61. Rotate List

Description | Hints | Submissions | Discuss | Solution

Pick One

Given a linked list, rotate the list to the right by *k* places, where *k* is non-negative.

Example 1:

```
Input: 1->2->3->4->5->NULL, k = 2
Output: 4->5->1->2->3->NULL
Explanation:
rotate 1 steps to the right: 5->1->2->3->4->NULL
rotate 2 steps to the right: 4->5->1->2->3->NULL
```

Example 2:

```
Input: 0->1->2->NULL, k = 4
Output: 2->0->1->NULL
Explanation:
rotate 1 steps to the right: 2->0->1->NULL
rotate 2 steps to the right: 1->2->0->NULL
rotate 3 steps to the right: 0->1->2->NULL
rotate 4 steps to the right: 2->0->1->NULL
```

```java
public ListNode rotateRight(ListNode head, int k) {
        if(head == null || k == 0)
            return head;
        ListNode p = head;
        int len = 1;
        //获得整个链表的长度
        while (p.next != null) {
            len ++;
            p = p.next;
        }
        p.next = head; //回到头，重新开始
        k = k % len;
        for(int i = 0; i < len -k; i ++) {
            p = p.next;
        }
        //旋转链表
        head = p.next;
        p.next = null;
        return head;
}
```