# 题目

给你一个仅由字符'0'和'1'组成的字符串s。一步操作中，你可以将任一'0'变成'1'，或者将'1'变成'0'。

交替字符串定义为：如果字符串中不存在相邻两个字符相等的情况，那么该字符串就是交替字符串。例如，字符串"010"是交替字符串，而字符串"0100"不是。

返回使s变成交替字符串所需的最少操作数。

示例 1：

输入：s = "0100"

输出：1

解释：如果将最后一个字符变为 '1' ，s 就变成 "0101" ，即符合交替字符串定义。

示例 2：

输入：s = "10"

输出：0

解释：s 已经是交替字符串。

示例 3：

输入：s = "1111"

输出：2

解释：需要 2 步操作得到 "0101" 或 "1010" 。

提示：

1 <= s.length <= 10^4

s[i] 是 '0' 或 '1'

# 分析

## 方法一：模拟

思路：

可以通过统计两种情况下需要操作的次数，然后取最小值作为结果。

1、假设交替字符串以'0'开头，那么需要将所有奇数位置上的字符变为'1'，偶数位置上的字符变为'0'。统计需要操作的次数。

2、假设交替字符串以'1'开头，那么需要将所有奇数位置上的字符变为'0'，偶数位置上的字符变为'1'。统计需要操作的次数。

3、返回以上两种情况下的最小操作次数。

代码：

class Solution {

public:

int minOperations(string s) {

int n = s.size();

int cnt1 = 0, cnt2 = 0;

for (int i = 0; i < n; ++i) {

if (s[i] != '0' + (i % 2)) {

++cnt1;

}

if (s[i] != '0' + ((i + 1) % 2)) {

++cnt2;

}

}

return min(cnt1, cnt2);

}

};