Problem # 1492: The kth Factor of N (Medium)

<https://leetcode.com/problems/the-kth-factor-of-n/>

My Solution:

<https://leetcode.com/problems/the-kth-factor-of-n/discuss/960334/Simple-Python-3-Solution-Runtime-beats-95.02>

1. If k is greater than n, this is not possible. So return -1.
2. if n is 1, then return 1 (since k has to be 1)
3. Check if n is even or odd. Set is\_even to True if n is even and is\_even to False otherwise.
4. Initialize count to 0.
5. If n is odd, we need to check only odd factors in the range 1 to n//2 (both inclusive).  
   If we have a factor, increment count. Check if count is equal to k, and if so, return the factor.
6. if n is even, we need to check all factors in the range 1 to n//2 (both inclusive).  
   If we have a factor, increment count. Check if count is equal to k, and if so, return the factor.
7. We have n to be a factor of itself. So, increment count. Check if count is equal to k. If so, return n. Otherwise, return -1 (for kth factor not found).

class Solution:

def kthFactor(self, n: int, k: int) -> int:

if k > n:

return -1

if n == 1:

return 1

if n % 2 == 0:

is\_even = True

else:

is\_even = False

count = 0

if not is\_even:

for i in range(1, n//2 + 1, 2):

if n % i == 0:

count += 1

if count == k:

return i

else:

for i in range(1, n//2 + 1):

if n % i == 0:

count += 1

if count == k:

return i

# n is a factor of itself

count += 1

if count == k:

return n

return -1 # count < k