

2472. Maximum Number of Non-overlapping Palindrome Substrings

[My Submissions \(/contest/weekly-contest-319/problems/maximum-number-of-non-overlapping-palindrome-substrings/submissions/\)](/contest/weekly-contest-319/problems/maximum-number-of-non-overlapping-palindrome-substrings/submissions/)

[Back to Contest \(/contest/weekly-contest-319/\)](/contest/weekly-contest-319/)

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.

User Accepted:	1883
User Tried:	3413
Total Accepted:	2068
Total Submissions:	8051
Difficulty:	Hard

Example 1:

**Input:** `s = "abaccdbbd"`, `k = 3`

**Output:** `2`

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

Example 2:

**Input:** `s = "adbcd"`, `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.

[Discuss \(https://leetcode.com/problems/maximum-number-of-non-overlapping-palindrome-substrings/discuss/\)](https://leetcode.com/problems/maximum-number-of-non-overlapping-palindrome-substrings/discuss/)

JavaScript



```
1 const initialize2DArray = (n, m) => { let d = []; for (let i = 0; i < n; i++) { let t = new Int32Array(m).fill(0); d.push(t); } return d; };
2
3 const countPalindromeSub = () => {
4   for (let i = 0; i < n; i++) {
5     isPal(i, i);
6     isPal(i, i + 1);
7   }
8 };
9
10 const isPal = (l, r) => {
11   while (l >= 0 && r < n && s[l] == s[r]) {
12     ok[l][r] = 1;
13     l--;
14     r++;
15   }
16 };
17
18 let s, n, ok;
19 const maxPalindromes = (S, k) => {
20   s = S, n = s.length, ok = initialize2DArray(n, n);
21   countPalindromeSub();
22   let dp = Array(n + 1).fill(0);
23   for (let i = 1; i <= n; i++) {
24     dp[i] = dp[i - 1];
```


```
25  for (let j = i - k; j >= 0; j--) {  
26      if (ok[j][i - 1]) dp[i] = Math.max(dp[i], dp[j] + 1);  
27  }  
28  }  
29  return dp[n];  
30  };
```

☐ Custom Testcase[Use Example Testcases](#)[Run](#)[Submit](#)Submission Result: **Accepted** (/submissions/detail/842515661/) ?[More Details](#) (/submissions/detail/842515661/)

Share your acceptance!

[◀ 1](#)

Copyright © 2022 LeetCode

[Help Center \(/support\)](#) | [Jobs \(/jobs\)](#) | [Bug Bounty \(/bugbounty\)](#) | [Online Interview \(/interview/\)](#) | [Students \(/student\)](#) | [Terms \(/terms\)](#) | [Privacy Policy \(/privacy\)](#) [United States \(/region\)](#)