Problem 2027: Minimum Moves to Convert String

Problem Information

Difficulty: Easy
Acceptance Rate: 57.27%
Paid Only: No
Tags: String, Greedy

Problem Description

You are given a string `s` consisting of `n` characters which are either `'X'` or `'O'`.

A **move** is defined as selecting **three** **consecutive characters** of `s` and converting them to `'O'`. Note that if a move is applied to the character `'O'`, it will stay the **same**.

Return _the**minimum** number of moves required so that all the characters of _`s` _are converted to_`'O'`.

Example 1:

Input: s = "XXX" **Output:** 1 **Explanation:** _XXX_ -> OOO We select all the 3 characters and convert them in one move.

Example 2:

Input: s = "XXOX" **Output:** 2 **Explanation:** _XXO_ X -> O _OOX_ -> OOOO We select the first 3 characters in the first move, and convert them to 'O'. Then we select the last 3 characters and convert them so that the final string contains all 'O's.

Example 3:

Input: s = "OOOO" **Output:** 0 **Explanation:** There are no 'X's in s to convert.

Constraints:

* `3 <= s.length <= 1000` * `s[i]` is either `'X'` or `'O'`.

Code Snippets

C++:

```
class Solution {
public:
int minimumMoves(string s) {

}
};
```

Java:

```
class Solution {
public int minimumMoves(String s) {

}
}
```

Python3:

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