# Substring with Concatenation of All Words

We must find all starting indices in s where a substring is formed by concatenating all words exactly once (any order).

## Key Observations:

1. All words have the same length.
2. Valid concatenated substring length = word_length * number_of_words
3. We must check every possible starting index in s where such a substring could occur.

## Approach: Sliding Window + Hash Maps:

## Steps:

1. Build a frequency map (word Count) for words.
2. Let:
   - wordLen = length of each word.
   - totalLen = wordLen * words.size()
3. For each possible offset from 0 to wordLen - 1:
   - Use two pointers (left and right) to form a sliding window.
   - Track counts in seen map.
   - If counts exceed wordCount or encounter invalid word:
     - Move left pointer to shrink window.
   - If window size == total words, record left as valid index.

## Time Complexity:

- $O(n * wordLen)$ worst case, where $n = s.length()$
- Efficient compared to checking every permutation.