Problem 3137: Minimum Number of Operations to Make Word K-Periodic

Problem Information

Difficulty: Medium
Acceptance Rate: 60.31%
Paid Only: No
Tags: Hash Table, String, Counting

Problem Description

You are given a string `word` of size `n`, and an integer `k` such that `k` divides `n`.

In one operation, you can pick any two indices `i` and `j`, that are divisible by `k`, then replace the substring of length `k` starting at `i` with the substring of length `k` starting at `j`. That is, replace the substring `word[i..i + k - 1]` with the substring `word[j..j + k - 1]`.

Return _the**minimum** number of operations required to make_ `word` _**k-periodic**_.

We say that `word` is **k-periodic** if there is some string `s` of length `k` such that `word` can be obtained by concatenating `s` an arbitrary number of times. For example, if `word == "ababab"`, then `word` is 2-periodic for `s = "ab"`.

Example 1:

Input: word = "leetcodeleet", k = 4

Output: 1

Explanation:

We can obtain a 4-periodic string by picking i = 4 and j = 0. After this operation, word becomes equal to "leetleetleet".

Example 2:

Input: word = "leetcoleet", k = 2

Output: 3

Explanation:

We can obtain a 2-periodic string by applying the operations in the table below.

i | j | word ---|---|--- 0 | 2 | etetcoleet 4 | 0 | etetetleet 6 | 0 | etetetetet

Constraints:

* `1 <= n == word.length <= 105` * `1 <= k <= word.length` * `k` divides `word.length`. * `word` consists only of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
int minimumOperationsToMakeKPeriodic(string word, int k) {

}
};
```

Java:

```
class Solution {
public int minimumOperationsToMakeKPeriodic(String word, int k) {

}
}
```

Python3:

```
class Solution:
def minimumOperationsToMakeKPeriodic(self, word: str, k: int) -> int:
```