1544. Make The String Great

Easy   ⬦ Topics   🔒 Companies   💡 Hint

Given a string `s` of lower and upper case English letters.

A good string is a string which doesn't have **two adjacent characters** `s[i]` and `s[i + 1]` where:

- `0 <= i <= s.length - 2`

- `s[i]` is a lower-case letter and `s[i + 1]` is the same letter but in upper-case or vice-versa.

To make the string good, you can choose **two adjacent** characters that make the string bad and remove them. You can keep doing this until the string becomes good.

Return *the string* after making it good. The answer is guaranteed to be unique under the given constraints.

Notice that an empty string is also good.

```
class Solution {
    func makeGood(_ s: String) -> String {
        var stack: [Character] = []
        for c in s {
            if stack.isEmpty {
                stack.append(c)
            } else {
                if(stack.last?.uppercased() == String(c) &&
stack.last!.isLowercase && c.isUppercase) {
                    stack.popLast()
                }
                else if(stack.last?.lowercased() == String(c) &&
stack.last!.isUppercase && c.isLowercase) {
                    stack.popLast()
                } else {
                    stack.append(c)
                }
            }
        }
        return String(stack)
    }
}
```