Problem List

Description | Editorial | Solutions | Accepted | Submissions

3571. Find the Shortest Superstring II

Solved

Premium

Easy | Topics | Hint

You are given **two** strings, `s1` and `s2`. Return the **shortest** *possible* string that contains both `s1` and `s2` as substrings. If there are multiple valid answers, return *any* one of them.

A **substring** is a contiguous sequence of characters within a string.

Example 1:

> **Input:** `s1 = "aba", s2 = "bab"`
>
> **Output:** `"abab"`
>
> **Explanation:**
>
> `"abab"` is the shortest string that contains both `"aba"` and `"bab"` as substrings.

Example 2:

> **Input:** `s1 = "aa", s2 = "aaa"`
>
> **Output:** `"aaa"`
>
> **Explanation:**
>
> `"aa"` is already contained within `"aaa"`, so the shortest superstring is `"aaa"`.

Constraints:

- `1 <= s1.length <= 100`
- `1 <= s2.length <= 100`
- `s1` and `s2` consist of lowercase English letters only.

Seen this question in a real interview before? 1/5

Yes | No

Accepted 638/1.3K | Acceptance Rate 50.0%

Topics

Hint 1

Similar Questions

Discussion (3)

8 | 3 | 0 Online

Code

Python3 • Auto

```python
class Solution:
    def shortestSuperstring(self, s1: str, s2: str) -> str:

        n, m = len(s1), len(s2)

        if n > m:                           # <-- 1)
            s1, s2, n = s2, s1, m           #

        if s1 in s2: return s2              # <-- 2)

        for i in range(1, n):
            if s1[i:] == s2[:n-i]:          #
                return s1[:i] + s2          # <-- 3)
            if s2[-i:] == s1[:i]:           #
                return s2 + s1[i:]          #

        return s1 + s2

```

Saved

Ln 18, Col 9

Testcase | Test Result

Accepted Runtime: 0 ms

Case 1 | Case 2

Input

s1 =
"aba"

s2 =
"bab"

Output