Curated DSA Concepts for Python Developers

Core Pythonic Data Structures

- List (`list`): Dynamic array, used for all kinds of sequences.
- Set (`set`): Fast O(1) membership checking and duplicate removal.
- Dictionary (`dict`): Built-in hash table, very common in Python.
- Stack: Use list with append() and pop().
- Queue: Use `collections.deque` for O(1) operations on both ends.
- Heap: Use `heapq` for priority queues.
- Counter / Defaultdict: Use `collections.Counter`, `defaultdict`.

Intermediate Concepts & Patterns

- Sliding Window: Used for max/min subarray problems.
- Two Pointers: Reverse strings, find pairs.
- Prefix Sum: Cumulative totals to solve range queries efficiently.
- Recursion: Core to backtracking and divide-and-conquer.
- Backtracking: Used in N-Queens, permutations, combinations.

Essential Algorithms in Python Style

- Binary Search: Use `bisect` or implement manually.
- Sorting: Use `sorted()`, often with `lambda` keys.
- BFS/DFS: Use `deque` and `dict` to store adjacency lists.
- Hashing: `dict`, `set`, `frozenset`.
- Memoization: Use `functools.lru_cache`.
- Dynamic Programming: Bottom-up with arrays or top-down memoization.
- Greedy Algorithms: Smart selection using sorting and iteration.
- Graph Algorithms: Use dict of list/set for adjacency list.

Interview-Heavy Topics (Python Versions)

Curated DSA Concepts for Python Developers

- Linked Lists: Implement manually using class.
- Trees: Use recursive traversals, define node classes.
- Graphs: Traverse using BFS/DFS with visited set.
- Heap / PQ: Use `heapq`.
- Trie: Use nested dicts or classes.
- LRU Cache: Use OrderedDict or custom DLL.
- Union-Find: Use list with path compression.

Python-Specific DSA Perks

- `lru_cache` from functools: Memoization in one line.
- `Counter` from collections: Frequency counting.
- `itertools`: Combinations, permutations.
- `heapq`: Priority queues.
- `bisect`: Binary search on sorted arrays.
- `enumerate`, `zip`, `map`, `filter`: Pythonic looping and transformations.

Recommended Practice Platforms

- Leetcode: Filter by Python.
- GeeksforGeeks: Great explanations.
- HackerRank: 30 Days of Python.
- InterviewBit: Good for Linked Lists and Trees.
- Striver's DSA Sheet: Try solving in Python.
- Blind 75 List: Master core problems.