

## 最长的美好子字符串

当一个字符串  $s$  包含的每一种字母的大写和小写形式 **同时** 出现在  $s$  中，就称这个字符串  $s$  是 **美好** 字符串。比方说，"abABB" 是美好字符串，因为 'A' 和 'a' 同时出现了，且 'B' 和 'b' 也同时出现了。然而，"abA" 不是美好字符串因为 'b' 出现了，而 'B' 没有出现。

给你一个字符串  $s$ ，请你返回  $s$  最长的 **美好子字符串**。如果有多个答案，请你返回 **最早** 出现的一个。如果不存在美好子字符串，请你返回一个空字符串。

**示例 1：**

**输入：**  $s = \text{"YazaAay"}$

**输出：**  $\text{"aAa"}$

**解释：** "aAa" 是一个美好字符串，因为这个子串中仅含一种字母，其小写形式 'a' 和大写形式 'A' 也同时出现了。

"aAa" 是最长的美好子字符串。

**示例 2：**

**输入：**  $s = \text{"Bb"}$

**输出：**  $\text{"Bb"}$

**解释：** "Bb" 是美好字符串，因为 'B' 和 'b' 都出现了。整个字符串也是原字符串的子字符串。

**示例 3：**

**输入：**  $s = \text{"c"}$

**输出：**  $\text{""}$

**解释：** 没有美好子字符串。

**示例 4：**

**输入：**  $s = \text{"dDzeE"}$

**输出：**  $\text{"dD"}$

**解释：** "dD" 和 "eE" 都是最长美好子字符串。

由于有多个美好子字符串，返回 "dD"，因为它出现得最早。

```
class Solution:
    def longestNiceSubstring(self, s: str) -> str:
```

```
def NiceSubstring(s):
    tmp_s = set(s)
    if len(tmp_s) % 2 == 0:
        for k in tmp_s:
            if (k.islower() and k.upper() in tmp_s) or (k.isupper()
and k.lower() in tmp_s):
                continue
            else:
                return False
        return True
    else:
        return False

if len(s) < 2: return ''
ans = ''
for i in range(len(s)-1):
    for j in range(i+2, len(s)+1):
        if NiceSubstring(s[i:j]) and j-i > len(ans):
            ans = s[i:j]
return ans
```