Remove Nth Node From End Of List

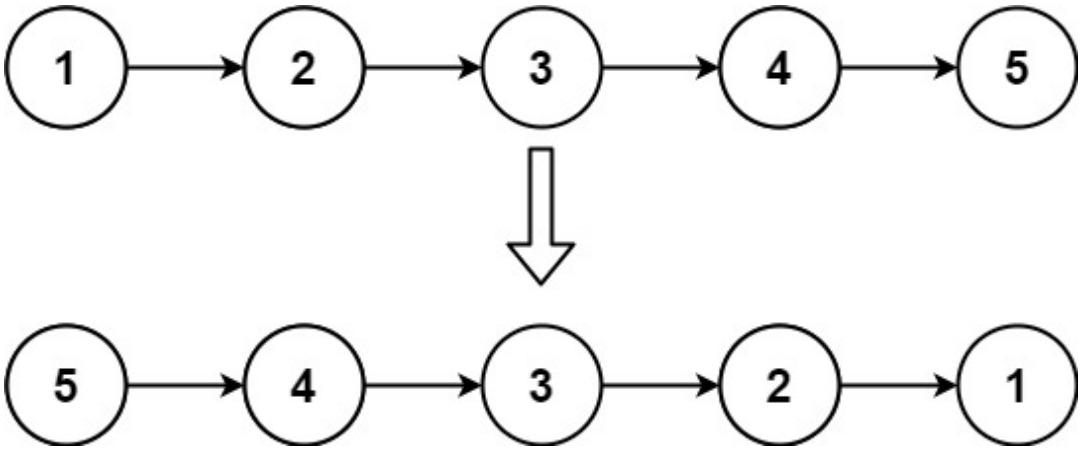
Difficulty	Medium
Category	LinkedList
Question	https://leetcode.com/problems/remove-nth-node-from-end-of-list/
Solution	https://youtu.be/XVuQxVej6y8
Status	Done

Question

> Given the `head` of a singly linked list, reverse the list, and return *the reversed list*.

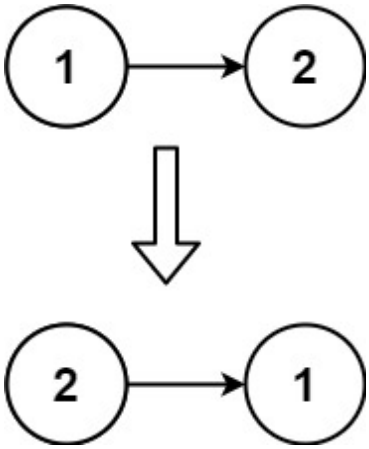
Example

Example 1:



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

Example 2:




```
Input: head = [1,2]
Output: [2,1]
```

Example 3:

```
Input: head = []
Output: []
```

Idea

Solution

```
class Solution:
    def removeNthFromEnd(self, head, n: int):
        # Create a dummy node with a value of 0 and make it the new head.
        # This helps handle edge cases where the target node is the first node.
        dummy = ListNode(0, head)

        # Initialize two pointers, 'left' and 'right', both initially pointing to the head.
        left = dummy
```

```
        right = head

        # Move the 'right' pointer 'n' nodes ahead in the linked list.
        while n > 0 and right:
            right = right.next
            n -= 1

        # Move both 'left' and 'right' pointers simultaneously until 'right' reaches the end.
        while right:
            left = left.next
            right = right.next

        # Update the 'next' pointer of the 'left' node to skip the target node.
        left.next = left.next.next

        # Return the modified linked list starting from the dummy node's 'next'.
        return dummy.next
```

💡 **Time Complexity:**

The time complexity of this code is **O(n)**, where 'n' is the No. of nodes in linked list.

Space Complexity:

The space complexity of this code is **O(1)**, or constant space.