

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.

User Accepted:	0
User Tried:	0
Total Accepted:	0
Total Submissions:	0
Difficulty:	Medium

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 no other string of the same length can contain all three strings.

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 there is no other string of the same length that contains all three strings, "aba" is the answer.

Constraints:

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

JavaScript

📄

↺

⚙

```
1 const lexical_smallest_comp = (x, y) => x < y ? -1 : x > y ? 1 : 0;
2
3 const minimumString = (a, b, c) => {
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) => {
6     if (x.length !== y.length) return x.length - y.length;
7     return lexical_smallest_comp(x, y);
8   })
9   return d[0];
10 };
11
12 const merge = (s, t) => {
13   if (s.indexOf(t) !== -1) return s;
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

 Run

 Submit

Submission Result: **Accepted** (/submissions/detail/1007447267/) ⓘ

More Details > (/submissions/detail/1007447267/)

Share your acceptance!

Copyright © 2023 LeetCode

[Help Center \(/support\)](#) | [Jobs \(/jobs\)](#) | [Bug Bounty \(/bugbounty\)](#) | [Online Interview \(/interview/\)](#) | [Students \(/student\)](#) | [Terms \(/terms\)](#) | [Privacy Policy \(/privacy\)](#)

 [United States \(/region\)](#)