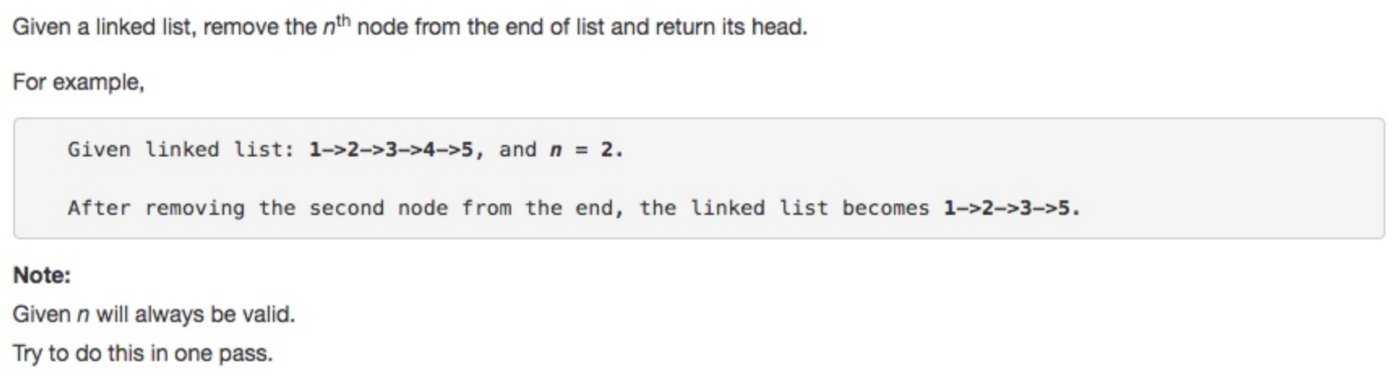
019. Remove Nth Node From End of List

019 Remove Nth Node From End of List

- **Linked List+Two Pointers**

Description



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

For example,

```
Given linked list: 1->2->3->4->5, and n = 2.

After removing the second node from the end, the linked list becomes 1->2->3->5.
```

Note:
Given *n* will always be valid.
Try to do this in one pass.

1. Thought Line

2. Linked List+Two Pointers

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

        // calculate the total size of list
        ListNode* node = head;
        while(node!=nullptr){
            ++sizeList;
            node = node->next;
        }
        if(n<1 || n>sizeList) return head;

        // find the node at N+1 from end (sizeList - N from head)
        ListNode* dummyHead = new ListNode(0);
        dummyHead->next = head;
        int count = 0;
        ListNode* findTheNodeBeforeDeleteNode = dummyHead;
        ListNode* findTheNodeOfDeleteNode = head;
        while(count<sizeList - n){
            findTheNodeBeforeDeleteNode = findTheNodeBeforeDeleteNode->next;
            findTheNodeOfDeleteNode = findTheNodeOfDeleteNode->next;
            ++count;
        }
        findTheNodeBeforeDeleteNode->next = findTheNodeOfDeleteNode->next;
        findTheNodeOfDeleteNode->next = nullptr;
        return dummyHead->next;
    }
};
```