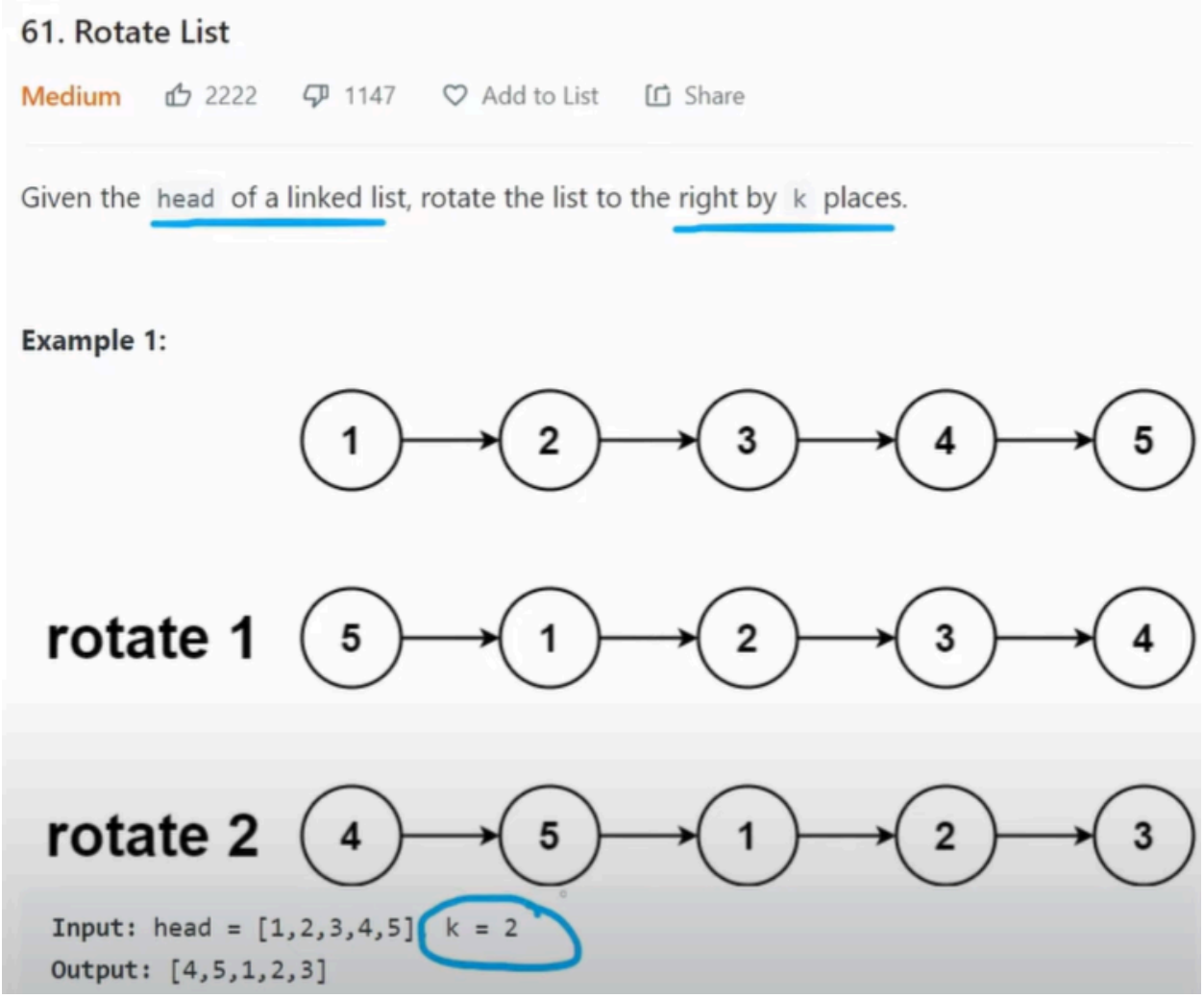
61. Rotate List

Initial thoughts:

1. 

2. To use two pointers and travel the 2nd pointer till k steps and then travel two pointers until the 2nd pointer.next is null

Python Code:

1.
```python
class Solution:
    def rotateRight(self, head: ListNode, k: int) -> ListNode:
        if not head:
            return head

        # Get length
        length, tail = 1, head
        while tail.next:
            tail = tail.next
            length += 1

        k = k % length
        if k == 0:
            return head

        # Move to the pivot and rotate
        cur = head
        for i in range(length - k - 1):
            cur = cur.next
        newHead = cur.next
        cur.next = None
        tail.next = head
        return newHead
```
2. Basically the approach is the same…to move the 2nd pointer k steps
3. But what if the k is larger than the length of the array? To solve this we have divided k with length to get the required number of rotations
4. Next we'll move our first pointer till "length - k -1"
5. Now our new head will be curr.next
6. See code and get the idea