

# Collections Problem Set

## Problem 1

You ingest weekly product lists from multiple vendors. Each vendor uses their own alias for a product (e.g., “Galaxy S24 Ultra 256 Black” vs “SM-S928B-256-BLK”). A separate file maps every alias to sku (Stock Keeping Unit), your canonical identifier.

Link to sample input:

### Task

- Build the unique set of SKUs offered this week (by mapping vendor aliases to skus).
- Compare last week’s SKUs to find new and discontinued items.
- Flag data quality issues:
  - Unknown aliases (appear in vendor files but missing from the alias map)
  - Alias conflicts (same alias mapped to different skus in the alias map)

### Inputs

- alias\_map.csv: alias,sku
- last\_week\_skus.txt: one sku per line
- One or more vendor CSVs: alias,price,stock,... (we only need alias)

## Problem 2: Ticket Triage

You run a helpdesk. Tickets arrive with a priority and an arrival order. The team must always handle the highest-priority ticket next. If two tickets share the same priority, handle the one that arrived earlier.

### Task

- Read tickets from a file.
- Insert all tickets into a priority queue ordered by:
  - priority (e.g., 1 = Critical, 2 = High, 3 = Normal...), then
  - arrivalIndex (smaller = arrived earlier).
- Produce:
  - The full service order (IDs in the exact order they will be handled), and
  - (optionally) the top K tickets to handle right now (given K)

### Inputs

- tickets.csv: ticketId,priority,arrivalIndex
  - Example row: T-1021,2,17 means priority 2 (High), arrived 17th

# Problem 3: Order Fulfillment

You run a small warehouse. Orders arrive in a line and must be served FIFO. Each order requests a quantity of a SKU. If there isn't enough stock for an order right now, it should wait in a backorder queue. We'll make one clean pass: separate orders that can be fulfilled immediately from those that must wait.

## Task

- Build an inventory lookup from file (sku to availableQty).
- Read incoming orders in arrival order into a main FIFO queue.
- While the main queue has orders:
  - If inventory has enough for the order's SKU, fulfill and decrement stock.
  - Else, move the order to a backorder FIFO queue.
- Produce simple reports:
  - fulfilledCount, backorderedCount
  - Lists (or samples) of fulfilledOrderIds and backorderedOrderIds
  - remainingInventory per SKU
- Optional
  - Ignore duplicate orderIds (use a Set).
  - Track unknown SKUs seen in orders but not in inventory.

## Inputs

- inventory.csv: sku,qty
- orders.csv: orderId,sku,qty (*Arrival order in this file is the processing order.*)