Problem 466: Count The Repetitions

Problem Information

Difficulty: Hard
Acceptance Rate: 33.11%
Paid Only: No
Tags: String, Dynamic Programming

Problem Description

We define `str = [s, n]` as the string `str` which consists of the string `s` concatenated `n` times.

* For example, `str == ["abc", 3] =="abcabcabc"`.

We define that string `s1` can be obtained from string `s2` if we can remove some characters from `s2` such that it becomes `s1`.

* For example, `s1 = "abc"` can be obtained from `s2 = "ab** _dbe_** c"` based on our definition by removing the bolded underlined characters.

You are given two strings `s1` and `s2` and two integers `n1` and `n2`. You have the two strings `str1 = [s1, n1]` and `str2 = [s2, n2]`.

Return _the maximum integer_`m` _such that_`str = [str2, m]`_can be obtained from_`str1`.

Example 1:

Input: s1 = "acb", n1 = 4, s2 = "ab", n2 = 2 **Output:** 2

Example 2:

Input: s1 = "acb", n1 = 1, s2 = "acb", n2 = 1 **Output:** 1

Constraints:

* `1 <= s1.length, s2.length <= 100` * `s1` and `s2` consist of lowercase English letters. * `1 <= n1, n2 <= 106`

Code Snippets

C++:

```
class Solution {
public:
int getMaxRepetitions(string s1, int n1, string s2, int n2) {

}
};
```

Java:

```
class Solution {
public int getMaxRepetitions(String s1, int n1, String s2, int n2) {

}
}
```

Python3:

```
class Solution:
def getMaxRepetitions(self, s1: str, n1: int, s2: str, n2: int) -> int:
```