1. Sum of Digits
    - Description: Given a non-negative integer, return the sum of its digits.
    - Topics: strings/int operations, loops.
    - Example: input 432 -> output 9
2. Palindrome Checker (string)
    - Description: Check whether a string is a palindrome (ignore case and non-alphanumeric characters).
    - Topics: string processing, regex or isalnum, two-pointer technique.
    - Example: "A man, a plan, a canal: Panama" -> True
3. FizzBuzz (classic)
    - Description: For numbers 1..n, print "Fizz" if divisible by 3, "Buzz" if divisible by 5, "FizzBuzz" if both, else the number.
    - Topics: modulus, loops, conditional logic.
4. Count Vowels
    - Description: Count vowels (a, e, i, o, u) in a string (both cases).
    - Topics: iteration, sets.
5. Merge Two Sorted Lists (arrays)
    - Description: Given two sorted lists, return a single sorted list (without using built-in sort).
    - Topics: two-pointer merge, lists.
6. Remove Duplicates from List (preserve order)
    - Description: From a list, remove duplicates while keeping the original order.
    - Topics: sets, list comprehension, iteration.
7. Factorial (iterative and recursive)
    - Description: Implement factorial both iteratively and recursively. (Handle 0 correctly.)
    - Topics: recursion, loops, base case.

Intermediate (algorithms & data structures, file/IO, dictionaries) 8. Two Sum (return indices)

- Description: Given list nums and target, return indices of two numbers that add up to target.
- Topics: hash map (dict) for O(n) solution.
- Example: nums=[2,7,11,15], target=9 -> [0,1]

9. Longest Common Prefix
    - Description: Find the longest common prefix among an array of strings. Return "" if none.
    - Topics: string comparison, min/max trick or vertical scanning.
10. Anagram Grouping
    - Description: Given a list of strings, group anagrams together.
    - Topics: dict with sorted string or char-count tuple keys.
    - Example: ["eat","tea","tan","ate","nat","bat"] -> [["eat","tea","ate"],["tan","nat"],["bat"]]
11. Valid Parentheses
    - Description: Check if parentheses/brackets are balanced in a string (supports (), {}, []).
    - Topics: stack.
12. Rotate Matrix (NxN) by 90 degrees (in-place)
    - Description: Rotate an N x N matrix 90 degrees clockwise in-place.
    - Topics: matrix indices, layers.
13. Word Frequency from File
    - Description: Given a text file path, return the top k most common words (case-insensitive), ignoring punctuation.
    - Topics: file I/O, collections.Counter, regex.
14. Sliding Window: Max Sum Subarray of Size k
    - Description: Given array and k, find maximum sum of any contiguous subarray of length k.

- Topics: sliding window, O(n) approach.

15. Merge Intervals

    - Description: Given list of intervals [start,end], merge overlapping intervals and return the merged list.
    - Topics: sorting, interval merging.

Advanced (graphs, dynamic programming, performance) 16. LRU Cache (design)
- Description: Implement an LRU (least recently used) cache with get and put in O(1) time.
- Topics: OrderedDict or combined dict + doubly-linked list.

17. Word Ladder (shortest transformation)

    - Description: Given beginWord, endWord, and dictionary list, return length of shortest transformation sequence from beginWord to endWord changing one letter at a time.
    - Topics: BFS, pattern buckets, graph construction.

18. Longest Increasing Subsequence (LIS) — O(n log n)

    - Description: Find length of LIS in an unsorted array. Aim for O(n log n) solution (patience sorting).
    - Topics: binary search, DP optimization.

19. Subarray Sum Equals K (count)

    - Description: Given array and k, count the number of continuous subarrays whose sum equals k.
    - Topics: prefix sums + hashmap for O(n).

20. N-Queens (backtracking)

    - Description: Place N queens on N×N chessboard so that none attack each other; return all valid board configurations (or count).
    - Topics: backtracking, pruning with columns and diagonals sets.

21. Minimum Window Substring

    - Description: Given strings s and t, find minimum window in s which will contain all characters of t (including multiplicity).

- Topics: two-pointer sliding window, character counts.

22. Serialize/Deserialize Binary Tree

    - Description: Design methods to serialize a binary tree to a string and deserialize back to tree (preserve structure).
    - Topics: tree traversal, BFS/DFS, null placeholders.

Challenge/Bonus problems 23. Sudoku Solver (backtracking with heuristics)
- Description: Solve a 9x9 Sudoku board. Aim for efficient backtracking with row/col/box tracking.
- Topics: backtracking, constraint propagation.

24. Regex Engine (simple)

    - Description: Implement a basic regex matcher supporting '.' and '*' (like LeetCode Regex problem).
    - Topics: DP, recursion.

25. Top K Frequent Elements (large data)

    - Description: Given a huge stream of numbers (cannot store all), maintain and return top-k frequent elements. Discuss approaches (heap, count-min sketch).
    - Topics: heaps, streaming algorithms.