Problem # 1640: Check Array Formation Through Concatenation (Easy)

<https://leetcode.com/problems/check-array-formation-through-concatenation>

My Solution:

<https://leetcode.com/problems/check-array-formation-through-concatenation/discuss/997022/Simple-Python-3-Solution-Runtime-beats-99.65>

Runtime beats 99.5% of python submissions.

1. Let n be the length of array arr.
2. Now, use variable count to count the number of individual elements in sublists in pieces,  
   So, initialize count to 0.
3. Iterate through the pieces array.  
   For each piece in the array, find its length and add it to count.  
   If the first element of each piece is not present in arr, then return False.  
   If the piece has more than 1 element, get the index called idx of the element in arr.  
   Initialize j to the next index, i.e. j = idx + 1.  
   Iterate through the piece list and compare it with the elements of arr in consecutive order. If they do not match return False.
4. Check if count and n match, then return True if they match and False otherwise

class Solution:

def canFormArray(self, arr: List[int], pieces: List[List[int]]) -> bool:

n = len(arr)

count = 0

for piece in pieces:

count += len(piece)

if len(piece) < 1:

return False

elif len(piece) == 1:

if piece[0] not in arr:

return False

else: # len(piece) > 1

m = len(piece)

if piece[0] not in arr:

return False

idx = arr.index(piece[0])

j = idx + 1

if j >= n:

return False

for i in range(1, m):

if piece[i] != arr[j]:

return False

else:

if j <= n:

j += 1

else:

return False

return count == n