Problem 3707: Equal Score Substrings

Problem Information

Difficulty: Easy
Acceptance Rate: 56.56%
Paid Only: No
Tags: String, Prefix Sum

Problem Description

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

The **score** of a string is the sum of the positions of its characters in the alphabet, where `'a' = 1`, `'b' = 2`, ..., `'z' = 26`.

Determine whether there exists an index `i` such that the string can be split into two **non-empty** ****substrings**** `s[0..i]` and `s[(i + 1)..(n - 1)]` that have **equal** scores.

Return `true` if such a split exists, otherwise return `false`.

Example 1:

Input: s = "adcb"

Output: true

Explanation:

Split at index `i = 1`:

* Left substring = `s[0..1] = "ad"` with `score = 1 + 4 = 5` * Right substring = `s[2..3] = "cb"` with `score = 3 + 2 = 5`

Both substrings have equal scores, so the output is `true`.

Example 2:

Input: s = "bace"

Output: false

Explanation:■■■■■■

■■■■■■■ No split produces equal scores, so the output is `false`.

Constraints:

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

Code Snippets

C++:

```
class Solution {
public:
bool scoreBalance(string s) {

}
};
```

Java:

```
class Solution {
public boolean scoreBalance(String s) {

}
}
```

Python3:

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