

Coding Interview Patterns

Python Templates for Muscle Memory

Version: September 28, 2025

This reference combines the **printer-friendly focus** of the GPT-5 version, the **beautiful code formatting** of the Gemini version, and the **boxed summaries/tables** of the Claude version. Use the boxed notes for quick recognition, and the templates for daily rewriting drills.

Contents

1	Core Patterns & Templates	2
1.1	Depth-First Search (DFS)	2
2	Pattern Recognition Guide	3
3	4-Week Study Schedule	3
4	Daily Practice Routine	3
5	Time & Space Complexity	3

Core Patterns & Templates

Depth-First Search (DFS)

When to Use DFS

Use when exploring connectivity, components, cycle detection, or recursion on trees/graphs.
Complexity $O(V + E)$.

Recursive DFS

```
1 from typing import List, Dict, Set
2
3 def dfs_recursive(graph: Dict[int, List[int]], start: int) -> List[int]:
4     visited: Set[int] = set()
5     order: List[int] = []
6
7     def dfs(u: int) -> None:
8         if u in visited:
9             return
10        visited.add(u)
11        order.append(u)
12        for v in graph.get(u, []):
13            dfs(v)
14
15    dfs(start)
16    return order
```

Iterative DFS

```
1 from typing import List, Dict
2
3 def dfs_iterative(graph: Dict[int, List[int]], start: int) -> List[int]:
4     stack: List[int] = [start]
5     visited = set()
6     order: List[int] = []
7
8     while stack:
9         u = stack.pop()
10        if u in visited:
11            continue
12        visited.add(u)
13        order.append(u)
14        for v in reversed(graph.get(u, [])):
15            if v not in visited:
16                stack.append(v)
```

Pattern Recognition Guide

Trigger Words

- “longest/shortest subarray/substring” \Rightarrow Sliding Window
- “sorted array” \Rightarrow Two Pointers / Binary Search
- “dynamic connectivity” \Rightarrow Union-Find
- “prerequisites ordering” \Rightarrow Topological Sort
- “prefix/suffix queries” \Rightarrow Prefix Sum / Trie

4-Week Study Schedule

Overview

Week 1: Fundamentals (Two Pointers, Sliding Window, Binary Search, Prefix Sum, Kadane)
 Week 2: Graphs (DFS, BFS, Topological Sort, Union-Find, Heaps)
 Week 3: DP & Backtracking
 Week 4: Advanced (Trie, Monotonic Stack, Cyclic Sort, Mixed Drills)

Daily Practice Routine

45–90 Minutes

1. Warm-up: Rewrite 1–2 templates from memory.
2. Focused Drill: Solve 2 problems of today’s pattern.
3. Flash Review: Recap 3 old problems aloud.
4. Retrospective: Write down mistakes and triggers.

Time & Space Complexity

Summary

DFS/BFS: $O(V + E)$ time, $O(V)$ space.
 Binary Search: $O(\log n)$ time.
 Sliding Window: $O(n)$ time, small space.
 DP: $O(n)$ or $O(mn)$ depending on states.
 Union-Find: $\alpha(n)$ amortized.
 Heaps: $O(\log n)$ push/pop.
 Trie: $O(L)$ per word.
 Monotonic Stack: $O(n)$.