Problem 1044: Longest Duplicate Substring

Problem Information

Difficulty: Hard
Acceptance Rate: 30.97%
Paid Only: No
Tags: String, Binary Search, Sliding Window, Rolling Hash, Suffix Array, Hash Function

Problem Description

Given a string `s`, consider all _duplicated substrings_ : (contiguous) substrings of s that occur 2 or more times. The occurrences may overlap.

Return **any** duplicated substring that has the longest possible length. If `s` does not have a duplicated substring, the answer is `""`.

Example 1:

Input: s = "banana" **Output:** "ana"

Example 2:

Input: s = "abcd" **Output:** ""

Constraints:

* `2 <= s.length <= 3 * 104` * `s` consists of lowercase English letters.

Code Snippets

C++:

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


}
};
```

Java:

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

}
}
```

Python3:

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