Problem 1682: Longest Palindromic Subsequence II

Problem Information

Difficulty: Medium
Acceptance Rate: 50.35%
Paid Only: Yes
Tags: String, Dynamic Programming

Problem Description

A subsequence of a string `s` is considered a **good palindromic subsequence** if:

* It is a subsequence of `s`. * It is a palindrome (has the same value if reversed). * It has an **even** length. * No two consecutive characters are equal, except the two middle ones.

For example, if `s = "abcabcabb"`, then `"abba"` is considered a **good palindromic subsequence** , while `"bcb"` (not even length) and `"bbbb"` (has equal consecutive characters) are not.

Given a string `s`, return _the**length** of the **longest good palindromic subsequence** in _`s`.

Example 1:

Input: s = "bbabab" **Output:** 4 **Explanation:** The longest good palindromic subsequence of s is "baab".

Example 2:

Input: s = "dcbccacdb" **Output:** 4 **Explanation:** The longest good palindromic subsequence of s is "dccd".

Constraints:

* `1 <= s.length <= 250` * `s` consists of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
int longestPalindromeSubseq(string s) {

}
};
```

Java:

```
class Solution {
public int longestPalindromeSubseq(String s) {

}
}
```

Python3:

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