LeetCode 937. Reorder Data in Log Files

1. Problem Title & Link

- **937. Reorder Data in Log Files**
- https://leetcode.com/problems/reorder-data-in-log-files/

2. Problem Statement (Short Summary)

We are given an array of logs. Each log is a string with an identifier and content separated by a space.

- **Letter-logs**: content has lowercase English letters.
- **Digit-logs**: content has digits.

Rules:

1. All **letter-logs** come before any **digit-log**.
2. Letter-logs are sorted by content, and if contents are the same, then by identifier.
3. Digit-logs remain in the same order.

3. Examples (Input → Output)

Input: logs = ["dig1 8 1 5 1","let1 art can","dig2 3 6","let2 own kit dig","let3 art zero"]

Output: ["let1 art can","let3 art zero","let2 own kit dig","dig1 8 1 5 1","dig2 3 6"]

Input: logs = ["a1 9 2 3 1","g1 act car","zo4 4 7","ab1 off key dog","a8 act zoo"]

Output: ["g1 act car","a8 act zoo","ab1 off key dog","a1 9 2 3 1","zo4 4 7"]

4. Constraints

- 1 <= logs.length <= 100
- 3 <= logs[i].length <= 100
- Each log has at least one identifier + one word.

5. Thought Process (Step by Step)

- Split logs into two groups: **letter-logs** and **digit-logs**.
- Sort letter-logs with a custom rule:
  - Primary key → content (after identifier).
  - Secondary key → identifier.
- Append digit-logs at the end (no sorting).

6. Pseudocode (Language-Independent)

```
function reorderLogFiles(logs):
    letters = []
    digits = []

    for log in logs:
        split log into id and content
        if first char of content is digit:
            add to digits
        else:
            add to letters

    sort letters by (content, id)
    return letters + digits
```

7. Code Implementation

✅ Python

```python
class Solution:
    def reorderLogFiles(self, logs: List[str]) -> List[str]:
        letters, digits = [], []

        for log in logs:
            identifier, rest = log.split(" ", 1)
            if rest[0].isdigit():
                digits.append(log)
            else:
                letters.append((rest, identifier, log))

        letters.sort(key=lambda x: (x[0], x[1]))
        return [log for _, _, log in letters] + digits
```

✅ Java

```java
class Solution {
    public String[] reorderLogFiles(String[] logs) {
        Arrays.sort(logs, (a, b) -> {
            String[] splitA = a.split(" ", 2);
            String[] splitB = b.split(" ", 2);

            boolean isDigitA = Character.isDigit(splitA[1].charAt(0));
            boolean isDigitB = Character.isDigit(splitB[1].charAt(0));

            if (!isDigitA && !isDigitB) {
                int cmp = splitA[1].compareTo(splitB[1]);
                if (cmp != 0) return cmp;
                return splitA[0].compareTo(splitB[0]);
            }
            if (!isDigitA && isDigitB) return -1;
            if (isDigitA && !isDigitB) return 1;
            return 0;
        });
        return logs;
    }
}
```

8. Time & Space Complexity Analysis

- Sorting complexity: O(n log n * k), where k = max log length.
- Space: O(n) for separating logs.

9. Common Mistakes / Edge Cases

- Forgetting that **digit-logs keep original order**.
- Sorting only by identifier, not content first.
- Assuming logs always have at least 2 words.

10. Variations / Follow-Ups

- Custom log sorting for different rules.
- Streaming version where logs come continuously.

11. Dry Run (Step by Step Execution)

👉 Input:

["dig1 8 1 5 1","let1 art can","dig2 3 6","let2 own kit dig","let3 art zero"]

1. Separate logs:
   - Letters:
     - "let1 art can" → (content: "art can", id: "let1")
     - "let2 own kit dig" → (content: "own kit dig", id: "let2")
     - "let3 art zero" → (content: "art zero", id: "let3")
   - Digits:
     - "dig1 8 1 5 1", "dig2 3 6"
2. Sort letters by (content, id)
   - "art can" < "art zero" → "let1" before "let3"
   - "own kit dig" comes after both
3. Sorted letters = ["let1 art can", "let3 art zero", "let2 own kit dig"]

4. Append digits (unchanged order):
   → Final =
   ["let1 art can","let3 art zero","let2 own kit dig","dig1 8 1 5 1","dig2 3 6"]

✅ Output matches expected.