

Problem 3472: Longest Palindromic Subsequence After at Most K Operations

Problem Information

Difficulty: Medium

Acceptance Rate: 37.08%

Paid Only: No

Tags: String, Dynamic Programming

Problem Description

You are given a string `s` and an integer `k`.

In one operation, you can replace the character at any position with the next or previous letter in the alphabet (wrapping around so that `'a'` is after `'z'`). For example, replacing `'a'` with the next letter results in `'b'`, and replacing `'a'` with the previous letter results in `'z'`. Similarly, replacing `'z'` with the next letter results in `'a'`, and replacing `'z'` with the previous letter results in `'y'`.

Return the length of the **longest palindromic subsequence** of `s` that can be obtained after performing **at most** `k` operations.

Example 1:

Input: `s = "abcd"`, `k = 2`

Output: 3

Explanation:

* Replace `s[1]` with the next letter, and `s` becomes `"acced"`. * Replace `s[4]` with the previous letter, and `s` becomes `"acce"`.

The subsequence `"ccc"` forms a palindrome of length 3, which is the maximum.

****Example 2:****

****Input:**** s = "aaazzz", k = 4

****Output:**** 6

****Explanation:****

* Replace `s[0]` with the previous letter, and `s` becomes `"zaazzz"`. * Replace `s[4]` with the next letter, and `s` becomes `"zaazaz"`. * Replace `s[3]` with the next letter, and `s` becomes `"zaaaaz"`.

The entire string forms a palindrome of length 6.

****Constraints:****

* `1 <= s.length <= 200` * `1 <= k <= 200` * `s` consists of only lowercase English letters.

Code Snippets

C++:

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

    }

};
```

Java:

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

    }

}
```

Python3:

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