# 题目

给你三个字符串 s1、s2 和 s3。 你可以根据需要对这三个字符串执行以下操作 任意次数 。

在每次操作中，你可以选择其中一个长度至少为 2 的字符串 并删除其 最右位置上 的字符。

如果存在某种方法能够使这三个字符串相等，请返回使它们相等所需的 最小 操作次数；否则，返回 -1。

示例 1：

输入：s1 = "abc"，s2 = "abb"，s3 = "ab"

输出：2

解释：对 s1 和 s2 进行一次操作后，可以得到三个相等的字符串。

可以证明，不可能用少于两次操作使它们相等。

示例 2：

输入：s1 = "dac"，s2 = "bac"，s3 = "cac"

输出：-1

解释：因为 s1 和 s2 的最左位置上的字母不相等，所以无论进行多少次操作，它们都不可能相等。因此答案是 -1 。

提示：

1 <= s1.length, s2.length, s3.length <= 100

s1、s2 和 s3 仅由小写英文字母组成。

# 分析

## 方法一：最长公共前缀

思路：

设lcp为三个字符串的最长公共前缀的长度。

如果lcp=0，无法操作成一样的，返回−1。

否则返回三个字符串的长度之和，减去剩下的长度3⋅lcp。

代码：

class Solution {

public:

int findMinimumOperations(string s1, string s2, string s3) {

int n = min({s1.length(), s2.length(), s3.length()});

int i = 0;

while (i < n && s2[i] == s1[i] && s3[i] == s1[i]) {

i++;

}

return i == 0 ? -1 : s1.length() + s2.length() + s3.length() - i \* 3;

}

};