Problem 61: Rotate List

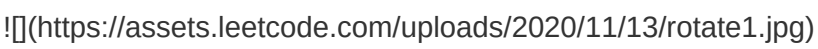
Problem Information

Difficulty: Medium
Acceptance Rate: 40.78%
Paid Only: No
Tags: Linked List, Two Pointers

Problem Description

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

Example 1:



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

Example 2:

Input: head = [0,1,2], k = 4 **Output:** [2,0,1]

Constraints:

* The number of nodes in the list is in the range `[0, 500]`. * `-100 <= Node.val <= 100` * `0 <= k <= 2 * 109`

Code Snippets

C++:

```
/**
* Definition for singly-linked list.
```

```
* struct ListNode {
* int val;
* ListNode *next;
* ListNode() : val(0), next(nullptr) {}
* ListNode(int x) : val(x), next(nullptr) {}
* ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
ListNode* rotateRight(ListNode* head, int k) {

}
};
```

Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
public ListNode rotateRight(ListNode head, int k) {

}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
# def __init__(self, val=0, next=None):
# self.val = val
# self.next = next
class Solution:
def rotateRight(self, head: Optional[ListNode], k: int) ->
```

```
Optional[ListNode]:
```