DSA Battleground: Day 14

Question 27: Job Scheduling with Priority and Dependencies

💡 Problem Statement:

You're given a list of jobs. Each job has:
- a `name`,
- a `priority` (higher number = higher priority),
- a list of other jobs it depends on (must be done first).

You need to write a function `schedule_jobs(jobs: List[Dict]) -> List[str]` that:
- Runs jobs only after all their dependencies are complete.
- Picks the job with the highest priority (among available ones).
- If a job cannot be scheduled because one of its dependencies is missing forever (i.e., not in the list), skip it.

🧪 Input:

```
jobs = [
   {"name": "jobA", "priority": 4, "depends_on": ["jobB"]},
   {"name": "jobB", "priority": 5, "depends_on": []},
   {"name": "jobC", "priority": 3, "depends_on": ["jobB", "jobA"]},
   {"name": "jobD", "priority": 6, "depends_on": ["jobX"]}  # jobX
doesn't exist
]
```

Expected Output:

```
['jobB', 'jobA', 'jobC']
```

🔍 **Explanation:**
- `jobB` has no dependency → scheduled first.
- `jobA` depends only on `jobB` (done) → scheduled next.
- `jobC` depends on both `jobB` and `jobA` (both done) → scheduled.
- `jobD` depends on `jobX` (not found) → skipped.

Logic Hint:
Use a loop to repeatedly check which jobs can be added (whose dependencies are all in the scheduled list). At each step, pick the one with highest priority among them.

Question 28: Analyzing Survey Data

💡 Problem Statement:

You're given a list of survey responses from customers. Each response is a dictionary with:

- 'name': name of the customer,
- 'responses': a dictionary where keys are **question IDs** and values are **ratings from 1 to 5**.

Write a function find_consistent_customers(surveys: List[Dict]) -> List[str] that returns the names of all customers who:

1. Answered **at least 3 questions**, and
2. Gave the **same rating** to **every question they answered**.

🧪 Input:

surveys = [

{'name': 'Amit', 'responses': {'Q1': 5, 'Q2': 5, 'Q3': 5}},

{'name': 'Priya', 'responses': {'Q1': 4, 'Q2': 4, 'Q3': 5}},

{'name': 'Sahil', 'responses': {'Q1': 3, 'Q2': 3, 'Q3': 3, 'Q4': 3}},

{'name': 'Riya', 'responses': {'Q1': 1, 'Q2': 2}},

{'name': 'Karan', 'responses': {'Q1': 5, 'Q2': 5, 'Q3': 5, 'Q4': 4}}

]

Expected Output:

['Amit', 'Sahil']

🔍 **Explanation:**

- **Amit** → 3 questions, all 5 → ✅
- **Priya** → 3 questions, but has 5 and 4 → ❌
- **Sahil** → 4 questions, all 3 → ✅
- **Riya** → only 2 questions → ❌
- **Karan** → 4 questions, but has both 5 and 4 → ❌