

1963. Minimum Number of Swaps to Make the String Balanced

You are given a **0-indexed** string S of **even** length n . The string consists of **exactly** $n / 2$ opening brackets '[' and $n / 2$ closing brackets ']'

A string is called **balanced** if and only if:

- It is the empty string, or
- It can be written as AB , where both A and B are **balanced** strings, or
- It can be written as $[C]$, where C is a **balanced** string.

You may swap the brackets at **any** two indices **any** number of times.

Return the *minimum* number of swaps to make S **balanced**.

Example 1:

Input: $s = "][]["$

Output: 1

Explanation: You can make the string balanced by swapping index 0 with index 3. The resulting string is "[[]]".

Example 2:

Input: $s = "]]][[["$

Output: 2

Explanation: You can do the following to make the string balanced:

- Swap index 0 with index 4. $s = "[[]][["$.
- Swap index 1 with index 5. $s = "[[]][[]]"$.

The resulting string is "[[]][[]]".

Example 3:

Input: $s = "[]"$

Output: 0

Explanation: The string is already balanced.

Constraints:

- $n == s.length$
- $2 \leq n \leq 10^6$
- n is even.
- $s[i]$ is either '[' or ']'.
- The number of opening brackets '[' equals $n / 2$, and the number of closing brackets ']' equals $n / 2$.

1963. Minimum Number of Swaps to Make the String Balanced

```
/*  
    Stack  
    Time complexity: O(n)  
    Space complexity: O(n)  
*/  
class Solution {  
    public:  
        int minSwaps(std::string s) {  
            std::stack<char> st;  
            int ans=0;  
            for (char& c: s){  
                if (c=='[') st.push(c);  
                else{  
                    if(!st.empty()) st.pop();  
                    else ans++;  
                }  
            }  
            return (ans+1)/2;  
        }  
};
```

1963. Minimum Number of Swaps to Make the String Balanced

```
/*  
    Space optimization  
    Time complexity: O(n)  
    Space complexity: O(1)  
*/  
  
class Solution {  
    public:  
        int minSwaps(std::string s) {  
            int ans=0;  
            for(char& c: s) {  
                if (c=='[') ans++;  
                else if (ans>0) ans--;  
            }  
            return (ans+1)/2;  
        }  
};
```