Problem # 859: Buddy Strings

<https://leetcode.com/problems/buddy-strings/>

My Solution:

<https://leetcode.com/problems/buddy-strings/discuss/891636/Simple-Python-3-Solution-Runtime-beats-75.76>

Simple Python 3 Solution -- Runtime beats 75.76%

1. If the length of the two strings A and B are not equal, return False
2. Create A\_dict and B\_dict with Counter from collections and if the dictionaries are not equal, return False
3. Iterate through the two strings and compare the values at each index. If they are different, count them.
4. If the count of the difference is 2, return true. This means that two of the characters have been swapped.
5. If the count of the difference is 0, and if the length of the dictionary are greater than 0 and maximum value in each dictionary is greater than 1, then return True. This accounts for the case where we can swap the identical values in A at two different indexes i and j. For example: A = "aa", B = "aa"
6. Return False otherwise.

from collections import Counter

class Solution:

def buddyStrings(self, A: str, B: str) -> bool:

if len(A) != len(B):

return(False)

A\_dict = Counter(A)

B\_dict = Counter(B)

if A\_dict != B\_dict:

return(False)

count = 0

for i in range(len(A)):

if A[i] != B[i]:

count += 1

if count == 2:

return(True)

if count == 0 and len(A\_dict.values()) > 0 and max(A\_dict.values()) > 1 and len(B\_dict.values()) > 0 and max(B\_dict.values()) > 1:

return(True)

return(False)