# Python Problem Solving Plan: Level 3 → Level 6-7 (4 Weeks)

This is a 4-week structured plan designed to help you go from Level 3 to Level 6–7 in Python problem-solving. You’ll spend 30 minutes daily learning core Python concepts, mastering algorithmic patterns, and solving LeetCode problems. This plan focuses on depth, not breadth.

## Week 1: Python Core & Arrays

🎯 Goals:

* • Strengthen Python fundamentals: lists, dicts, sets, functions, comprehensions.
* • Get comfortable solving basic array/string problems.

🗓️ Daily Focus:

- Mon: Lists, slicing, loops — [Exercism.io Python Track](https://exercism.io)

- Tue: Dict & Set usage — practice frequency counts

- Wed: Functions, \*args/\*\*kwargs, list comprehensions

- Thu: LeetCode Easy: Two Sum, Contains Duplicate

- Fri: LeetCode Easy: Maximum Subarray, Valid Anagram

- Sat: String manipulation: reverse, join, format

- Sun: Review + Write notes + solve 1-2 old problems again

## Week 2: Patterns — Two Pointers & Sliding Window

🎯 Goals:

* • Learn Two Pointer and Sliding Window techniques.
* • Improve understanding of hash maps and frequency-based solutions.

🗓️ Daily Focus:

- Mon: Learn Two Pointer concept — [NeetCode Two Pointers](https://neetcode.io)

- Tue: LeetCode: Valid Palindrome, Merge Sorted Array

- Wed: Learn Sliding Window — fixed and variable length

- Thu: LeetCode: Longest Substring Without Repeating Characters

- Fri: LeetCode: Minimum Size Subarray Sum

- Sat: Practice with edge cases & debugging

- Sun: Review + Summarize patterns + revisit mistakes

## Week 3: Data Structures & Recursion

🎯 Goals:

* • Understand stacks, queues, heaps, and recursion basics.
* • Learn when and how to use them in problem-solving.

🗓️ Daily Focus:

- Mon: Stack and Queue — implement using list and deque

- Tue: LeetCode: Valid Parentheses, Implement Queue using Stacks

- Wed: Min Heap with heapq — solve Top K Frequent Elements

- Thu: Recursion 101 — Fibonacci, factorial, basic tree recursion

- Fri: LeetCode: Invert Binary Tree, Maximum Depth of Tree

- Sat: Mix practice + implement all structures from scratch

- Sun: Review + Write down templates for recursion and stack use

## Week 4: Prefix Sum, Hashing, and Light Dynamic Programming

🎯 Goals:

* • Practice prefix sums and hash maps in-depth.
* • Get an intro to dynamic programming (1D).

🗓️ Daily Focus:

- Mon: Prefix sum — range sum queries, subarray sum basics

- Tue: LeetCode: Subarray Sum Equals K, Running Sum of 1D Array

- Wed: Learn 1D DP — Climbing Stairs (Bottom-up and Top-down)

- Thu: LeetCode: House Robber, Fibonacci variations

- Fri: LeetCode: Best Time to Buy and Sell Stock

- Sat: Practice + Review hash map tricks

- Sun: Mock interview (30 min timed), review notes & mistakes

## Tips for Success

* • ⏱️ Be consistent. 30 minutes daily is powerful if done with focus.
* • 🧠 Always review what went wrong in problems — this is where real learning happens.
* • 📓 Keep a journal or GitHub repo of your notes and solutions.
* • 💬 Explain your code out loud or to a peer. It solidifies understanding.
* • 🛠️ Use tools like LeetCode, Exercism, and NeetCode for structured practice.
* • 🔁 Revisit older problems — solving them faster is proof of growth.