

Problem 2399: Check Distances Between Same Letters

Problem Information

Difficulty: **Easy**

Acceptance Rate: 71.29%

Paid Only: No

Tags: Array, Hash Table, String

Problem Description

You are given a **0-indexed** string `s` consisting of only lowercase English letters, where each letter in `s` appears **exactly twice**. You are also given a **0-indexed** integer array `distance` of length `26`.

Each letter in the alphabet is numbered from `0` to `25` (i.e. `'a' -> 0`, `'b' -> 1`, `'c' -> 2`, ..., `'z' -> 25`).

In a **well-spaced** string, the number of letters between the two occurrences of the `i`th letter is `distance[i]`. If the `i`th letter does not appear in `s`, then `distance[i]` can be **ignored**.

Return `true` if `s` is a **well-spaced** string, otherwise return `false`.

Example 1:

Input: `s = "abaccb"`, `distance = [1,3,0,5,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]`

Output: `true` **Explanation:** - 'a' appears at indices 0 and 2 so it satisfies `distance[0] = 1`.
- 'b' appears at indices 1 and 5 so it satisfies `distance[1] = 3`. - 'c' appears at indices 3 and 4 so it satisfies `distance[2] = 0`. Note that `distance[3] = 5`, but since 'd' does not appear in `s`, it can be ignored. Return `true` because `s` is a well-spaced string.

Example 2:

Input: `s = "aa"`, `distance = [1,0]` **Output:** `false` **Explanation:** - 'a' appears at indices 0 and 1 so there are zero letters between them.

Because $\text{distance}[0] = 1$, s is not a well-spaced string.

****Constraints:****

* $2 \leq s.length \leq 52$ * s consists only of lowercase English letters. * Each letter appears in s exactly twice. * $\text{distance.length} == 26$ * $0 \leq \text{distance}[i] \leq 50$

Code Snippets

C++:

```
class Solution {
public:
    bool checkDistances(string s, vector<int>& distance) {

    }
};
```

Java:

```
class Solution {
    public boolean checkDistances(String s, int[] distance) {

    }
}
```

Python3:

```
class Solution:
    def checkDistances(self, s: str, distance: List[int]) -> bool:
```