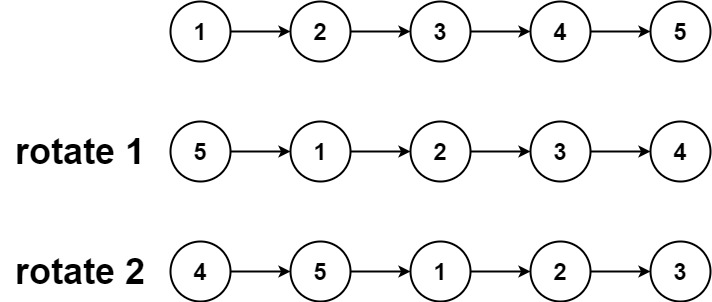
Given the head of a linked list, rotate the list to the right by k places.

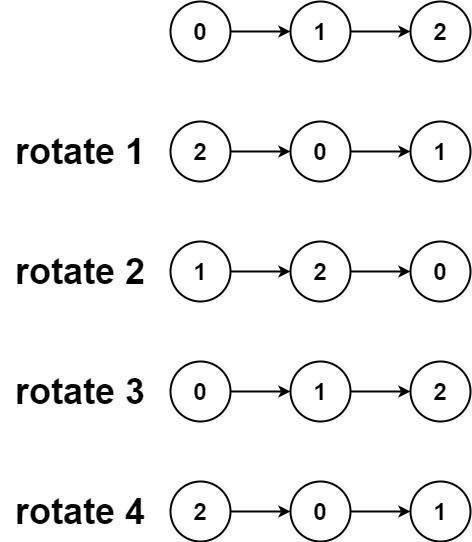
**Example 1:**



**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]

Solution:

/\*\*

\* 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) {

//edge cases

if(head == null || head.next == null || k==0)

return head;

ListNode curr = head;

//compute the length

int len = 1;

while(curr.next != null){

len++;

curr = curr.next;

}

//go till that node

curr.next = head;

k = k%len;

k = len-k;

while(k-->0)

curr = curr.next;

//make head to curr next node and break connection

head = curr.next;

curr.next = null;

return head;

} Traversing Nodes for total count

}



Comlexity= O(N) + O(N-N%K)

== O(N)



Traversing Nodes to break and rotate