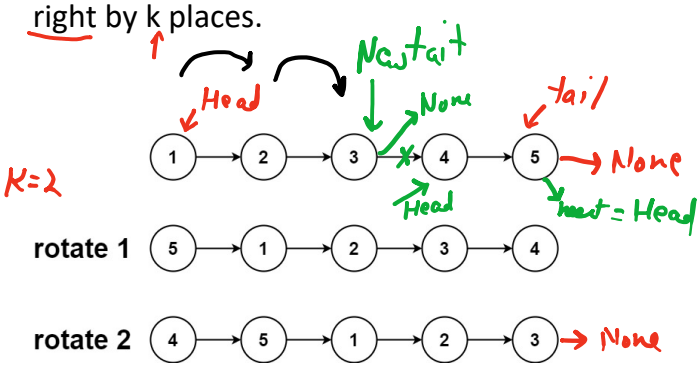
Rotate List (Leetcode 61)

Given the head of a linked list, rotate the list to the right by k places.



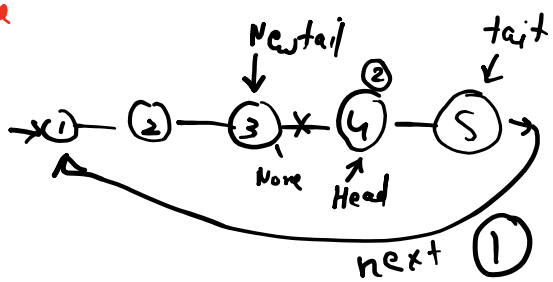
K=2
L= 5

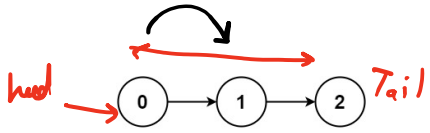
① Head ✓ Newtail = 5-2-1 = 2

② Tail ✓

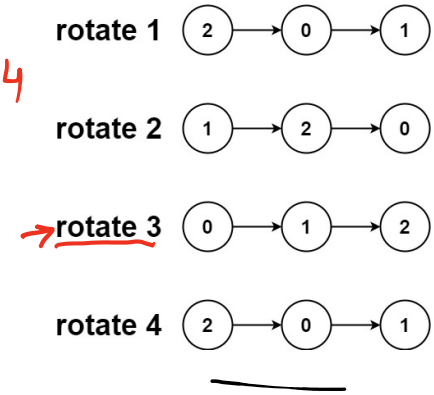
③ K ✓ → K % length

④ Length ✓





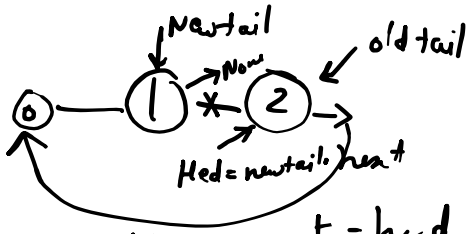
K=4

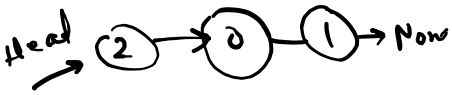


K = K % length → (length - k - 1) ← Jumps

4 % 3 = 1 = K

K = 1
L = 3
Newtail = 3 - 1 - 1 = 1





old tail. next = head

Base condition

Head ≠ None

(length = 1 ...

Length { loop → tail.next == None
length += 1

update { K = K % length

if K == 0:
    return head

& new_tail = head
loop → (length - K - 1)
    update new_tail = new_tail.next

→ new_head = new_tail.next
tail.next = head
new_tail.next = None

return new_head