Problem 19: Remove Nth Node From End of List

Problem Information

Difficulty: Medium
Acceptance Rate: 50.30%
Paid Only: No
Tags: Linked List, Two Pointers

Problem Description

Given the `head` of a linked list, remove the `nth` 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` * `1 <= n <= sz`

Follow up: Could you do this in one pass?

Code Snippets

C++:

```
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode() : val(0), next(nullptr) {}
* ListNode(int x) : val(x), next(nullptr) {}
* ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
ListNode* removeNthFromEnd(ListNode* head, int n) {

}
};
```

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) {

}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
```

```
# def __init__(self, val=0, next=None):
# self.val = val
# self.next = next
class Solution:
def removeNthFromEnd(self, head: Optional[ListNode], n: int) ->
Optional[ListNode]:
```