**难度: 简单**

字符串压缩。利用字符重复出现的次数，编写一种方法，实现基本的字符串压缩功能。比如，字符串aabcccccaaa会变为a2b1c5a3。若“压缩”后的字符串没有变短，则返回原先的字符串。你可以假设字符串中只包含大小写英文字母（a至z）。

示例1:

输入："aabcccccaaa"

输出："a2b1c5a3"

示例2:

输入："abbccd"

输出："abbccd"

解释："abbccd"压缩后为"a1b2c2d1"，比原字符串长度更长。

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/compress-string-lcci

著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。

**思路1:**

1). 创建一个StringBuilder用来存储压缩后的

2). 定义三个变量,分别是 count(压缩的数量)、(index)开始的下标, (endIndex)最后的下标

3). 循环比较是否相同

1). 相同就把count加1

2). 不相同就把这个字符还有count分别加入StringBuilder里面

1). 然后把这个不同的下标给到 开始的下标index

4). 每次第二层的循环结束循环后, 就用最后的下标endIndex比较是否大于原本字符串的长度-1,

1). 成立就把字符和count分别加入StringBuilder里面,并且结束循环,

2) 不成立count 变成1

5). 比较压缩后的长度是否小于原本的长度

1). 小于返回压缩后的字符串

2). 不小于返回原本的字符串

|  |
| --- |
| public String compressString(String S) {  StringBuilder sb = new StringBuilder();  int count = 1; //记录最后一个相同的下标  int index = 0; //开始的下标  int endIndex = 0; //最后的下标  while(index < S.length()) {  for (endIndex = index + 1; endIndex < S.length(); endIndex++) {  if (S.charAt(index) == S.charAt(endIndex)) {  count++;  }else {  sb.append(S.charAt(index));  sb.append(count);  index = endIndex; //把最后一个相同的下标给到i  break;  }  }  if (endIndex > S.length() -1) {  sb.append(S.charAt(index));  sb.append(count);  break;  }  count = 1;  }  return sb.toString().length() < S.length() ? sb.toString() : S;  } |

**思路2(单循环使用双指针):**

1). 先比较一下当前数组的长度,如果是小于或等于2就直接返回, 因为即使2个都是一样的压缩也只能是两个. 例如: aa 压缩之后 a2

2). 创建一个StringBuilder用于存储压缩后的数据

3). 定义三个变量,分别是 count(压缩的数量)、(index)开始的下标, (nextIndex)下一个的下标

4). 循环比较

1) 如果nextIndex 小于等于len -1 并且 index的字符等于nextIndex下标的字符,就把count++ 和 nextIndex++

2). 不成立这个话就把当前index的字符和count放入到StringBuilder,并且把nextIndex的下标给到index 且nextIndex加1, count = 1

5). 比较压缩后的长度是否小于原本的长度

1). 小于返回压缩后的字符串

2). 不小于返回原本的字符串

|  |
| --- |
| public String compressString2(String S) {  int len = S.length();  if (len <= 2) {  return S;  }  StringBuilder sb = new StringBuilder();  int index = 0;  int nextIndex = 1;  int count = 1;  while (index < len){  if (nextIndex <= len -1 && S.charAt(index) == S.charAt(nextIndex)) {  nextIndex++;  count++;  }else {  sb.append(S.charAt(index));  sb.append(count);  index = nextIndex;  nextIndex++;  count = 1;  }  }  return sb.toString().length() < len ? sb.toString() : S;  } |