Perfect 👌 thanks for clarifying. Yes — **2 hours daily for 3 months** is absolutely enough to reach **basic-to-medium DSA interview level** (especially as a fresher). You don't need to master hard-level problems for most entry-level jobs — **clarity in basics + ability to solve medium problems** is what interviewers test.

Here's a **90-day (3-month) DSA roadmap** for you, assuming **2 hrs/day**:

🗓️ Month 1: Foundations (Basics + Easy Problems)

Goal: Get comfortable with syntax, patterns, and problem-solving in Python.

- Week 1:
    - Time & Space Complexity basics
    - Arrays (traversal, min/max, reverse, rotation, prefix sum)
    - 15–20 problems
- Week 2:
    - Strings (palindrome, anagram, substring search, frequency count)
    - Hashing basics (dict, set in Python)
    - 15–20 problems
- Week 3:
    - Recursion basics (factorial, Fibonacci, sum of array, subsets)
    - Sorting (Bubble, Selection, Insertion, Merge, Quick)
    - 15–20 problems
- Week 4:
    - Searching (binary search, first/last occurrence, sqrt, peak element)
    - Practice mixed problems from arrays & strings
    - 20+ problems

✅ By end of Month 1 → You'll be strong in **arrays, strings, recursion, sorting, searching**.

🗓️ Month 2: Data Structures

Goal: Start applying DSA to more structured problems.

- Week 5:
    - Linked List (SLL, DLL, reverse, middle, cycle detection)
    - 10–15 problems
- Week 6:
    - Stacks (using list/collections.deque)
    - Queues (circular queue, deque, priority queue)
    - 15 problems
- Week 7:
    - Trees (binary tree basics: traversal, height, diameter, leaf nodes)
    - Binary Search Tree (insertion, search, inorder traversal)
    - 15 problems
- Week 8:
    - Hashing (map, set, frequency, 2-sum, subarray sum problems)
    - Mix practice (linked list + stack + queue)
    - 15–20 problems

✅ By end of Month 2 → You'll know **linked list, stack, queue, trees, hashing**.

🗓️ **Month 3: Intermediate & Interview Prep**

Goal: Medium-level problem-solving + confidence for interviews.

- Week 9:
    - Recursion + Backtracking (n-queen, rat in maze, permutations, subsets)
    - 10–12 problems
- Week 10:
    - Graph basics (BFS, DFS, adjacency list, shortest path BFS)
    - 10–12 problems
- Week 11:
    - Dynamic Programming (Fibonacci, knapsack, coin change, LIS basics)
    - 10–12 problems

- Week 12:
    - Mock interview sets (arrays, strings, linked lists, trees, recursion)
    - Revise formulas + practice 30–40 mixed problems

✅ By end of Month 3 → You'll be **interview ready** for fresher roles.

📊 **Problem Count (3 months target)**

- Easy: ~150 problems
- Medium: ~80 problems
  ➡️ Total: ~230–250 problems (that's more than enough for fresher-level interviews).

⚡ **Tips for Success**

- Use **LeetCode + GeeksforGeeks + Striver's A2Z DSA sheet** for problem sets.
- Track your progress (e.g., 2 problems/day minimum).
- Focus on **patterns** → most interview problems are variations of common patterns.

👉 So yes, **2 hrs/day × 3 months = enough to crack DSA-based interviews** as a fresher.

Do you want me to make this into a **daily study timetable** (like Day 1: arrays intro + 3 problems, Day 2: reverse array + 2 problems, etc.) so it feels like a structured course?

🔥 That's the spirit! You'll build confidence much faster if you solve **topic-wise problems** in the same order as the roadmap. I'll give you a **curated problem set for each week** — not too overwhelming, but enough to cover all interview patterns.

🗓️ **3-Month DSA Roadmap with Problems**

Month 1 – Foundations

✅ Week 1: Arrays (Basics)

- Find minimum and maximum in an array
- Reverse an array
- Rotate array by k steps (left/right)
- Check if array is sorted
- Move all zeros to the end
- Find the missing number in 1–N
- Kadane's Algorithm (Maximum subarray sum)

✅ Week 2: Strings

- Reverse a string
- Check if a string is palindrome
- Count vowels and consonants
- Check if two strings are anagrams
- Remove duplicates from string
- First non-repeating character
- Longest common prefix

✅ Week 3: Recursion + Sorting

- Factorial of n
- Fibonacci series (recursive)
- Print all subsets of an array
- Generate all permutations of a string
- Tower of Hanoi
- Merge Sort implementation
- Quick Sort implementation

✅ Week 4: Searching

- Linear Search
- Binary Search (basic)
- First and Last occurrence of element in sorted array
- Search in rotated sorted array
- Square root of a number (binary search method)
- Peak element in array
- Floor and Ceil in sorted array

Month 2 – Data Structures

✅ **Week 5: Linked List**

- Implement singly linked list (insert, delete, traverse)
- Reverse a linked list (iterative + recursive)
- Find middle of linked list
- Detect cycle in linked list (Floyd's cycle detection)
- Merge two sorted linked lists

✅ **Week 6: Stacks & Queues**

- Implement stack using array
- Implement stack using 2 queues
- Next Greater Element
- Valid Parentheses (balanced brackets)
- Implement queue using array
- Implement circular queue
- Sliding window maximum

✅ **Week 7: Trees**

- Inorder, Preorder, Postorder traversals (recursive + iterative)
- Height of binary tree

- Count leaf nodes
- Diameter of binary tree
- Check if two trees are identical
- Lowest common ancestor in BST
- Level order traversal (BFS in tree)

✅ **Week 8: Hashing**

- Find first repeating element in array
- Count frequency of elements in array
- Two Sum problem
- Subarray with given sum (hashing approach)
- Longest subarray with zero sum
- Intersection of two arrays
- Longest substring without repeating characters

Month 3 – Intermediate & Interview Prep

✅ **Week 9: Recursion + Backtracking**

- Generate all subsets of a string
- Rat in a Maze problem
- N-Queens problem
- Word Search (matrix backtracking)
- Sudoku Solver

✅ **Week 10: Graphs**

- BFS traversal of graph
- DFS traversal of graph
- Detect cycle in graph (DFS/BFS)
- Shortest path in unweighted graph (BFS)

- Number of islands (grid graph)
- Topological Sort

✅ **Week 11: Dynamic Programming**

- Fibonacci using DP (memoization + tabulation)
- Climbing Stairs problem
- Coin Change problem
- 0/1 Knapsack problem
- Longest Increasing Subsequence (LIS)
- Edit Distance

✅ **Week 12: Final Interview Prep (Mixed)**

- Majority element in array
- Stock Buy and Sell (max profit)
- Rotate Linked List
- Validate Binary Search Tree
- Minimum Spanning Tree (Kruskal/Prim basic)
- Maximum Subarray Product
- Word Break problem

📊 If you complete this **curated set (~150–180 problems)** in 3 months, you'll be at **basic-to-medium DSA mastery level**, good enough for most **fresher interviews**.

👉 Do you want me to also **map these problems day-by-day** (like Day 1: Array min/max + reverse, Day 2: Rotate array + check sorted, etc.), so you can directly follow a daily plan without thinking what to pick?