Coding Questions Mentioned:

LeetCode / Data Structures & Algorithms (DSA):

- **Linked Lists:**
  - Merge K Sorted Linked Lists (LC 23 / Hard) - Mentioned multiple times.
  - Find K Sorted Lists (Not linked lists) - Solved with sorting then heap.
  - Find the intersection point of two linked lists (Constant space complexity expected).
  - Linked List Cycle Detection.
  - Reorder Linked List (LC 143 / Medium).
  - Reverse linked list with insert, find, and delete follow-ups.
  - Copy list with random pointer.
- **Trees & Graphs:**
  - Variation of Flood Fill Algo.
  - Course Scheduler 1 and 2 variant (LC 207, 210) - Mentioned multiple times.
  - Binary Tree questions (general, debugging assumption mentioned).
  - Find Cycle / Topological Sort (Toposort) - Mentioned multiple times.
  - Longest path in a directed acyclic graph.
  - N-ary tree DFS.
  - Variation of Lowest Common Ancestor (LCA) using adjacency list (not binary tree).
  - DFS/BFS general questions ("when is DFS or BFS better?", specific implementations) - Mentioned multiple times.
  - Tree traversal (Level Order mentioned specifically multiple times, general DFS mentioned).
  - Number of Islands (LC 200) / Count Islands - Mentioned multiple times.
  - Word Search (LC 79 / Medium) - Mentioned multiple times, including follow-ups (same direction DFS, Trie optimization).
  - Word Search II (LC 212 / Hard) - Focus on algorithm/data structure trade-offs.
  - Valid Parentheses (LC 20 / Easy).
  - Rotten Oranges variation (LC 994 / Medium).
  - Clone data structure with a random point (Similar to LC 138 Copy List with Random Pointer, but maybe graph?).
  - Unique Path III variation (LC 980 / Hard).
  - Alien Dictionary (LC 269 / Hard).
  - Find Lowest Common Manager (Similar to LCA).
  - Graph path finding: Given matrix src, dest, blockers, find if path exists (4 directions). Follow up: move only in given dir or right.
- **Arrays / Strings / Subarrays:**
  - Frequency of occurrence of a substring (using HashMap).
  - Medium string questions (general).
  - String rearrangement (LC 767 / Medium).
  - Find all subsets that sum to target (array with 0 and negative numbers).
  - Easy sliding window.
  - Substring problem (general).
  - Longest string without repeating characters (LC 3 / Medium).

- Two pointer + subarray problems - Mentioned multiple times.
- Kadane's algorithm subarray question (Max subarray sum, LC 53 / Medium).
- Integer to English representation variation (LC 273 / Hard).
- List of matched preferences between two lists with least index sum (Similar to LC 599 / Easy).
- String + HashMap related questions.
- Reorganise String (LC 767 / Medium).
- Next Greater Element variation (LC 496 / Easy).
- Stack and string validation.
- Find the shortest superstring (LC 943 / Hard).
- Group Anagrams (LC 49 / Medium) - Mentioned multiple times.
- Best Time to Buy and Sell Stock II variation (LC 122 / Medium) - Mentioned as tougher variation.
- Given char stream and dictionary, find all dictionary words in the stream.

➔ **Heaps / Priority Queues:**
- Merge K Sorted Lists (Can be solved with Heap).
- Priority queue/Max Heap question similar to Kth largest element in an array (LC 215 / Medium).
- Heaps/Hashtable questions (general).
- Median of stream (LC 295 / Hard).
- HashMap + Priority Queue question (not LC tagged).
- Heap type question: top k lottery winners based on item count/price (confusing requirement mentioned).
- K Closest Points to Origin variation (LC 973 / Medium).
- Nearest K elements (LC 658 / Medium).

➔ **HashMaps / Sets:**
- Frequency of occurrence of a substring.
- Basic Hashmap solution (Easy).
- Difference between array and set/hashset.
- K-Frequent elements variation (LC 347 / Medium).
- Hashmap involved in OOD-like question.

➔ **Dynamic Programming (DP):**
- DP mentioned for a string question.
- MST and DP to get objects from a matrix (Hard).

➔ **Other DSA / Logic:**
- Tic Tac Toe variant.
- Course Schedule III variation (LC 630 / Hard).
- Josephus problem (LC 1823 / Medium).
- Implement Stack (Array vs Dynamic/Linked List).
- Implement Fibonacci (Recursive and constant space).
- Basic Calculator variation (LC 224 / Hard) - Mentioned with follow-up.
- Amazon lottery logic (Ambiguous problem-solving).
- Given code, add functionality (e.g., find best fit locker for package) + testing.
- BFS with controlled depth.
- Backtracking questions.

Object-Oriented Design (OOD) / Low-Level Design (LLD):

- ➔ Calculate pizza price - Mentioned multiple times (sometimes as OOD, sometimes OOP).
- ➔ Create a pizza and make it a meal deal (OOP).
- ➔ Create find queries (OOD).
- ➔ Design question (not OOD) on find cycle / top sort.
- ➔ Graph-based OOD with DFS traversal.
- ➔ Unix API design (OOD).
- ➔ Design a pizza store (OOD).
- ➔ Design a 2D game world (add player, food, weapon; scalable) (OOD).
- ➔ Design an interface for encrypting/decrypting algorithms (pros/cons of approaches) (OOD).
- ➔ Convert variable naming patterns (e.g., camelCase to SNAKE_CASE) (OOD).
- ➔ Design Wordle (OOD) - Mentioned multiple times (sometimes asked like LC first).
- ➔ Search files in directories/sub-directories recursively (OOD).
- ➔ Data structure redesign question (OOD).
- ➔ Design pizza pricing system (OOP).
- ➔ Add products with expiry date, remove expired products (Data structure design).
- ➔ Design an elevator (OOD).
- ➔ Pizza slice question (OOD).
- ➔ Design a recommendation system (OOD, command pattern used).
- ➔ Circles crossing problem: Minimal borders to cross from point A to B within nested/separate circles (OOD) - Mentioned multiple times.
- ➔ Load balancing a server (OOD).
- ➔ Design based on Alexa (LLD).
- ➔ Design OOP for Pizza order.
- ➔ Make a basic calculator (from OOD perspective).
- ➔ Adding reviews to a product on Amazon website (Design problem, no code).
- ➔ Detecting errors in a log file (OOD).
- ➔ Encoder/decoder (e.g., AAAAA -> A5) (OOD).
- ➔ Design poker game (straights, flush, etc.) (OOD).
- ➔ Design system to keep elements with expiry (OOD).
- ➔ Design movie ticketing application (OOD).
- ➔ Design memory card game (LLD).
- ➔ Design a log processing system (statistical analysis, avg/peak times, thresholds, summaries) (OOD).
- ➔ Designing a Multimap (LLD).
- ➔ Vending machine problem (LLD).
- ➔ Design a 2-player drone battle game on an 8×8 grid (LLD).
- ➔ Feature flag system (user-specific vs global) (LLD).
- ➔ File system directory search with filters (LLD).

Cache Specific:

- LRU Cache variation - get the most recent user.
- LRU Cache (LC 146 / Medium) - Mentioned multiple times.
- LFU Cache variation (Doubly linked list involved, tough) (LC 460 / Hard).

Leadership Principle (LP) Questions Mentioned:

- **General Themes Repeatedly Mentioned:**
  - Dive Deep (multiple mentions, including project deep dives, troubleshooting)
  - Ownership (multiple mentions)
  - Learn and Be Curious (multiple mentions, including learning something new/unfamiliar, learning process)
  - Bias for Action / Taking Initiative (multiple mentions, including taking on work outside responsibilities, going beyond)
  - Deliver Results / Meeting Deadlines (multiple mentions, including tight deadlines, missed deadlines, obstacles, commitments)
  - Invent and Simplify
  - Customer Obsession (explicitly mentioned, also implied in questions about client betterment, customer feedback, working backwards from customer)
  - Earn Trust (mentioned)
  - Disagree and Commit (implied in questions about disagreement with team/manager/peer)
  - Think Big (implied in questions about risk-taking, influencing)
  - Frugality (mentioned via budget decision question)
  - Are Right, A Lot (implied in defending choices, pushback handling)
  - Hire and Develop the Best (mentioned via mentoring question)
  - Insist on the Highest Standards (implied in feedback questions, improving metrics)

- **Specific Questions/Scenarios:**
  - Tell me about yourself / Introduction / Resume/Project deep dives (very common)
  - A time you worked on something you didn't know / weren't familiar with.
  - Found a metric not performing well, what did you do to improve it?
  - When you didn't agree with your team/peer/manager? (multiple mentions)
  - A time you influenced someone / someone influenced you.
  - A time you took a risk.
  - What was a hard problem you solved? What were the results? What could you have done better?
  - Were you able to reach a commitment?
  - Twisted/Complex LP questions (difficult to recall).
  - Follow-ups probing the story / proving you did the work / digging for inconsistencies.
  - Failures or negative feedback related questions.
  - Team conflict scenario.
  - Sacrificed something for the long-term betterment of the client/customer.
  - Significant hurdle in an academic/industrial project.

- Difficult decision you made.
- What feedback did you get (manager/professor)? How did you use it to improve? / Responding to critical feedback. (multiple mentions)
- Took on something outside your responsibilities? How did you navigate it? / Went beyond your usual responsibilities? (multiple mentions)
- Used a metric to guide yourself? How was it? / Detailed questions about metrics (how measured, why chosen).
- A time you needed to solve a problem, how did you learn?
- Didn't know what to do to solve a challenging problem? How did you learn options and decide?
- Trying to understand a complex problem, used analysis? Why was it complex?
- Team had under-performance issue, what did you do? / How did you raise team morale?
- Convinced a stakeholder/peer to prioritize work you needed them to do? / How did you influence a peer with different opinions?
- Obstacles faced delivering a project, how resolved?
- Fixing a crazy bug under a deadline.
- Experience of crazy issues happening before deadline.
- Any extra work done outside main work to grow?
- Challenging project you worked on. (multiple mentions)
- Tell me about a time you almost missed a deadline.
- Biggest technical challenge faced so far.
- Coolest project you've ever worked on.
- Time you felt you reached a point wanting to learn more / know your limits.
- What is your learning process / how do you approach new problems?
- Time you helped mentor/assist a teammate having trouble? / Mentoring experience.
- Troubleshooting scenarios.
- Proposed something that became the team's direction?
- Received pushback on your idea?
- Time you hindered progress?
- Made a decision that helped your team's budget?
- How did you deal with customer feedback?