

Amazon's coding exercises tend to be **data structures and algorithms problems** that test problem-solving under constraints. They usually come in the format of **LeetCode Medium/Hard** and often tie to arrays, strings, graphs, and system-like problems.

Here's a breakdown with examples:

1. Arrays & Strings

- Two Sum / 3Sum variations
 - Longest substring without repeating characters
 - Sliding window problems (maximum subarray sum $\leq k$, minimum window substring)
 - Search in a 2D matrix (sorted rows & cols)
 - Rotate matrix, spiral traversal
 - Trapping rain water
-

2. Hashing & LRU-style problems

- LRU Cache (HashMap + Doubly Linked List)
 - Group Anagrams
 - Top K frequent elements (heap or bucket sort)
 - Subarray sum equals K (prefix sum + hashmap)
-

3. Linked List

- Detect cycle and return start (Floyd's algorithm)
 - Merge K sorted lists
 - Reverse nodes in K-group
-

4. Stacks & Queues

- Min stack (support O(1) getMin)
 - Valid parentheses with stack
 - Sliding window maximum (deque)
 - Implement queue using stacks
-

5. Trees & Graphs

- Lowest Common Ancestor (binary tree / BST)
 - Serialize and deserialize binary tree
 - Word ladder (BFS shortest path)
 - Course schedule (topological sort)
 - Number of islands (DFS/BFS/Union Find)
-

6. Dynamic Programming

- Longest increasing subsequence
 - Edit distance
 - Coin change / min coins
 - Knapsack variants
 - Unique paths with obstacles
-

7. Math & Bit Manipulation

- Implement division without using *, /, %
 - Single number (XOR trick)
 - Reverse bits, count set bits
 - Uniform random generator ($\text{rand7} \rightarrow \text{rand10}$)
-

8. System-like / Amazon-specific

These sometimes show up because they tie to scale:

- Design a URL shortener (generate short codes, handle collisions)
 - Stock price fluctuation tracker (with $\text{getMax}/\text{getMin}$ in $O(1)$)
 - Log storage system with prefix filtering
 - Implement an order book (matching buy/sell orders)
 - K most recently used (variation of LRU/LFU)
-

9. Complexity Expectations

- Most exercises should run in $O(n)$ or $O(n \log n)$.

- Space should be justified — Amazon loves asking “can you optimize further?”.
-

Tip: Practice on LeetCode "Amazon" tagged questions, and be prepared to:

- Write working code in 30–40 minutes.
 - Explain trade-offs.
 - Test with edge cases (empty input, large input, duplicates).
-

Do you want me to put together a **cheat sheet with sample problems + C++ template solutions (arrays, hashmaps, LRU, graphs, DP)** that you can use as a prep pack for Amazon coding rounds?