

Problem 2696: Minimum String Length After Removing Substrings

Problem Information

Difficulty: Easy

Acceptance Rate: 77.08%

Paid Only: No

Tags: String, Stack, Simulation

Problem Description

You are given a string `s` consisting only of **uppercase** English letters.

You can apply some operations to this string where, in one operation, you can remove **any** occurrence of one of the substrings `"AB"` or `"CD"` from `s`.

Return `_the` **minimum** possible length of the resulting string that you can obtain.

Note that the string concatenates after removing the substring and could produce new `"AB"` or `"CD"` substrings.

Example 1:

Input: `s = "ABFCACDB"` **Output:** `2` **Explanation:** We can do the following operations:
- Remove the substring `"_AB_ FCACDB"`, so `s = "FCACDB"`.
- Remove the substring `"FCA _CD_ B"`, so `s = "FCAB"`.
- Remove the substring `"FC _AB_ "`, so `s = "FC"`.
So the resulting length of the string is 2. It can be shown that it is the minimum length that we can obtain.

Example 2:

Input: `s = "ACBBD"` **Output:** `5` **Explanation:** We cannot do any operations on the string so the length remains the same.

Constraints:

*`1 <= s.length <= 100` *`s` consists only of uppercase English letters.

Code Snippets

C++:

```
class Solution {  
public:  
    int minLength(string s) {  
  
    }  
};
```

Java:

```
class Solution {  
    public int minLength(String s) {  
  
    }  
}
```

Python3:

```
class Solution:  
    def minLength(self, s: str) -> int:
```