side View
- 513 — Find Bottom Left Tree Value

🗓️ Day 19 — Iterative Tree Traversal

Learn

- Explicit stack usage

LeetCode

- 173 — BST Iterator (revisit)
- 331 — Verify Preorder Serialization of a Binary Tree

🗓️ Day 20 — Full Tree Revision

Do

- Re-solve 102, 236, 863
- Identify pattern instantly ⭐

🗓️ Day 21 — Weekly Revision

Do

- Draw trees manually
- Explain approach verbally ⭐

🔹 WEEK 4 — Heaps + Selection Problems

🗓️ Day 22 — Heap Basics

Learn

- Min heap vs max heap
- Array representation

LeetCode

- 703 — Kth Largest Element in a Stream
- 1046 — Last Stone Weight

🗓️ Day 23 — heapq in Python

Learn

- push, pop, heapify

LeetCode

- 215 — Kth Largest Element in an Array
- 973 — K Closest Points to Origin

🗓️ Day 24 — Top-K Problems

Learn

- Heap vs sort decision ⭐

LeetCode

- 347 — Top K Frequent Elements
- 692 — Top K Frequent Words

🗓️ Day 25 — Heap + Greedy

Learn

- Priority-based selection

LeetCode

- 295 — Find Median from Data Stream
- 621 — Task Scheduler

🗓️ Day 26 — Heap Internals Awareness

Learn

- Why operations are O(log n)

LeetCode

- 506 — Relative Ranks
- 1337 — The K Weakest Rows in a Matrix

🗓️ Day 27 — Mixed Tree + Heap Practice

LeetCode

- 515 — Find Largest Value in Each Tree Row
- 1005 — Maximize Sum Of Array After K Negations

🗓️ Day 28 — Full Revision

Do

- Re-solve:
    - 236
    - 863
    - 215
- Explain DS choice ⭐

🗓 Day 29 — Mock Interview Day ⭐

Solve in 90 minutes

- 102 — Level Order Traversal
- 236 — LCA
- 215 — Kth Largest Element

Explain:

- DFS vs BFS
- Heap vs sort
- Time & space

🗓 Day 30 — Template Writing & Reflection ⭐

Do

- Write templates for:
    - DFS
    - BFS
    - LCA
    - Heap Top-K
- List common mistakes

🔵 MONTH 9 — 30-DAY DAILY SCHEDULE

Goal: Optimal decision making (Greedy mastery)

🔹 WEEK 1 — Greedy Fundamentals (Thinking Discipline)

🗓 Day 1 — What is Greedy?

Learn

- Greedy choice property ⭐
- Optimal substructure ⭐

LeetCode

- 455 — Assign Cookies
- 860 — Lemonade Change

🗓 Day 2 — When Greedy FAILS

Learn

- Counterexamples
- Why greedy is not universal

LeetCode

- 55 — Jump Game
- 45 — Jump Game II

🗓 Day 3 — Greedy for Optimization

Learn

- Local vs global optimum

LeetCode

- 121 — Best Time to Buy and Sell Stock
- 122 — Best Time to Buy and Sell Stock II

🗓️ Day 4 — Greedy with Sorting

Learn

- Why sorting enables greedy

LeetCode

- 1710 — Maximum Units on a Truck
- 2410 — Maximum Matching of Players With Trainers

🗓️ Day 5 — Greedy Boundaries ⭐

Learn

- Inclusive vs exclusive logic

LeetCode

- 605 — Can Place Flowers
- 605 — Revisit with edge cases

🗓️ Day 6 — Mixed Greedy Practice

LeetCode

- 1403 — Minimum Subsequence in Non-Increasing Order
- 409 — Longest Palindrome

🗓️ Day 7 — Weekly Revision

Do

- Re-solve 455 or 860
- Explain greedy proof aloud ⭐

🔹 WEEK 2 — Interval Problems (CORE INTERVIEW AREA)

🗓️ Day 8 — Interval Representation

Learn

- [start, end] meaning
- Sorting by start vs end ⭐

LeetCode

- 228 — Summary Ranges
- 252 — Meeting Rooms

🗓 Day 9 — Interval Merging

Learn

- Overlap detection

LeetCode

- 56 — Merge Intervals
- 57 — Insert Interval

🗓 Day 10 — Non-overlapping Intervals

Learn

- Remove minimum intervals

LeetCode

- 435 — Non-overlapping Intervals
- 452 — Minimum Number of Arrows to Burst Balloons

🗓 Day 11 — Boundary Handling ⭐

Learn

- Touching intervals
- Closed vs open ranges

LeetCode

- 763 — Partition Labels
- 986 — Interval List Intersections

🗓️ Day 12 — Interval + Greedy Proof

Learn

- Why sorting by end works

LeetCode

- 1029 — Two City Scheduling
- 1221 — Split a String in Balanced Strings

🗓️ Day 13 — Interval Revision

Do

- Re-solve 56 & 435
- Explain boundary logic ⭐

🗓️ Day 14 — Weekly Revision

Do

- Identify:
    - start-sort problems
    - end-sort problems

🔹 WEEK 3 — Scheduling + Greedy + Heap

🗓️ Day 15 — Scheduling Basics

Learn

- Activity selection idea

LeetCode

- 252 — Meeting Rooms (revisit)
- 253 — Meeting Rooms II

🗓️ Day 16 — Greedy + Heap (🔥 IMPORTANT)

Learn

- Min heap for earliest end time

LeetCode

- 253 — Meeting Rooms II (revisit)
- 1094 — Car Pooling

🗓 Day 17 — Deadline-based Scheduling

Learn

- Replace worst choice

LeetCode

- 630 — Course Schedule III
- 1353 — Maximum Number of Events That Can Be Attended

🗓 Day 18 — Greedy with Replacement

Learn

- Remove largest / smallest

LeetCode

- 1833 — Maximum Ice Cream Bars
- 2383 — Minimum Hours of Training

🗓 Day 19 — Greedy + Heap Practice

LeetCode

- 621 — Task Scheduler
- 767 — Reorganize String

🗓 Day 20 — Full Scheduling Revision

Do

- Re-solve 253 or 630

- Explain heap usage ⭐

🗓 Day 21 — Weekly Revision

Do

- Write greedy decision checklist ⭐

🔹 **WEEK 4 — Mixed Practice + Mocks**

🗓 Day 22 — Mixed Greedy Problems

LeetCode

- 406 — Queue Reconstruction by Height
- 665 — Non-decreasing Array

🗓 Day 23 — Greedy vs DP Thinking

Learn

- Why DP is overkill here

LeetCode

- 376 — Wiggle Subsequence
- 1323 — Maximum 69 Number

🗓 Day 24 — Hard Greedy (Controlled)

LeetCode

- 134 — Gas Station
- 968 — Binary Tree Cameras *(intuition only)*

🗓 Day 25 — Edge Case Discipline ⭐

Do

- Test:

    - empty input
    - single interval
    - fully overlapping

🗓️ Day 26 — Full Revision

Do

- Re-solve 5 weakest greedy problems
- Focus on proof, not code

🗓️ Day 27 — Mock Interview Day ⭐

Solve in 90 minutes

- 56 — Merge Intervals
- 253 — Meeting Rooms II
- 455 — Assign Cookies

Explain:

- Greedy choice
- Sorting key
- Time & space

🗓️ Day 28 — Template Writing

Do

- Write templates for:
    - Greedy proof
    - Interval merging
    - Scheduling with heap

🗓️ Day 29 — Light Practice

LeetCode

- 122 — Best Time to Buy and Sell Stock II
- 605 — Can Place Flowers

🗓️ Day 30 — Reflection & Confidence Check ⭐

Do

- Answer:
    - When greedy works
    - When greedy fails
- Re-solve one medium confidently

🔵 MONTH 10 — 30-DAY DAILY SCHEDULE

Goal: Complex relationship handling (Graph confidence)

🔹 WEEK 1 — Graph Fundamentals + Traversals

🗓️ Day 1 — Graph Basics

Learn

- Graph terminology
- Adjacency list vs adjacency matrix ⭐
- Space/time tradeoffs

LeetCode

- 1971 — Find if Path Exists in Graph
- 1791 — Find Center of Star Graph

🗓️ Day 2 — Graph Representation

Learn

- Build adjacency list (dict of lists)
- Directed vs undirected ⭐

LeetCode

- 997 — Find the Town Judge
- 1436 — Destination City

🗓️ Day 3 — BFS Basics

Learn

- Queue-based traversal
- Visited set

LeetCode

- 733 — Flood Fill

- 200 — Number of Islands

🗓 Day 4 — DFS Basics

Learn

- Recursive DFS
- Stack-based DFS

LeetCode

- 695 — Max Area of Island
- 463 — Island Perimeter

🗓 Day 5 — BFS vs DFS Decisions ⭐

Learn

- Shortest path vs exploration
- Python recursion safety

LeetCode

- 994 — Rotting Oranges
- 1091 — Shortest Path in Binary Matrix

🗓 Day 6 — Connected Components

Learn

- Count components

LeetCode

- 547 — Number of Provinces
- 323 — Number of Connected Components in an Undirected Graph

---

🗓 Day 7 — Weekly Revision

Do

- Re-solve 200 or 994

- Explain BFS vs DFS choice aloud ⭐

🔹 **WEEK 2 — Cycle Detection + DAG**

🗓 **Day 8 — Cycle Detection (Undirected)**

Learn

- Parent tracking
- DFS logic

LeetCode

- 684 — Redundant Connection
- 261 — Graph Valid Tree

🗓 **Day 9 — Cycle Detection (Directed)**

Learn

- Recursion stack
- Visited states ⭐

LeetCode

- 207 — Course Schedule
- 802 — Find Eventual Safe States

🗓 **Day 10 — DAG Concept**

Learn

- What is DAG
- Why cycles break DAG

LeetCode

- 207 — Course Schedule (revisit)
- 210 — Course Schedule II

🗓️ Day 11 — Topological Sort (Kahn's Algorithm)

Learn

- Indegree
- Queue processing ⭐

LeetCode

- 210 — Course Schedule II (revisit)
- 1203 — Sort Items by Groups Respecting Dependencies *(conceptual)*

🗓️ Day 12 — Topo Sort via DFS (Awareness)

Learn

- Postorder stack
- Reverse result

LeetCode

- 207 — Course Schedule (DFS approach)
- 1192 — Critical Connections in a Network *(intuition only)*

🗓️ Day 13 — When Topo Sort Fails ⭐

Learn

- Cycle detection using topo

LeetCode

- 802 — Find Eventual Safe States (revisit)
- 1557 — Minimum Number of Vertices to Reach All Nodes

🗓️ Day 14 — Weekly Revision

Do

- Re-solve 207 or 210
- Explain DAG logic aloud ⭐

🔹 **WEEK 3 — Union-Find (DSU) Mastery**

🗓 **Day 15 — Union-Find Basics**

Learn

- Parent array
- Find & union

LeetCode

- 547 — Number of Provinces (DSU version)
- 684 — Redundant Connection (DSU version)

🗓 **Day 16 — Path Compression**

Learn

- Flatten tree
- Amortized complexity ⭐

LeetCode

- 1319 — Number of Operations to Make Network Connected
- 721 — Accounts Merge

🗓 **Day 17 — Union by Rank**

Learn

- Attach smaller tree to larger

LeetCode

- 947 — Most Stones Removed with Same Row or Column
- 990 — Satisfiability of Equality Equations

🗓 **Day 18 — DSU vs DFS/BFS** ⭐

Learn

- When DSU is better
- When DSU fails

LeetCode

- 399 — Evaluate Division (DFS better)
- 1061 — Lexicographically Smallest Equivalent String

🗓 Day 19 — DSU on Grids

Learn

- Mapping 2D → 1D

LeetCode

- 959 — Regions Cut By Slashes
- 305 — Number of Islands II *(conceptual)*

🗓 Day 20 — Full DSU Revision

Do

- Re-solve 721 or 947
- Explain DSU decision aloud ⭐

🗓 Day 21 — Weekly Revision

Do

- Write DSU template
- Note common mistakes ⭐

🔹 WEEK 4 — Mixed Graph Practice + Mocks

🗓 Day 22 — Graph Traversal Practice

LeetCode

- 130 — Surrounded Regions

- 417 — Pacific Atlantic Water Flow

🗓️ Day 23 — BFS on Graphs

Learn

- Multi-source BFS

LeetCode

- 542 — 01 Matrix
- 286 — Walls and Gates

🗓️ Day 24 — DFS on Graphs

LeetCode

- 329 — Longest Increasing Path in a Matrix
- 133 — Clone Graph

🗓️ Day 25 — Edge Case Discipline ⭐

Do

- Empty graph
- Single node
- Disconnected graph

🗓️ Day 26 — Full Revision

Do

- Re-solve 3 medium graph problems
- No notes, no hints

🗓️ Day 27 — Mock Interview Day ⭐

Solve in 90 minutes

- 200 — Number of Islands

- 207 — Course Schedule
- 721 — Accounts Merge

Explain:

- Graph type
- Traversal choice
- Time & space

🗓 Day 28 — Template Writing

Do

- Write templates for:
    - BFS
    - DFS
    - Topological sort
    - Union-Find

🗓 Day 29 — Light Practice

LeetCode

- 997 — Find the Town Judge
- 1971 — Find if Path Exists in Graph

🗓 Day 30 — Reflection & Confidence Check ⭐

Do

- Answer:
    - BFS vs DFS
    - DSU vs traversal
- Re-solve one medium confidently

🔴 MONTH 11 — 30-DAY DAILY SCHEDULE

Goal: DP-safe (not DP-panic)

🔹 WEEK 1 — DP Mindset + 1D DP (FOUNDATION)

🗓️ Day 1 — When to Use DP

Learn

- Overlapping subproblems
- Optimal substructure
- DP vs recursion vs greedy ⭐

LeetCode

- 70 — Climbing Stairs
- 509 — Fibonacci Number

🗓️ Day 2 — DP Framework

Learn

- State
- Transition
- Base case

LeetCode

- 746 — Min Cost Climbing Stairs
- 1137 — N-th Tribonacci Number

🗓️ Day 3 — Memoization

Learn

- Top-down DP
- Cache repeated work

LeetCode

- 198 — House Robber
- 740 — Delete and Earn

🗓️ Day 4 — Tabulation

Learn

- Bottom-up DP
- Table filling order ⭐

LeetCode

- 198 — House Robber (tabulation)
- 213 — House Robber II

🗓️ Day 5 — Space Optimization ⭐

Learn

- Why full DP table isn't needed

LeetCode

- 70 — Climbing Stairs (O(1) space)
- 746 — Min Cost Climbing Stairs (optimized)

🗓️ Day 6 — DP Time & Space Complexity

Learn

- State count
- Transition cost

LeetCode

- 53 — Maximum Subarray
- 152 — Maximum Product Subarray

🗓️ Day 7 — Weekly Revision

Do

- Re-solve 198 or 746
- Write DP state & transition in words ⭐

🔹 **WEEK 2 — 2D DP + Grid DP**

🗓️ **Day 8 — 2D DP Basics**

Learn

- Row–column DP table

LeetCode

- 62 — Unique Paths
- 63 — Unique Paths II

🗓️ **Day 9 — Grid DP (Path Problems)**

Learn

- From top-left to bottom-right

LeetCode

- 64 — Minimum Path Sum
- 120 — Triangle

🗓️ **Day 10 — Grid DP Variants**

Learn

- Multiple directions

LeetCode

- 931 — Minimum Falling Path Sum
- 221 — Maximal Square

🗓️ **Day 11 — DP with Obstacles & Constraints**

Learn

- Blocked cells

LeetCode

- 63 — Unique Paths II (revisit)
- 174 — Dungeon Game *(intuition first)*

🗓 Day 12 — State Explosion Awareness ⭐

Learn

- Why DP blows up
- How to reduce state

LeetCode

- 416 — Partition Equal Subset Sum
- 494 — Target Sum

🗓 Day 13 — Grid DP Revision

Do

- Re-solve 62 or 64
- Explain transition order aloud ⭐

🗓 Day 14 — Weekly Revision

Do

- Write:
    - 1D DP template
    - 2D DP template

🔹 WEEK 3 — Knapsack + LIS (CORE INTERVIEW)

🗓 Day 15 — 0/1 Knapsack Concept

Learn

- Pick / not pick logic

LeetCode

- 416 — Partition Equal Subset Sum (revisit)
- 1049 — Last Stone Weight II

🗓 Day 16 — Knapsack Variants

Learn

- Capacity-based DP

LeetCode

- 322 — Coin Change
- 518 — Coin Change II

🗓 Day 17 — Unbounded Knapsack

Learn

- Reuse items

LeetCode

- 377 — Combination Sum IV
- 139 — Word Break

🗓 Day 18 — LIS ($O(n^2)$) ⭐

Learn

- dp[i] meaning

LeetCode

- 300 — Longest Increasing Subsequence
- 673 — Number of Longest Increasing Subsequence

🗓 Day 19 — LIS Optimization (Awareness)

Learn

- Binary search optimization (idea only)

LeetCode

- 300 — LIS (optimized idea)
- 354 — Russian Doll Envelopes *(intuition)*

🗓️ Day 20 — Knapsack + LIS Revision

Do

- Re-solve 322 or 300
- Write state & transition ⭐

🗓️ Day 21 — Weekly Revision

Do

- Explain knapsack in plain English
- Explain LIS logic aloud ⭐

🔹 **WEEK 4 — DP vs Greedy + Mocks**

🗓️ Day 22 — DP vs Greedy ⭐

Learn

- Why greedy fails sometimes

LeetCode

- 55 — Jump Game
- 45 — Jump Game II

🗓️ Day 23 — DP on Strings

Learn

- Compare prefixes

LeetCode

- 1143 — Longest Common Subsequence
- 583 — Delete Operation for Two Strings

🗓️ **Day 24 — DP on Subsequences**

Learn

- Matching logic

LeetCode

- 516 — Longest Palindromic Subsequence
- 647 — Palindromic Substrings

🗓️ **Day 25 — Iterative DP Safety** ⭐

Learn

- Why recursion TLEs in Python

LeetCode

- 322 — Coin Change (iterative only)
- 198 — House Robber (revisit)

🗓️ **Day 26 — Full DP Revision**

Do

- Re-solve 3 medium DP problems
- No notes

🗓️ **Day 27 — Mock Interview Day** ⭐

Solve in 90 minutes

- 198 — House Robber
- 62 — Unique Paths
- 322 — Coin Change

Explain:

- State
- Transition
- Complexity

🗓️ Day 28 — DP Template Writing ⭐

Do

- Write templates for:
    - 1D DP
    - 2D DP
    - Knapsack
    - LIS

🗓️ Day 29 — Light Practice

LeetCode

- 338 — Counting Bits
- 91 — Decode Ways

🗓️ Day 30 — Reflection & Confidence Check ⭐

Do

- Answer:
    - When DP is needed
    - How to define state
- Re-solve one DP problem confidently

🟢 MONTH 12 — 30-DAY DAILY SCHEDULE

Goal: Offer-ready (DSA + OOP + communication)

🔹 WEEK 1 — Pattern-Wise Revision (CORE)

🗓️ Day 1 — Arrays + Two Pointers

Revise

- Traversal
- Two pointers
- In-place logic

LeetCode

- 189 — Rotate Array
- 167 — Two Sum II

Focus

- Boundary handling
- Time & space explanation ⭐

🗓️ Day 2 — Sliding Window

Revise

- Fixed vs variable window
- Expand / shrink logic

LeetCode

- 3 — Longest Substring Without Repeating Characters
- 209 — Minimum Size Subarray Sum

🗓️ Day 3 — Prefix + Hashing

Revise

- Prefix sum + hashmap

- Why sliding window fails (negatives)

LeetCode

- 560 — Subarray Sum Equals K
- 525 — Contiguous Array

🗓 Day 4 — Hashing Patterns

Revise

- Frequency counting
- Lookup optimization

LeetCode

- 1 — Two Sum
- 49 — Group Anagrams

🗓 Day 5 — Stack + Queue

Revise

- Stack patterns
- Queue / deque usage

LeetCode

- 20 — Valid Parentheses
- 239 — Sliding Window Maximum

🗓 Day 6 — Linked List

Revise

- Dummy node
- Fast & slow pointers

LeetCode

- 206 — Reverse Linked List
- 19 — Remove Nth Node From End

🗓️ **Day 7 — Weekly Revision**

Do

- Re-solve 2 weakest problems
- Update **mistake log** ⭐

🔹 **WEEK 2 — Trees, Graphs, DP (CONFIDENCE BUILDING)**

🗓️ **Day 8 — Trees (DFS / BFS)**

Revise

- Traversals
- BFS vs DFS decision

LeetCode

- 102 — Level Order Traversal
- 236 — LCA of Binary Tree

🗓️ **Day 9 — Tree as Graph**

Revise

- Parent mapping
- BFS from target

LeetCode

- 863 — All Nodes Distance K
- 199 — Right Side View

🗓️ **Day 10 — Heaps**

Revise

- Heap vs sort decisions ⭐

LeetCode

- 215 — Kth Largest Element
- 347 — Top K Frequent Elements

🗓️ Day 11 — Graphs

Revise

- BFS / DFS
- Cycle detection
- Topo sort

LeetCode

- 200 — Number of Islands
- 207 — Course Schedule

🗓️ Day 12 — Union-Find

Revise

- DSU vs DFS decision ⭐

LeetCode

- 721 — Accounts Merge
- 684 — Redundant Connection

🗓️ Day 13 — Dynamic Programming

Revise

- State → transition → base
- Iterative DP preference ⭐

LeetCode

- 198 — House Robber
- 62 — Unique Paths

🗓️ Day 14 — Weekly Revision

Do

- Re-solve 1 DP + 1 graph problem
- Explain approach aloud ⭐

🔹 WEEK 3 — Timed Practice + Mock Interviews

🗓️ Day 15 — Timed Practice (Easy–Medium)

Do

- 3 problems in 60 minutes

Suggested:

- 121 — Best Time to Buy and Sell Stock
- 20 — Valid Parentheses
- 704 — Binary Search

🗓️ Day 16 — Timed Practice (Medium)

Do

- 2 problems in 60 minutes

Suggested:

- 560 — Subarray Sum Equals K
- 239 — Sliding Window Maximum

🗓️ Day 17 — Mock Interview (DSA)

Do

- Solve 2 problems
- Think aloud ⭐
- Explain tradeoffs

Suggested:

- 236 — LCA
- 215 — Kth Largest

🗓 Day 18 — Mock Interview (OOP)

Practice

- Explain:
    - Encapsulation vs abstraction
    - Composition vs inheritance ⭐

Code

- Simple BankAccount or Student class

🗓 Day 19 — Debugging Discipline

Practice

- Edge cases
- Dry runs
- Assertions
- Failure explanation ⭐

LeetCode

- 33 — Search in Rotated Sorted Array

🗓 Day 20 — Full Mock Day ⭐

90 minutes

- 1 array/string
- 1 tree/graph
- 1 DP

Example:

- 3 — Longest Substring
- 102 — Level Order

- 198 — House Robber

🗓️ **Day 21 — Reflection**

Do

- Update mistake log
- Identify top 5 weak patterns ⭐

🔹 **WEEK 4 — Interview Polish + Bonus Prep**

🗓️ **Day 22 — Clean Coding Practice**

Focus

- Variable naming
- Helper functions
- Early returns

LeetCode

- 704 — Binary Search
- 206 — Reverse Linked List

🗓️ **Day 23 — Handling Interruptions** ⭐

Practice

- Pause → clarify → continue
- Answer “what if” questions

LeetCode

- 56 — Merge Intervals

🗓️ **Day 24 — Python → Java (Optional)**

Do

- Solve 2 known problems in Java

Suggested:

- 1 — Two Sum
- 20 — Valid Parentheses

🗓️ Day 25 — LRU Cache (Bonus)

Learn

- HashMap + Doubly Linked List

LeetCode

- 146 — LRU Cache *(understand well, code if possible)*

🗓️ Day 26 — Simple Class Design

Practice

- Design:
    - Library
    - Parking lot (basic)

Explain decisions ⭐

🗓️ Day 27 — Final Mock Interview ⭐

Do

- Full interview simulation
- 45 min coding
- 15 min explanation

🗓️ Day 28 — Light Revision

Do

- Re-solve 2 favorite problems
- Build confidence

🗓️ Day 29 — Resume + Story Prep

Prepare

- Project explanation
- DSA learning story
- Strengths & weaknesses ⭐

🗓️ Day 30 — RESET DAY 🧠

Do

- Light revision only
- Sleep well
- Confidence check