**难度: 简单**

给你一个由大小写英文字母组成的字符串 s 。

一个整理好的字符串中，两个相邻字符 s[i] 和 s[i+1]，其中 0<= i <= s.length-2 ，要满足如下条件:

若 s[i] 是小写字符，则 s[i+1] 不可以是相同的大写字符。

若 s[i] 是大写字符，则 s[i+1] 不可以是相同的小写字符。

请你将字符串整理好，每次你都可以从字符串中选出满足上述条件的 两个相邻 字符并删除，直到字符串整理好为止。

请返回整理好的 字符串 。题目保证在给出的约束条件下，测试样例对应的答案是唯一的。

注意：空字符串也属于整理好的字符串，尽管其中没有任何字符。

示例 1：

输入：s = "leEeetcode"

输出："leetcode"

解释：无论你第一次选的是 i = 1 还是 i = 2，都会使 "leEeetcode" 缩减为 "leetcode" 。

示例 2：

输入：s = "abBAcC"

输出：""

解释：存在多种不同情况，但所有的情况都会导致相同的结果。例如：

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

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

示例 3：

输入：s = "s"

输出："s"

提示：

1 <= s.length <= 100

s 只包含小写和大写英文字母

来源：力扣（LeetCode）

链接：https://leetcode-cn.com/problems/make-the-string-great

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(使用栈):**

1). 使用栈先进后出的特点,每次比较栈顶的值是否与需要入栈的值,大小字母是否相同

2). 相同直接退栈,不同则直接入栈

3). 最后把栈中剩余的值,全部退栈到StringBuilder里面,然后颠倒顺序,返回String的数据类型

|  |
| --- |
| public String makeGood(String s) {  if (s.length() == 0 || s.length() == 1) {  return s;  }  Deque<Character> stack = new ArrayDeque<>();  char[] chars = s.toCharArray();  for (int i = 0; i < chars.length; i++) {  if (stack.isEmpty() || (stack.peek() != chars[i] - 32 && stack.peek() != chars[i]+32) ){  stack.push(chars[i]);  }else if (stack.peek() == chars[i] - 32 || stack.peek() == chars[i]+32){  stack.pop();  }  }  StringBuilder sb = new StringBuilder();  while(!stack.isEmpty()) {  sb.append(stack.pop());  }  return sb.reverse().toString();  } |

**思路2(不使用栈,效率相对高):**

1).基本思路与思路1相同,只是不使用栈,而是只使用StringBuilder来进行操作

|  |
| --- |
| public String makeGood2(String s) {  StringBuilder sb = new StringBuilder();  for (int i = 0, top = -1; i < s.length(); i++) {  char c = s.charAt(i);  if (top < 0 || (sb.charAt(top) != c -32 && sb.charAt(top) != c +32)) {  sb.append(c);  top++;  }else {  sb.deleteCharAt(top);  top--;  }  }  return sb.toString();  } |