Problem 730: Count Different Palindromic Subsequences

Problem Information

Difficulty: Hard
Acceptance Rate: 47.24%
Paid Only: No
Tags: String, Dynamic Programming

Problem Description

Given a string s, return _the number of different non-empty palindromic subsequences in_ `s`. Since the answer may be very large, return it **modulo** `109 + 7`.

A subsequence of a string is obtained by deleting zero or more characters from the string.

A sequence is palindromic if it is equal to the sequence reversed.

Two sequences `a1, a2, ...` and `b1, b2, ...` are different if there is some `i` for which `ai != bi`.

Example 1:

Input: s = "bccb" **Output:** 6 **Explanation:** The 6 different non-empty palindromic subsequences are 'b', 'c', 'bb', 'cc', 'bcb', 'bccb'. Note that 'bcb' is counted only once, even though it occurs twice.

Example 2:

Input: s = "abcdabcdabcdabcdabcdabcdabcdabcddcbadcbadcbadcbadcbadcbadcbadcba" **Output:** 104860361 **Explanation:** There are 3104860382 different non-empty palindromic subsequences, which is 104860361 modulo 109 + 7.

Constraints:

* `1 <= s.length <= 1000` * `s[i]` is either `'a'`, `'b'`, `'c'`, or `'d'`.

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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