Problem of the Week – Dropbox Question

🏢 Company

Dropbox

🎯 Difficulty

Hard

📌 Topics

- Hashing
- Sliding Window
- String Matching
- Two Pointers

📖 Problem Description

You are given a string `s` and a list of words `words`, where each word in `words` has the same length. Your task is to **find all starting indices** of substrings in `s` that form a concatenation of **every word in `words` exactly once** (without any extra characters in between).

If no such substring exists, return an empty list.
The order of indices in the output does not matter.

🖋️ Input Format

- A string `s`.
- A list of words `words` (all words have the same length).

🖋️ Output Format

- A list of integers representing all starting indices of valid substrings.

💡 Examples

Example 1

Input:

```
s = "dogcatcatcodecatdog"
words = ["cat", "dog"]
```

Output:

```
[0, 13]
```

Explanation:

- At index `0`, substring `"dogcat"` contains `"dog"` + `"cat"`.
- At index `13`, substring `"catdog"` contains `"cat"` + `"dog"`.

Example 2

Input:

```
s = "barfoobazbitbyte"
words = ["dog", "cat"]
```

Output:

```
[]
```

Explanation:
No substring in `s` matches concatenation of `"dog"` and `"cat"`.

⚡ Constraints

- `1 <= len(s) <= 10^5`
- `1 <= len(words) <= 5000`
- Each word has length `k` such that `1 <= k <= 10`.

🚀 Challenge

Solve this efficiently using **Sliding Window + Hash Maps** in `O(N * wordLen * numWords)` time.

📚 References

- LeetCode: Substring with Concatenation of All Words
- GeeksforGeeks: Find all starting indices of concatenated words