Problem 686: Repeated String Match

Problem Information

Difficulty: Medium
Acceptance Rate: 37.85%
Paid Only: No
Tags: String, String Matching

Problem Description

Given two strings `a` and `b`, return _the minimum number of times you should repeat string_`a` _so that string_ `b` _is a substring of it_. If it is impossible for `b`■■■■■■ to be a substring of `a` after repeating it, return `-1`.

Notice: string `"abc"` repeated 0 times is `""`, repeated 1 time is `"abc"` and repeated 2 times is `"abcabc"`.

Example 1:

Input: a = "abcd", b = "cdabcdab" **Output:** 3 **Explanation:** We return 3 because by repeating a three times "ab**cdabcdab** cd", b is a substring of it.

Example 2:

Input: a = "a", b = "aa" **Output:** 2

Constraints:

* `1 <= a.length, b.length <= 104` * `a` and `b` consist of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
int repeatedStringMatch(string a, string b) {

}
};
```

Java:

```
class Solution {
public int repeatedStringMatch(String a, String b) {

}
}
```

Python3:

```
class Solution:
def repeatedStringMatch(self, a: str, b: str) -> int:
```