Amazon SDE I Interview: HashMaps & Sets Cheat Sheet

Core Concepts

HashMaps store key-value pairs with average O(1) access and update time.

Sets store unique elements and offer O(1) average time for add, remove, and lookup.

Use cases: frequency count, uniqueness check, fast lookup, grouping by key.

Common Patterns

- Frequency counting (e.g., Valid Anagram)
- First unique/non-repeating character (use map with index tracking)
- Grouping items (e.g., Group Anagrams)
- Sliding window with set for duplicates (e.g., Longest Substring Without Repeating Characters)
- Prefix sum with map for subarray sum problems

Best Practices

- Always check if key exists before accessing: `if key in hashmap:` or use `.get()`
- Use defaultdict or Counter in Python for default values
- Be careful when modifying dictionary/set size during iteration
- Watch out for hash collisions and ensure proper key types

Time Complexities

- HashMap/Set Insert: O(1) average
- HashMap/Set Lookup: O(1) average
- Worst-case: O(n) due to collisions (rare with good hash functions)

Top LeetCode Problems

1. Two Sum
2. Valid Anagram
3. Group Anagrams
4. Top K Frequent Elements

Amazon SDE I Interview: HashMaps & Sets Cheat Sheet

5. Longest Substring Without Repeating Characters
6. Subarray Sum Equals K
7. Longest Consecutive Sequence