Problem 1763: Longest Nice Substring

Problem Information

Difficulty: Easy
Acceptance Rate: 63.20%
Paid Only: No
Tags: Hash Table, String, Divide and Conquer, Bit Manipulation, Sliding Window

Problem Description

A string `s` is **nice** if, for every letter of the alphabet that `s` contains, it appears **both** in uppercase and lowercase. For example, `"abABB"` is nice because `'A'` and `'a'` appear, and `'B'` and `'b'` appear. However, `"abA"` is not because `'b'` appears, but `'B'` does not.

Given a string `s`, return _the longest**substring** of `s` that is **nice**. If there are multiple, return the substring of the **earliest** occurrence. If there are none, return an empty string_.

Example 1:

Input: s = "YazaAay" **Output:** "aAa" **Explanation:** "aAa" is a nice string because 'A/a' is the only letter of the alphabet in s, and both 'A' and 'a' appear. "aAa" is the longest nice substring.

Example 2:

Input: s = "Bb" **Output:** "Bb" **Explanation:** "Bb" is a nice string because both 'B' and 'b' appear. The whole string is a substring.

Example 3:

Input: s = "c" **Output:** "" **Explanation:** There are no nice substrings.

Constraints:

* `1 <= s.length <= 100` * `s` consists of uppercase and lowercase English letters.

Code Snippets

C++:

```
class Solution {
public:
string longestNiceSubstring(string s) {

}
};
```

Java:

```
class Solution {
public String longestNiceSubstring(String s) {

}
}
```

Python3:

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