

61. Rotate List

Solved ✓

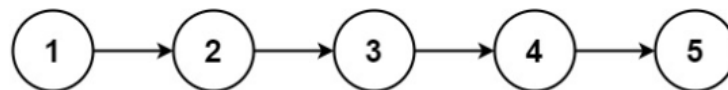
Medium

Topics

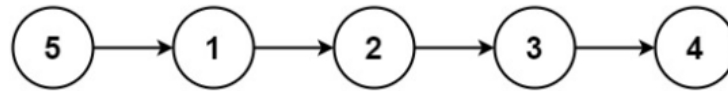
Companies

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

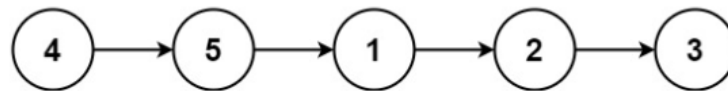
Example 1:



rotate 1



rotate 2



Input: `head = [1, 2, 3, 4, 5]` `k = 2`

Code

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     public var val: Int
 *     public var next: ListNode?
 *     public init() { self.val = 0; self.next = nil; }
 *     public init(_ val: Int) { self.val = val; self.next =
nil; }
 *     public init(_ val: Int, _ next: ListNode?) { self.val =
val; self.next = next; }
 * }
 */
class Solution {
    func rotateRight(_ head: ListNode?, _ k: Int) -> ListNode?
    {
        guard head?.next != nil else {
```

```
        return head
    }

    var temp = head
    var i = 1
    while(temp?.next != nil) {
        i = i + 1
        temp = temp?.next
    }

    // temp is last pointer

    temp?.next = head

    for _ in 0.. $(i - k \% i)$  {
        temp = temp?.next
    }

    let newHead = temp?.next
    temp?.next = nil

    return newHead
}
}
```