Problem 3093: Longest Common Suffix Queries

Problem Information

Difficulty: Hard
Acceptance Rate: 35.88%
Paid Only: No
Tags: Array, String, Trie

Problem Description

You are given two arrays of strings `wordsContainer` and `wordsQuery`.

For each `wordsQuery[i]`, you need to find a string from `wordsContainer` that has the **longest common suffix** with `wordsQuery[i]`. If there are two or more strings in `wordsContainer` that share the longest common suffix, find the string that is the **smallest** in length. If there are two or more such strings that have the **same** smallest length, find the one that occurred **earlier** in `wordsContainer`.

Return _an array of integers_`ans` _, where_`ans[i]`_is the index of the string in_`wordsContainer` _that has the**longest common suffix** with _`wordsQuery[i]`_._

Example 1:

Input: wordsContainer = ["abcd","bcd","xbcd"], wordsQuery = ["cd","bcd","xyz"]

Output: [1,1,1]

Explanation:

Let's look at each `wordsQuery[i]` separately:

* For `wordsQuery[0] = "cd"`, strings from `wordsContainer` that share the longest common suffix `"cd"` are at indices 0, 1, and 2. Among these, the answer is the string at index 1 because it has the shortest length of 3. * For `wordsQuery[1] = "bcd"`, strings from `wordsContainer` that share the longest common suffix `"bcd"` are at indices 0, 1, and 2. Among these, the answer is the string at index 1 because it has the shortest length of 3. * For

`wordsQuery[2] = "xyz"`, there is no string from `wordsContainer` that shares a common suffix. Hence the longest common suffix is `""`, that is shared with strings at index 0, 1, and 2. Among these, the answer is the string at index 1 because it has the shortest length of 3.

Example 2:

Input: wordsContainer = ["abcdefgh","poiuygh","ghghgh"], wordsQuery = ["gh","acbfgh","acbfegh"]

Output: [2,0,2]

Explanation:

Let's look at each `wordsQuery[i]` separately:

* For `wordsQuery[0] = "gh"`, strings from `wordsContainer` that share the longest common suffix `"gh"` are at indices 0, 1, and 2. Among these, the answer is the string at index 2 because it has the shortest length of 6. * For `wordsQuery[1] = "acbfgh"`, only the string at index 0 shares the longest common suffix `"fgh"`. Hence it is the answer, even though the string at index 2 is shorter. * For `wordsQuery[2] = "acbfegh"`, strings from `wordsContainer` that share the longest common suffix `"gh"` are at indices 0, 1, and 2. Among these, the answer is the string at index 2 because it has the shortest length of 6.

Constraints:

* `1 <= wordsContainer.length, wordsQuery.length <= 104` * `1 <= wordsContainer[i].length <= 5 * 103` * `1 <= wordsQuery[i].length <= 5 * 103` * `wordsContainer[i]` consists only of lowercase English letters. * `wordsQuery[i]` consists only of lowercase English letters. * Sum of `wordsContainer[i].length` is at most `5 * 105`. * Sum of `wordsQuery[i].length` is at most `5 * 105`.

Code Snippets

C++:

```
class Solution {
public:
vector<int> stringIndices(vector<string>& wordsContainer, vector<string>&
wordsQuery) {
```

```
}
};
```

Java:

```
class Solution {
public int[] stringIndices(String[] wordsContainer, String[] wordsQuery) {

}
}
```

Python3:

```
class Solution:
def stringIndices(self, wordsContainer: List[str], wordsQuery: List[str]) ->
List[int]:
```