213. String Compression

Implement a method to perform basic string compression using the counts of repeated characters. For example, the string aabcccccaaa would become a2b1c5a3.

If the "compressed" string would not become smaller than the original string, your method should return the original string.

You can assume the string has only upper and lower case letters (a-z).

Example

**Example 1:**

Input: str = "aabcccccaaa"

Output: "a2b1c5a3"

**Example 2:**

Input: str = "aabbcc"

Output: "aabbcc"

Input test data (one parameter per line)How to understand a testcase?

<https://www.lintcode.com/problem/string-compression/description>

1. */\**
2. *\* To change this license header, choose License Headers in Project Properties.*
3. *\* To change this template file, choose Tools | Templates*
4. *\* and open the template in the editor.*
5. *\*/*
6. **package** javaapplication63;
8. ***/\*\****
9. ***\****
10. ***\* @author Usuario***
11. ***\*/***
12. **public** **class** JavaApplication63 {
14. **public** String compress(String originalString) {
15. *// write your code here*
16. String ans = "";
17. **char** actual = originalString.charAt(0);
18. **int** cont = 1;
19. **for** (**int** i = 1; i < originalString.length(); i++)
20. {
21. **if** (originalString.charAt(i) == actual)
22. {
23. cont++;
24. }
25. **else**
26. {
27. ans += actual + "" + cont;
28. actual = originalString.charAt(i);
29. cont = 1;
30. }
31. }
33. **if**(cont > 0)
34. {
35. ans += actual + "" + cont;
36. }
38. **if**(ans.length() < originalString.length()) {
39. **return** ans;
40. }
42. **return** originalString;
44. }
46. **public** **static** **void** main(String[] args) {
47. *// TODO code application logic here*
48. }
50. }