Problem 3662: Filter Characters by Frequency

Problem Information

Difficulty: Easy
Acceptance Rate: 87.24%
Paid Only: Yes
Tags: Hash Table, String, Counting

Problem Description

You are given a string `s` consisting of lowercase English letters and an integer `k`.

Your task is to construct a new string that contains only those characters from `s` which appear **fewer** than `k` times in the entire string. The order of characters in the new string must be the **same** as their **order** in `s`.

Return the resulting string. If no characters qualify, return an empty string.

Note: **Every occurrence** of a character that occurs fewer than `k` times is kept.

Example 1:

Input: s = "aadbbcccca", k = 3

Output: "dbb"

Explanation:

Character frequencies in `s`:

* `'a'` appears 3 times * `'d'` appears 1 time * `'b'` appears 2 times * `'c'` appears 4 times

Only `'d'` and `'b'` appear fewer than 3 times. Preserving their order, the result is `"dbb"`.

Example 2:

Input: s = "xyz", k = 2

Output: "xyz"

Explanation:

All characters (`'x'`, `'y'`, `'z'`) appear exactly once, which is fewer than 2. Thus the whole string is returned.

Constraints:

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

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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