Sherlock and the Valid String ★

Problem   Submissions   Leaderboard   Editorial 🔒

Sherlock considers a string to be valid if all characters of the string appear the same number of times. It is also valid if he can remove just 1 character at 1 index in the string, and the remaining characters will occur the same number of times. Given a string s, determine if it is valid. If so, return YES, otherwise return NO.

Example

$s = abc$

This is a valid string because frequencies are $\{a : 1, b : 1, c : 1\}$.

$s = abcc$

This is a valid string because we can remove one c and have 1 of each character in the remaining string.

$s = abccc$

This string is not valid as we can only remove 1 occurrence of c. That leaves character frequencies of $\{a : 1, b : 1, c : 2\}$.

Function Description

Complete the isValid function in the editor below.

isValid has the following parameter(s):

- string s: a string

Returns

- string: either YES or NO

Input Format

A single string s.

Constraints

- $1 \leq |s| \leq 10^5$
- Each character $s[i] \in ascii[a - z]$

Sample Input 0

    aabbcd

Sample Output 0

    NO

Explanation 0

Given $s = $ **"aabbcd"**, we would need to remove two characters, both c and d → aabb or a and b → abcd, to make it valid. We are limited to removing only one character, so s is invalid.

Sample Input 1

    aabbccddeefghi

Sample Output 1

    NO

Explanation 1

Frequency counts for the letters are as follows:

`{'a': 2, 'b': 2, 'c': 2, 'd': 2, 'e': 2, 'f': 1, 'g': 1, 'h': 1, 'i': 1}`

There are two ways to make the valid string:

- Remove **4** characters with a frequency of **1**: **{fghi}**.
- Remove **5** characters of frequency **2**: **{abcde}**.

Neither of these is an option.

Sample Input 2

    abcdefghhgfedecba

Sample Output 2

    YES

Explanation 2

All characters occur twice except for *e* which occurs **3** times. We can delete one instance of *e* to have a valid string.

Change Theme   JavaScript (Node.js) ⌄

```javascript
26
27    // Complete the isValid function below.
28    function isValid(s) {
29        let isValid = 'YES';
30        const cache = {};
31
32        for (let i = 0; i < s.length; i++) {
33            cache[s[i]] = (cache[s[i]]||0) + 1;
34        }
35
36        const frequency = Object.entries(cache).reduce((acc, curVal) => {
37            const [char, freq] = curVal;
38            if (!acc[freq]) acc[freq] = 0;
39            acc[freq]++;
40            return acc;
41        }, {});
42
43        const freqEntries = Object.entries(frequency);
44
45        if (freqEntries.length > 2) return 'NO';
46        if (freqEntries.length <= 1) return isValid;
47
48        let minFreq = Math.min(+freqEntries[0][0], +freqEntries[1][0]);
49        let maxFreq = Math.max(+freqEntries[0][0], +freqEntries[1][0]);
```

Line: 51 Col: 75

⬆ Upload Code as File   ☐ Test against custom input   **Run Code**   **Submit Code**

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ **Sample Test case 0**

           Input (stdin)   Download

✓ **Sample Test case 1**   1   aabbcd

⊘ Sample Test case 2

Your Output (stdout)

```
1    NO
```

Expected Output

Download

```
1    NO
```

Debug output

```
1    { '1': 2, '2': 2 } 1
```