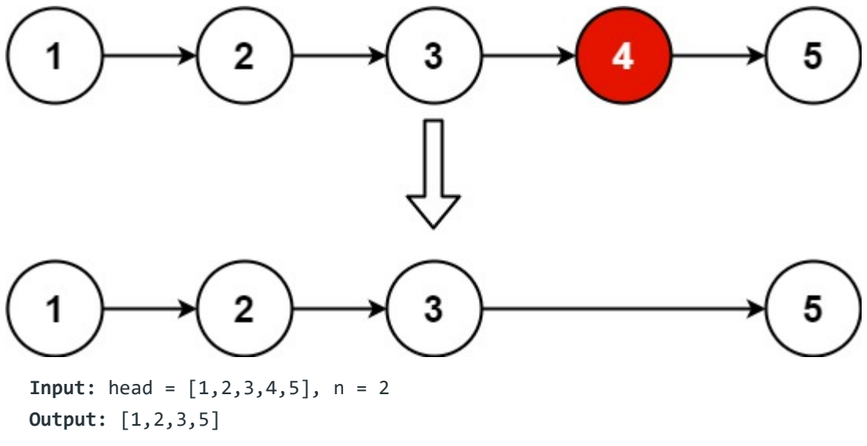
19. Remove Nth Node From End of List

Medium 8492 406 Add to List Share

Given the `head` of a linked list, remove the `n^th` node from the end of the list and return its head.

Example 1:




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

Example 2:

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

Example 3:

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

Constraints:

- The number of nodes in the list is `sz`.
- `1 <= sz <= 30`
- `0 <= Node.val <= 100`

Java Autocomplete

```java
/**
 * 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 removeNthFromEnd(ListNode head, int n) {

        ListNode slowPointer = head;
        ListNode fastPointer = head;

        //If there is only 1 element then return null
        if(head.next == null)
            return null;

        for(int i = 0; i < n; i++){
            fastPointer = fastPointer.next;
        }

        //Need to concerned only when n = listSize. In that case we have to remove head.
        // And in that case fastPointer will be null. Return just the second node.
        if(fastPointer == null)
            return head.next;

        while(fastPointer.next != null){
            slowPointer = slowPointer.next;
            fastPointer = fastPointer.next;
        }


        slowPointer.next = slowPointer.next.next;

        return head;
    }
}
```

Your previous code was restored from your local storage. Reset to default