6918. Shortest String That Contains Three Strings

My Submissions (/contest/weekly-contest-356/problems/shortest-string-that-contains-three-strings/submissions/)

Back to Contest (/contest/weekly-contest-356/)

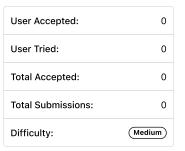
Given three strings a, b, and c, your task is to find a string that has the **minimum** length and contains all three strings as **substrings**.

If there are multiple such strings, return the lexicographically smallest one.

Return a string denoting the answer to the problem.

Notes

- A string a is **lexicographically smaller** than a string b (of the same length) if in the first position where a and b differ, string a has a letter that appears **earlier** in the alphabet than the corresponding letter in b.
- A **substring** is a contiguous sequence of characters within a string.



Example 1:

```
Input: a = "abc", b = "bca", c = "aaa"
Output: "aaabca"
Explanation: We show that "aaabca" contains all the given strings: a = ans[2...4], b = ans[3..5], c = ans[0..2]. It can be shown that "aaabca" contains all the given strings: a = ans[2...4], b = ans[3..5], c = ans[0..2].
```

Example 2:

```
Input: a = "ab", b = "ba", c = "aba"
Output: "aba"
Explanation: We show that the string "aba" contains all the given strings: a = ans[0..1], b = ans[1..2], c = ans[0..2]. Since the string that the string "aba" contains all the given strings: a = ans[0..1], b = ans[1..2], c = ans[0..2].
```

Constraints:

- 1 <= a.length, b.length, c.length <= 100
- a, b, c consist only of lowercase English letters.

```
\boldsymbol{\varepsilon}
JavaScript
    const lexical_smallest_comp = (x, y) \Rightarrow x < y ? -1 : x > y ? 1 : 0;
1
 2
 З ч
    const minimumString = (a, b, c) \Rightarrow \{
 4
         let d = [merge(merge(a, b), c), merge(merge(a, c), b), merge(merge(b, a), c), merge(merge(b, c), a), merge(merge(c,
    a), b), merge(merge(c, b), a)];
 5 ,
         d.sort((x, y) \Rightarrow \{
 6
             if (x.length != y.length) return x.length - y.length;
 7
             return lexical_smallest_comp(x, y);
 8
         3)
 9
         return d[0];
10
    };
11
12 •
    const merge = (s, t) \Rightarrow \{
         if (s.index0f(t) != -1) return s;
13
14
         for (let l = Math.min(s.length, t.length); l > 0; l--) {
15
             if (s.slice(-l) == t.slice(0, l)) return s.slice(0, -l) + t;
16
17
         return s + t;
    };
18
```

□ Custom Testcase

