Problem 2472: Maximum Number of Non-overlapping Palindrome Substrings

Problem Information

Difficulty: Hard
Acceptance Rate: 42.58%
Paid Only: No
Tags: Two Pointers, String, Dynamic Programming, Greedy

Problem Description

You are given a string `s` and a **positive** integer `k`.

Select a set of **non-overlapping** substrings from the string `s` that satisfy the following conditions:

* The **length** of each substring is **at least** `k`. * Each substring is a **palindrome**.

Return _the**maximum** number of substrings in an optimal selection_.

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

Example 1:

Input: s = "abaccdbbd", k = 3 **Output:** 2 **Explanation:** We can select the substrings underlined in s = "_**aba**_ cc _**dbbd**_ ". Both "aba" and "dbbd" are palindromes and have a length of at least k = 3. It can be shown that we cannot find a selection with more than two valid substrings.

Example 2:

Input: s = "adbcda", k = 2 **Output:** 0 **Explanation:** There is no palindrome substring of length at least 2 in the string.

Constraints:

* `1 <= k <= s.length <= 2000` * `s` consists of lowercase English letters.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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