1. **E-commerce Cart System:**

**ANSWER:**  
I’d use a LinkedHashMap to store items with their quantities since it keeps things in the order they were added and handles duplicates easily.

1. **Auto-Complete Feature:**

**ANSWER:**  
A TreeSet is perfect here because it keeps all the words sorted alphabetically and avoids any duplicates.

1. **LRU Cache:**

**ANSWER:**  
I’d go with a LinkedHashMap set to access-order and override a method to remove the oldest item when the cache is full.

1. **Leaderboard System:**

**ANSWER:**  
A PriorityQueue with a custom comparator works well to always keep the highest scoring players at the top.

1. **Real-Time Notification System:**

**ANSWER:**  
An ArrayDeque would do the job since it processes events in the exact order they come in.

1. **Eliminating Duplicate Emails:**

**ANSWER:**  
Using a LinkedHashSet makes sense because it removes duplicates but still keeps the first occurrence in the original order.

1. **Directory Structure Simulation:**

**ANSWER:**  
I’d model it as a tree where each folder is a node holding its subfolders and files in a map.

1. **Priority-based Task Scheduling:**

**ANSWER:**  
A PriorityQueue is ideal here since it lets you fetch tasks based on their priority without sorting manually.