# 题目

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 = "YazaAay"

输出："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 = "Bb"

输出："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 = "c"

输出：""

解释：没有美好子字符串。

示例 4：

输入：s = "dDzeE"

输出："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。

提示：

1 <= s.length <= 100

s 只包含大写和小写英文字母。

# 分析

## 方法一：枚举

思路：

题目要求找到最长的美好子字符串，题目中给定的字符串s的长度length范围为1≤length≤100。由于字符串的长度比较小，因此可以枚举所有可能的子字符串，然后检测该字符串是否为美好的字符串，并得到长度最长的美好字符串。

题目关于美好字符串的定义为: 字符串中的每个字母的大写和小写形式同时出现在该字符串中。检测时，由于英文字母 ‘a’−‘z’最多只有26个, 因此可以利用二进制位来进行标记，lower标记字符中出现过小写英文字母，upper标记字符中出现过大写英文字母。如果满足 lower=upper，我们则认为字符串中所有的字符都满足大小写形式同时出现，则认定该字符串为美好字符串。

题目要求如果有多个答案，返回在字符串中最早出现的那个。此时，只需要首先检测从以字符串索引0为起始的子字符串。

代码：

class Solution {

public:

string longestNiceSubstring(string s) {

int maxPos = 0;

int maxLen = 0;

for (int i = 0; i < s.size(); i++) {

int lower = 0;

int upper = 0;

for (int j = i; j < s.size(); j++) {

if (islower(s[j])) {

lower |= 1 << (s[j] - 'a');

} else {

upper |= 1 << (s[j] - 'A');

}

if (lower == upper && j - i + 1 > maxLen) {

maxPos = i;

maxLen = j - i + 1;

}

}

}

return s.substr(maxPos, maxLen);

}

};

复杂度分析：

时间复杂度：O(n^2)，其中n为字符串的长度。需要枚举所有可能的子字符串，因此需要双重循环遍历字符串，总共可能有n^2个连续的子字符串。

空间复杂度：O(1)。由于返回值不需要计算空间复杂度，除了需要两个整数变量用来标记以外不需要额外的空间。

## 方法二：滑动窗口