# 题目

给你一个字符串 s ，请你判断它是否 有效 。

字符串 s 有效 需要满足：假设开始有一个空字符串 t = "" ，你可以执行 任意次 下述操作将 t 转换为 s ：

将字符串 "abc" 插入到 t 中的任意位置。形式上，t 变为 tleft + "abc" + tright，其中 t == tleft + tright 。注意，tleft 和 tright 可能为 空 。

如果字符串 s 有效，则返回 true；否则，返回 false。

示例 1：

输入：s = "aabcbc"

输出：true

解释：

"" -> "abc" -> "aabcbc"

因此，"aabcbc" 有效。

示例 2：

输入：s = "abcabcababcc"

输出：true

解释：

"" -> "abc" -> "abcabc" -> "abcabcabc" -> "abcabcababcc"

因此，"abcabcababcc" 有效。

示例 3：

输入：s = "abccba"

输出：false

解释：执行操作无法得到 "abccba" 。

提示：

1 <= s.length <= 2 \* 104

s 由字母 'a'、'b' 和 'c' 组成

# 分析

## 方法一：栈

遍历字符串s，将当前访问到的字符c压入栈 stk中，如果栈元素数目大于等于3且栈顶的3个元素依次等于‘a’、‘b’和‘c’，那么将这三个元素出栈。如果最后栈为空，则字符串s有效。

代码：

class Solution {

public:

bool isValid(string s) {

string stk;

for (auto c : s) {

stk.push\_back(c);

if (stk.size() >=3 && stk.substr(stk.size() - 3, 3) == "abc") {

stk.erase(stk.end() - 3, stk.end());

}

}

return stk.empty();

}

};

**复杂度分析：**

时间复杂度：O(n)，其中n是字符串s的长度。

空间复杂度：O(n)。栈需要占用O(n)的空间。