**Problem #1758: Minimum Number of Changes to Make Alternating Binary String**

<https://leetcode.com/problems/minimum-changes-to-make-alternating-binary-string/description/>

My Solution:

1. Consider a binary string starting with ‘0’ and count number of changes that need to be made using count\_start\_0 which is initialized to 0. Also consider a binary string starting with ‘1’ and count number of changes that need to be made with count\_start\_1 which is initialized to 0.
2. Traverse through the length of the string s.

If the index i is even and the character at that index in the string is ‘0’, then count\_start1 should be incremented by 1.

If the index I is even and the character at that index in the string is ‘1’, then count\_start0 should be incremented by 1.

If the index I is odd and the character at index I in the string is ‘1’, then count\_start1 should be incremented by 1.

If the index I is odd and the character at index I in the string is ‘0’, then count\_start0 should be incremented by 1.

1. Return minimum of count\_start0 and count\_start1.

class Solution:

def minOperations(self, s: str) -> int:

count\_start0 = 0 # starting with 0

count\_start1 = 0 # starting with 1

for i in range(len(s)):

if i % 2 == 0: # even

if s[i] == '0':

count\_start1 += 1

else:

count\_start0 += 1

else: # odd

if s[i] == '1':

count\_start1 += 1

else:

count\_start0 += 1

return min(count\_start0, count\_start1)