

Description

Solution

Discuss (999+)

Submissions

61. Rotate List

Medium

4872

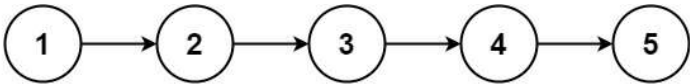
1263

Add to List

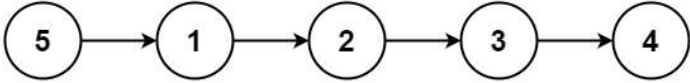
Share

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

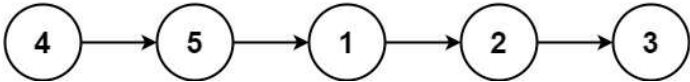
Example 1:



rotate 1

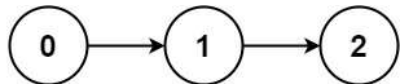


rotate 2

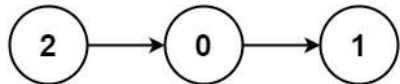


Input: head = [1,2,3,4,5], k = 2
Output: [4,5,1,2,3]

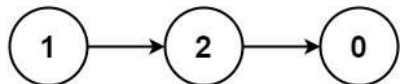
Example 2:



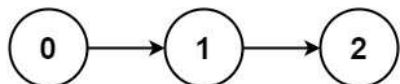
rotate 1



rotate 2



rotate 3



rotate 4



```
7     ListNode(int val) { this.val = val; }
8     *   ListNode(int val, ListNode next) { this.val = val;
        this.next = next; }
9     * }
10    */
11    class Solution {
12
13    public ListNode rotateRight( ListNode head, int k ) {
14
15        if (head == null || head.next == null)
16            return head;
17        else if (k == 0)
18            return head;
19
20        ListNode tempNode = head;
21        int listSizeCounter = 0;
22
23        while (tempNode != null) {
24            listSizeCounter++;
25            tempNode = tempNode.next;
26        }
27
28        k = k % listSizeCounter;
29
30        if(k == 0)
31            return head;
32
33        tempNode = head;
34
35        for (int i = 0; i < listSizeCounter - k - 1; i++)
36            tempNode = tempNode.next;
37
38        ListNode reverseStart = tempNode.next;
39
40        tempNode.next = null;
41        tempNode = reverseStart;
42
43        while (tempNode != null && tempNode.next != null) {
44            tempNode = tempNode.next;
45        }
46
47        tempNode.next = head;
48        head = reverseStart;
49    }
```