Problem 1745: Palindrome Partitioning IV

Problem Information

Difficulty: Hard
Acceptance Rate: 44.99%
Paid Only: No
Tags: String, Dynamic Programming

Problem Description

Given a string `s`, return `true` _if it is possible to split the string_ `s` _into three**non-empty** palindromic substrings. Otherwise, return _`false`.

A string is said to be palindrome if it the same string when reversed.

Example 1:

Input: s = "abcbdd" **Output:** true **Explanation:** "abcbdd" = "a" + "bcb" + "dd", and all three substrings are palindromes.

Example 2:

Input: s = "bcbddxy" **Output:** false **Explanation:** s cannot be split into 3 palindromes.

Constraints:

* `3 <= s.length <= 2000` * `s`■■■■■■ consists only of lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
bool checkPartitioning(string s) {
}
};
```

Java:

```
class Solution {
public boolean checkPartitioning(String s) {

}
}
```

Python3:

```
class Solution:
def checkPartitioning(self, s: str) -> bool:
```