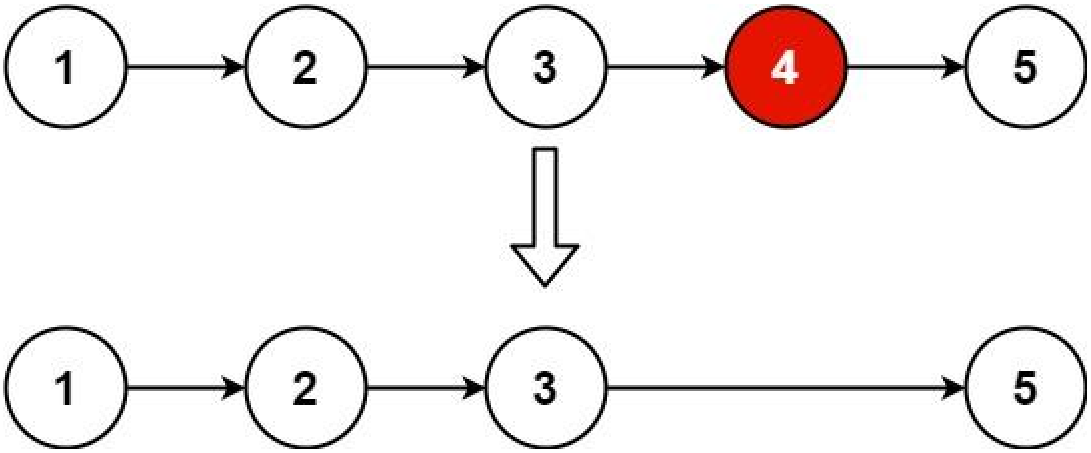
Problem Link:

https://leetcode.com/problems/remove-nth-node-from-end-of-list

Problem Description:

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



Problem Approach:

To remove the n-th node from the end of the list, we first advance the head pointer by n steps. Then, we iterate through the list, reconstructing a new list while skipping the target node to be removed. This ensures that by the time we reach the end of the list, we have skipped the correct node.

Solution:

We create two pointers, iterator and head. iterator will point to the current node in the original list, and head will move n steps ahead. We create a new dummy node resultList, which will hold the new linked list after we remove the n-th node. resultListIterator is used to construct this new list. We first move head forward by n nodes to establish a gap between head and iterator. We iterate through the list as long as head is not None, we append nodes from iterator to resultListIterator. Both head and iterator are advanced at the same pace, ensuring that when head reaches the end, the iterator is at the node just before the node to be removed. When head becomes None, the iterator points to the node just before the one we want to remove. We skip the next node by adjusting resultListIterator.next. We return resultList.next, which points to the head of the modified list.

The time complexity of this algorithm is O(n) and space complexity is O(1), making it an efficient solution.

Code (Python):

```python
def removeNthFromEnd(self, head: Optional[ListNode], n: int) ->
Optional[ListNode]:
        iterator = head
        resultList = ListNode(0)
        resultListIterator = resultList
        for _ in range(n):
            head = head.next
        while head != None:
            resultListIterator.next = iterator
            resultListIterator = resultListIterator.next
            iterator = iterator.next
            head = head.next
        resultListIterator.next = iterator.next
        return resultList.next
```