

Problem 3692: Majority Frequency Characters

Problem Information

Difficulty: Easy

Acceptance Rate: 67.25%

Paid Only: No

Tags: Hash Table, String, Counting

Problem Description

You are given a string `s` consisting of lowercase English letters.

The **frequency group** for a value `k` is the set of characters that appear exactly `k` times in `s`.

The **majority frequency group** is the frequency group that contains the largest number of **distinct** characters.

Return a string containing all characters in the majority frequency group, in **any** order. If two or more frequency groups tie for that largest size, pick the group whose frequency `k` is **larger**.

Example 1:

Input: `s = "aaabbbccddde"`

Output: `"ab"`

Explanation:

Frequency (k) | Distinct characters in group | Group size | Majority? ---|---|---|--- 4 | {d} | 1 | No 3 | {a, b} | 2 | **Yes** 2 | {c} | 1 | No 1 | {e} | 1 | No Both characters `'a'` and `'b'` share the same frequency 3, they are in the majority frequency group. `"ba"` is also a valid answer.

Example 2:

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

****Output:**** "abcd"

****Explanation:****

Frequency (k) | Distinct characters in group | Group size | Majority? ---|---|---|--- 1 | {a, b, c, d} | 4 | ****Yes**** All characters share the same frequency 1, they are all in the majority frequency group.

****Example 3:****

****Input:**** s = "pfpfgi"

****Output:**** "fp"

****Explanation:****

Frequency (k) | Distinct characters in group | Group size | Majority? ---|---|---|--- 2 | {p, f} | 2 | ****Yes**** 1 | {g, i} | 2 | No (tied size, lower frequency) Both characters `p` and `f` share the same frequency 2, they are in the majority frequency group. There is a tie in group size with frequency 1, but we pick the higher frequency: 2.

****Constraints:****

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

Code Snippets

C++:

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

    }

};
```

Java:

```
class Solution {  
    public String majorityFrequencyGroup(String s) {  
  
    }  
}
```

Python3:

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