Problem 395: Longest Substring with At Least K Repeating Characters

Problem Information

Difficulty: Medium
Acceptance Rate: 45.81%
Paid Only: No
Tags: Hash Table, String, Divide and Conquer, Sliding Window

Problem Description

Given a string `s` and an integer `k`, return _the length of the longest substring of_ `s` _such that the frequency of each character in this substring is greater than or equal to_ `k`.

if no such substring exists, return 0.

Example 1:

Input: s = "aaabb", k = 3 **Output:** 3 **Explanation:** The longest substring is "aaa", as 'a' is repeated 3 times.

Example 2:

Input: s = "ababbc", k = 2 **Output:** 5 **Explanation:** The longest substring is "ababb", as 'a' is repeated 2 times and 'b' is repeated 3 times.

Constraints:

* `1 <= s.length <= 104` * `s` consists of only lowercase English letters. * `1 <= k <= 105`

Code Snippets

C++:

```
class Solution {
public:
int longestSubstring(string s, int k) {

}
};
```

Java:

```
class Solution {
public int longestSubstring(String s, int k) {

}
}
```

Python3:

```
class Solution:
def longestSubstring(self, s: str, k: int) -> int:
```