Problem 3302: Find the Lexicographically Smallest Valid Sequence

Problem Information

Difficulty: Medium
Acceptance Rate: 21.15%
Paid Only: No
Tags: Two Pointers, String, Dynamic Programming, Greedy

Problem Description

You are given two strings `word1` and `word2`.

A string `x` is called **almost equal** to `y` if you can change **at most** one character in `x` to make it _identical_ to `y`.

A sequence of indices `seq` is called **valid** if:

* The indices are sorted in **ascending** order. * _Concatenating_ the characters at these indices in `word1` in **the same** order results in a string that is **almost equal** to `word2`.

Return an array of size `word2.length` representing the lexicographically smallest **valid** sequence of indices. If no such sequence of indices exists, return an **empty** array.

Note that the answer must represent the _lexicographically smallest array_ , **not** the corresponding string formed by those indices.

Example 1:

Input: word1 = "vbcca", word2 = "abc"

Output: [0,1,2]

Explanation:

The lexicographically smallest valid sequence of indices is `[0, 1, 2]`:

* Change `word1[0]` to `'a'`. * `word1[1]` is already `'b'`. * `word1[2]` is already `'c'`.

Example 2:

Input: word1 = "bacdc", word2 = "abc"

Output: [1,2,4]

Explanation:

The lexicographically smallest valid sequence of indices is `[1, 2, 4]`:

* `word1[1]` is already `'a'`. * Change `word1[2]` to `'b'`. * `word1[4]` is already `'c'`.

Example 3:

Input: word1 = "aaaaaa", word2 = "aaabc"

Output: []

Explanation:

There is no valid sequence of indices.

Example 4:

Input: word1 = "abc", word2 = "ab"

Output: [0,1]

Constraints:

* `1 <= word2.length < word1.length <= 3 * 105` * `word1` and `word2` consist only of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
vector<int> validSequence(string word1, string word2) {

}
};
```

Java:

```
class Solution {
public int[] validSequence(String word1, String word2) {

}
}
```

Python3:

```
class Solution:
def validSequence(self, word1: str, word2: str) -> List[int]:
```