# Problem 1278: Palindrome Partitioning III

## Problem Information

**Difficulty:** Hard
**Acceptance Rate:** 61.98%
**Paid Only:** No
**Tags:** String, Dynamic Programming

## Problem Description

You are given a string `s` containing lowercase letters and an integer `k`. You need to :

* First, change some characters of `s` to other lowercase English letters. * Then divide `s` into `k` non-empty disjoint substrings such that each substring is a palindrome.

Return _the minimal number of characters that you need to change to divide the string_.

**Example 1:**

**Input:** s = "abc", k = 2 **Output:** 1 **Explanation:** You can split the string into "ab" and "c", and change 1 character in "ab" to make it palindrome.

**Example 2:**

**Input:** s = "aabbc", k = 3 **Output:** 0 **Explanation:** You can split the string into "aa", "bb" and "c", all of them are palindrome.

**Example 3:**

**Input:** s = "leetcode", k = 8 **Output:** 0

**Constraints:**

* `1 <= k <= s.length <= 100`. * `s` only contains lowercase English letters.

## Code Snippets

### C++:

```
class Solution {
public:
int palindromePartition(string s, int k) {

}
};
```

### Java:

```
class Solution {
public int palindromePartition(String s, int k) {

}
}
```

### Python3:

```
class Solution:
def palindromePartition(self, s: str, k: int) -> int:
```