[LeetCode](https://leetcode.com/problems/make-the-string-great/description/?envType=daily-question&envId=2024-04-05)

<https://github.com/AdityaKonda6/-50DaysOfCoding>

<https://leetcode.com/problems/make-the-string-great/description/?envType=daily-question&envId=2024-04-05>

<https://www.linkedin.com/in/aditya-adi-konda/>

Day 46 of [#50dayscodingchallenge](https://www.linkedin.com/feed/hashtag/?keywords=50dayscodingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633):  
[#leetcode](https://www.linkedin.com/feed/hashtag/?keywords=leetcode&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodechallenge](https://www.linkedin.com/feed/hashtag/?keywords=leetcodechallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcodestreak](https://www.linkedin.com/feed/hashtag/?keywords=leetcodestreak&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode2024](https://www.linkedin.com/feed/hashtag/?keywords=leetcode2024&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#leetcode50day](https://www.linkedin.com/feed/hashtag/?keywords=leetcode50day&highlightedUpdateUrns=urn:li:activity:7166316239483461633)  
   
Ventured further into my coding journey today, tackling the engaging LeetCode Problem "Successfully solved LeetCode Problem ���"

“1544. Make The String Great.”

   
✨ Task: 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.

Examples:

Example 1:

Input: s = "leEeetcode"

Output: "leetcode"

Explanation: In the first step, either you choose i = 1 or i = 2, both will result "leEeetcode" to be reduced to "leetcode".

Example 2:

Input: s = "abBAcC"

Output: ""

Explanation: We have many possible scenarios, and all lead to the same answer. For example:

"abBAcC" --> "aAcC" --> "cC" --> ""

"abBAcC" --> "abBA" --> "aA" --> ""

Example 3:

Input: s = "s"

Output: "s"

Let's Connect:

If you find this problem intriguing or have insights to share, let's connect! I'm passionate about problem-solving, algorithmic thinking, and collaborative learning. Feel free to comment or reach out for engaging discussions and knowledge exchange.Unravel the mystery using your coding skills!

[#CodingChallenge](https://www.linkedin.com/feed/hashtag/?keywords=codingchallenge&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#Algorithm](https://www.linkedin.com/feed/hashtag/?keywords=algorithm&highlightedUpdateUrns=urn:li:activity:7166316239483461633) [#LinkedInPost](https://www.linkedin.com/feed/hashtag/?keywords=linkedinpost&highlightedUpdateUrns=urn:li:activity:7166316239483461633) #Algorithm #Optimization #DataStructures #CodingChallenge  
  
Excited about the progress and challenges ahead!  
   
Make Sure You Follow My GitHub For Solutions: <https://github.com/AdityaKonda6/-50DaysOfCoding>  
  
  
Happy coding!

**Solution:-**

class Solution {

  public String makeGood(String s) {

    StringBuilder sb = new StringBuilder();

    for (char c : s.toCharArray())

      if (sb.length() > 0 && isBadPair(sb.charAt(sb.length() - 1), c))

        sb.deleteCharAt(sb.length() - 1);

      else

        sb.append(c);

    return sb.toString();

  }

  private boolean isBadPair(char a, char b) {

    return a != b && Character.toLowerCase(a) == Character.toLowerCase(b);

  }

}

