30. Substring with Concatenation of All Words

You are given a string s and an array of strings words. All the strings of words are of the same length.

A concatenated substring in s is a substring that contains all the strings of any permutation of words concatenated.

For example, if words = ["ab","cd","ef"], then "abcdef", "abefcd", "cdabef", "cdefab", "efabcd", and "efcdab" are all concatenated strings. "acdbef" is not a concatenated substring because it is not the concatenation of any permutation of words.

Return *the starting indices of all the concatenated substrings in* s. You can return the answer in any order.

Example 1:

Input: s = "barfoothefoobarman", words = ["foo","bar"]

Output: [0,9]

Explanation: Since words.length == 2 and words[i].length == 3, the concatenated substring has to be of length 6.

The substring starting at 0 is "barfoo". It is the concatenation of ["bar","foo"] which is a permutation of words.

The substring starting at 9 is "foobar". It is the concatenation of ["foo","bar"] which is a permutation of words.

The output order does not matter. Returning [9,0] is fine too.

Example 2:

Input: s = "wordgoodgoodgoodbestword", words = ["word","good","best","word"]

Output: []

Explanation: Since words.length == 4 and words[i].length == 4, the concatenated substring has to be of length 16.

There is no substring of length 16 in s that is equal to the concatenation of any permutation of words.

We return an empty array.

Example 3:

Input: s = "barfoofoobarthefoobarman", words = ["bar","foo","the"]

Output: [6,9,12]

Explanation: Since words.length == 3 and words[i].length == 3, the concatenated substring has to be of length 9.

The substring starting at 6 is "foobarthe". It is the concatenation of ["foo","bar","the"] which is a permutation of words.

The substring starting at 9 is "barthefoo". It is the concatenation of ["bar","the","foo"] which is a permutation of words.

The substring starting at 12 is "thefoobar". It is the concatenation of ["the","foo","bar"] which is a permutation of words.

Constraints:

$1 <= s.length <= 10^4$

1 <= words.length <= 5000

1 <= words[i].length <= 30

s and words[i] consist of lowercase English letters.