Problem # 605 : Can Place Flowers (Easy)

<https://leetcode.com/problems/can-place-flowers/>

My Solution:

<https://leetcode.com/problems/can-place-flowers/discuss/961189/Simple-Python-3-Solution-Runtime-beats-95.22>

1. If n = 0, there are no flowers to place. So, return True.

2. If the length of flowerbed is 1 and flowerbed at index 0 is 0, then return True. Otherwise return False.

3. Check whether n flowers can be placed. For this, count the number of ones in the flowerbed called count\_ones.

4. if length of flowerbed is even and number of flowers to be placed exceeds (half the length of the flowerbeds – count\_ones), then return False. Also if length of flowerbed is odd and number of flowers to be placed exceeds ((half the length of the flowerbeds) + 1 – count\_ones), then return False.

5. Initialize count to 0 where count represents the number of flowers that can be placed.

6. Then we iterate through the flowerbed array and check if a flower can be placed. If at index i, the flowerbed has a value 0, check if a flower can be placed at this position. If it is the beginning of the array, we only need to check the flowerbed at the next position is 0. Again if it is the end of the array, we only need to check the flowerbed at the previous position is 0. Otherwise we need to check whether both the next and the previous flowerbed values are 0, so that a flower can be placed there. At the end of each iteration, check if count has reached n. If so, return True.

7. After iterating through the whole array, if count has not reached n, then return False.

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

class Solution:

def canPlaceFlowers(self, flowerbed: List[int], n: int) -> bool:

if n == 0:

return True

m = len(flowerbed)

if m == 1:

if flowerbed[0] == 0:

return True

else:

return False

count\_ones = flowerbed.count(1)

if (m % 2 == 0 and n > (m//2) - count\_ones) or ( n > (m//2) + 1 - count\_ones):

return False

count = 0

for i in range(m):

if flowerbed[i] == 0:

if (i == 0 and flowerbed[i+1]== 0) or (i == m-1 and flowerbed[i-1] == 0) or (flowerbed[i-1] == 0 and flowerbed[i+1] == 0):

flowerbed[i] = 1

count += 1

if count == n:

return True

return False