Problem #1007 : Minimum Domino Rotations for Equal Row (Medium)

<https://leetcode.com/problems/minimum-domino-rotations-for-equal-row/>

My Solution:

<https://leetcode.com/problems/minimum-domino-rotations-for-equal-row/discuss/901947/Simple-Python-3-Solution-Runtime-beats-97.54>

1. Get the length of array A and let it be n.

2. If length of array B is not n, return -1.

3. Initialize result with an empty list.

4. Numbers on the tiles are 1 through 6. So for each number from 1 through 6,

check Array A as follows:

(1) Initialize count to 0, i (the index) to 0 and a boolean done to False which will be set to True if the while loop is terminated when iterating through the array A[i]

(2) Check whether A[i] is not equal to the number; if so, check if B[i] is equal to the number and increment count. Otherwise we don’t want to continue checking and so set done to True.

(3) After iterating through A[i] if not done, then append the count to result list.

5. Numbers on the tiles are 1 through 6. So for each number from 1 through 6,

check Array B as follows:

(1) Initialize count to 0, i (the index) to 0 and a boolean done to False which will be set to True if the while loop is terminated when iterating through the array B[i]

(2) Check whether B[i] is not equal to the number; if so, check if A[i] is equal to the number and increment count. Otherwise we don’t want to continue checking and so set done to True.

(3) After iterating through B[i] if not done, then append the count to result list.

6. If length of result list is 0, then return -1 since it is not possible to change the tiles and get the required condition that either array A has all the tiles with the same value or array B has all the tiles with the same value.

Otherwise, return the minimum value of result list.

-----------------------------------------------------------------------

class Solution:

def minDominoRotations(self, A: List[int], B: List[int]) -> int:

n = len(A)

if len(B) != n:

return(-1)

result = []

for num in range(1, 7):

# Check Array A

count = 0

i = 0

done = False

while (i < n) and not done:

if A[i] != num:

if B[i] == num:

count += 1

else:

done = True

i += 1

if not done:

result.append(count)

# Check array B

count = 0

i = 0

done = False

while (i < n) and not done:

if B[i] != num:

if A[i] == num:

count += 1

else:

done = True

i += 1

if not done:

result.append(count)

if len(result) == 0:

return(-1)

return(min(result))