

61. Rotate List

题目描述: <https://leetcode.com/problems/rotate-list/>

将一个链表循环右移k次。求移动之后的结果

解题思路:

1. 直接处理向右移动。先找到新的head然后拼接回去
2. 将右移转换成左移
3. 将头和尾吧串在一起然后head向后移动。

注意: 将移动多圈的转化成移动一圈以内的

代码V1.0:

```

/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* rotateRight(ListNode* head, int k) {
        if(k == 0 || head == NULL || head->next == NULL)
            return head;
        ListNode* fakeHead = new ListNode(0);
        fakeHead->next = head;
        ListNode* pre = fakeHead;
        ListNode* now = fakeHead;
        int len = 0;
        while(now->next){
            now = now->next;
            len++;
        }
        k = k % len;
        if(k == 0)
            return head;
        cout << k;
        now = fakeHead;
        for(int i = 0; now && i <= k; i++){
            now = now->next;
        }
        while(now){
            now = now->next;
            pre = pre->next;
        }
        ListNode* newHead = pre->next;
        pre->next = NULL;
        now = newHead;
        while(now->next){
            now = now->next;
        }
        now->next = fakeHead->next;
        fakeHead->next = newHead;
        return fakeHead->next;
    }
};

```

代码V2.0:

```

/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* rotateRight(ListNode* head, int k) {
        if(head == NULL || head->next == NULL)
            return head;
        ListNode* now = head;
        ListNode* fakeHead = new ListNode(0);
        fakeHead->next = head;
        int len = 1;
        while (now->next) {
            now = now->next;
            len++;
        }
        k = len - (k % len);
        while(k > 0){
            k--;
            ListNode *t = head->next;
            now->next = head;
            now = now->next;
            head->next = NULL;
            head = t;
        }
        return head;
    }
};

```

代码V3.0:

```

/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* rotateRight(ListNode* head, int k) {
        if(head == NULL || head->next == NULL) return head;
        ListNode* t = head;
        int len = 1;
        while(t->next) {
            len++;
            t = t->next;
        }
        t->next = head;
        k = k % len;
        k = len - k;
        while(k > 0) {
            k--;
            t = t->next;
        }
        head = t->next;
        t->next = NULL;
        return head;
    }
};

```