Problem 2645: Minimum Additions to Make Valid String

Problem Information

Difficulty: Medium
Acceptance Rate: 50.70%
Paid Only: No
Tags: String, Dynamic Programming, Stack, Greedy

Problem Description

Given a string `word` to which you can insert letters "a", "b" or "c" anywhere and any number of times, return _the minimum number of letters that must be inserted so that`word` becomes **valid**._

A string is called **valid** if it can be formed by concatenating the string "abc" several times.

Example 1:

Input: word = "b" **Output:** 2 **Explanation:** Insert the letter "a" right before "b", and the letter "c" right next to "b" to obtain the valid string "**a** b**c** ".

Example 2:

Input: word = "aaa" **Output:** 6 **Explanation:** Insert letters "b" and "c" next to each "a" to obtain the valid string "a**bc** a**bc** a**bc** ".

Example 3:

Input: word = "abc" **Output:** 0 **Explanation:** word is already valid. No modifications are needed.

Constraints:

* `1 <= word.length <= 50` * `word` consists of letters "a", "b" and "c" only.

Code Snippets

C++:

```
class Solution {
public:
int addMinimum(string word) {

}
};
```

Java:

```
class Solution {
public int addMinimum(String word) {

}
}
```

Python3:

```
class Solution:
def addMinimum(self, word: str) -> int:
```