Description | Editorial | Solutions | Submissions

1933. Check if String Is Decomposable Into Value-Equal Substrings

Solved

Premium

Easy | Topics | Hint

A **value-equal** string is a string where **all** characters are the same.

- For example, `"1111"` and `"33"` are value-equal strings.
- In contrast, `"123"` is not a value-equal string.

Given a digit string `s`, decompose the string into some number of **consecutive value-equal** substrings where **exactly one** substring has a **length of** `2` and the remaining substrings have a **length of** `3`.

Return `true` *if you can decompose* `s` *according to the above rules. Otherwise, return* `false`.

A **substring** is a contiguous sequence of characters in a string.

Example 1:

```
Input: s = "000111000"
Output: false
Explanation: s cannot be decomposed according to the rules because ["000", "111", "000"] does not have a substring of length 2.
```

Example 2:

```
Input: s = "00011111222"
Output: true
Explanation: s can be decomposed into ["000", "111", "11", "222"].
```

Example 3:

```
Input: s = "011100022233"
Output: false
Explanation: s cannot be decomposed according to the rules because of the first '0'.
```

Constraints:

- `1 <= s.length <= 1000`
- `s` consists of only digits `'0'` through `'9'`.

Seen this question in a real interview before? 1/5

Yes | No

Accepted 3.7K | Submissions 7.2K | Acceptance Rate 50.6%

Topics

Hint 1

Hint 2

Hint 3

Discussion (2)

57 | 2

Code

Python3 • Auto

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

        prev = s[0]
        count = 1
        flag = True

        for i in range(1, len(s)):
            if s[i] == prev:
                count += 1
                if count > 3:
                    count = 1
                    prev = s[i]
            else:
                if count == 1:
                    return False
                elif count == 2:
                    if flag:
                        flag = False
                    else:
                        return False

                count = 1
                prev = s[i]

        if count == 2:
            if flag:
                flag = False
            else:
                return False
        elif count > 3:
```

Saved

Ln 17, Col 33

Testcase | Test Result

Accepted Runtime: 63 ms

• Case 1 • Case 2 • Case 3

Input

s =