

## Solution Review: Problem Challenge 2

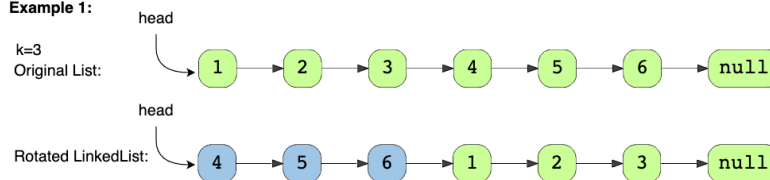
### We'll cover the following

- Rotate a LinkedList (medium)
- Solution
  - Code
  - Time complexity
  - Space complexity

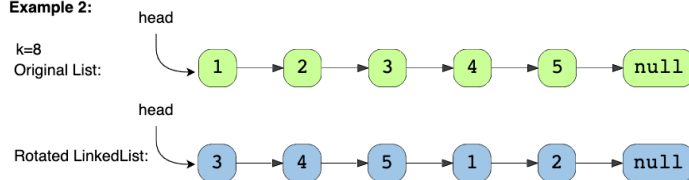
## Rotate a LinkedList (medium) #

Given the head of a Singly LinkedList and a number 'k', rotate the LinkedList to the right by 'k' nodes.

### Example 1:



### Example 2:



## Solution #

Another way of defining the rotation is to take the sub-list of 'k' ending nodes of the LinkedList and connect them to the beginning. Other than that we have to do three more things:

1. Connect the last node of the LinkedList to the head, because the list will have a different tail after the rotation.
2. The new head of the LinkedList will be the node at the beginning of the sublist.
3. The node right before the start of sub-list will be the new tail of the rotated LinkedList.

## Code #

Here is what our algorithm will look like:

```
Java Python3 C++ JS

1 import java.util.*;
2
3 class ListNode {
4     int value = 0;
5     ListNode next;
6
7     ListNode(int value) {
8         this.value = value;
9     }
10 }
11
12 class RotateList {
13
14     public static ListNode rotate(ListNode head, int rotations) {
15         if (head == null || head.next == null || rotations <= 0)
16             return head;
17
18         // find the length and the last node of the list
19         ListNode lastNode = head;
20         int listLength = 1;
21         while (lastNode.next != null) {
```

```
22     lastNode = lastNode.next;
23     listLength++;
24 }
25
26 lastNode.next = head; // connect the last node with the head to make it a circular list
27 rotations %= listLength; // no need to do rotations more than the length of the list
28 int skipLength = listLength - rotations;
```

Run

Save

Reset

### Time complexity #

The time complexity of our algorithm will be  $O(N)$  where 'N' is the total number of nodes in the LinkedList.

### Space complexity #

We only used constant space, therefore, the space complexity of our algorithm is  $O(1)$ .



[← Back](#)

Problem Challenge 2

[Next →](#)

Introduction

☒ Mark as Completed

 Report an Issue  Ask a Question