Minimum Number Of Changes To Make Binary String Beautiful

solved by	Senan
Platform	LeetCode
difficulty	Medium
# Serial	2914
tags	String Manipulation
language	C++
solved on	@05/11/2024
link	https://leetcode.com/problems/minimum-number-of-changes-to-make-binary-string-beautiful/description/
Completion	☑

Intuition

To make the given binary string beautiful, we need to ensure it can be split into one or more contiguous substrings, each of even length, where each substring contains only `0`s or only `1`s. The problem can be visualized as ensuring that each pair of consecutive characters in the string is the same. This means that we must modify the string in such a way that the resulting substrings are made of consistent bits (either all `0`s or all `1`s).

Approach

1. Iterate through the string in pairs (considering an even-length string, we can safely check consecutive characters).
2. Count the number of changes needed to make consecutive characters the same.
3. This can be done by iterating through the string with a step of 2 and checking if the character at the current index differs from the next character.
4. If they differ, increment the `changes` counter, as one change would be needed to make them the same.
5. Return the `changes` counter as the result, which will indicate the minimum number of modifications needed to make the string beautiful.

Complexity

Time Complexity:

- **O(n)**: The algorithm iterates through the string once, where `n` is the length of the string. Each operation within the loop runs in constant time.

Space Complexity:

- **O(1)**: The solution uses a constant amount of space, as we only need a single integer variable to keep track of the number of changes.

Code

```
class Solution {
public:
    int minChanges(string s) {
        int changes = 0;
        // Iterate through the string in pairs
        for(int i = 0; i < s.size() - 1; i += 2) {
            if(s[i] != s[i + 1]) {
                changes++;
            }
        }
        return changes;
    }
};
```