Problem 1003: Check If Word Is Valid After Substitutions

Problem Information

Difficulty: Medium
Acceptance Rate: 60.77%
Paid Only: No
Tags: String, Stack

Problem Description

Given a string `s`, determine if it is **valid**.

A string `s` is **valid** if, starting with an empty string `t = ""`, you can **transform**`t`**into**`s` after performing the following operation **any number of times** :

* Insert string `"abc"` into any position in `t`. More formally, `t` becomes `tleft + "abc" + tright`, where `t == tleft + tright`. Note that `tleft` and `tright` may be **empty**.

Return `true` _if_`s` _is a**valid** string, otherwise, return_ `false`.

Example 1:

Input: s = "aabcbc" **Output:** true **Explanation:** "" -> "_abc_ " -> "a _abc_ bc" Thus, "aabcbc" is valid.

Example 2:

Input: s = "abcabcababcc" **Output:** true **Explanation:** "" -> "_abc_ " -> "abc _abc_ " -> "abcabc _abc_ " -> "abcabcab _abc_ c" Thus, "abcabcababcc" is valid.

Example 3:

Input: s = "abccba" **Output:** false **Explanation:** It is impossible to get "abccba" using the operation.

Constraints:

* `1 <= s.length <= 2 * 104` * `s` consists of letters `'a'`, `'b'`, and `'c'`

Code Snippets

C++:

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

}
};
```

Java:

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

}
}
```

Python3:

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