Minimum Number Of Swaps To Make The String Balanced

solved by	Senan
Platform	LeetCode
difficulty	Medium
# Serial	1963
tags	String Manipulation
language	C++
solved on	@08/10/2024
link	https://leetcode.com/problems/minimum-number-of-swaps-to-make-the-string-balanced/description/
Completion	☑

Intuition

The problem involves determining the minimum number of swaps needed to balance a string of square brackets. Each swap can only involve adjacent characters, and a balanced string means that each closing bracket `]` should have a corresponding preceding opening bracket `[`. The key observation here is to count how many unmatched closing brackets `]` exist and perform swaps accordingly to balance the brackets.

Approach

1. Traverse the string character by character.
2. Maintain a counter `closing` that tracks how many more closing brackets `]` we need to balance the opening brackets `[`.
3. If we encounter a closing bracket `]` and no unmatched opening brackets `[`, a swap is required to balance it with an opening bracket. Increment both the swap count and the `closing` count.
4. If the `closing` count is greater than 0 and we encounter an opening bracket `[`, we decrement the `closing` counter to signify that we have balanced an unmatched closing bracket.
5. At the end of the traversal, the `swaps` variable holds the minimum number of swaps required to balance the string.

Complexity

Time Complexity:

- The solution iterates over the string exactly once, so the time complexity is **O(n)**, where `n` is the length of the string.

Space Complexity:

- The space complexity is **O(1)** because we are using a constant amount of extra space (for `swaps` and `closing` variables), regardless of the input size.

Code

```
class Solution {
public:
    int minSwaps(string s) {
        int swaps = 0;
        int closing = 0;

        for(auto ch : s){
            if(ch == ']'){
                if(closing == 0) swaps++, closing++;
                else closing--;
            }
            else closing++;
        }
        return swaps;
    }
};
```