LeetCode19: Remove Nth Node From End of List

题目：

Remove Nth Node From End of List

Total Accepted: **84986** Total Submissions: **304454** Difficulty: **Easy**

> 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.

思路：
问题是如何找到倒数的第n个节点，方法有很多，例如可以写一个size()函数把size求出来然后顺序找第 size-n个，但这样的话就是进行了两次查找题目，第一次求size，第二次找size-n。另外题目提出要求do it in one pass。
另一种方法是双指针法。

假设由两个指针，开始时都指向头节点。第一个指针先走n步，然后第二个指针开始和第一个指针同时开始走，一直到第一个指针走到链表的尾部。这个时候后走的那个指针正好指向倒数的第 n 个节点处。这样只需要查找一次，复杂度为O(N)

代码如下：

```python
class Solution(object):

    def removeNthFromEnd(self, head, n):

        # we create a new node to assistant the searching
        #  set the new node to be the head of the linkList
        nHead = ListNode(0)
        nHead.next = head

        pointer1 = 0            # set up pointer 1
        pointer1Node = nHead  # set up the node that is pointed by pointer1

        # let pointer 1 to process by n units,
        #  update the node that is pointed by pointer 1 correspondingly
        while pointer1 < n:
            pointer1Node = pointer1Node.next
            pointer1 += 1

        # after the pointer 1 went through n units,
        #  set up pointer 2 , actually we just need to set up the node that is pointed by pointer 2
        #   while no need to really set the pointer 2, because we no longer need to count how many step the pointer go through
        pointer2Node = nHead

        # let the two pointer process through the linklist simutaneously,
        #  until the pointer 1 get to the end of the list
        while pointer1Node is not None:
            pointer1Node = pointer1Node.next
            pointer2Node  = pointer2Node .next

        # now the pointer 2 is pointing to the right node that we want to delete
        #  so we just need to setup the "next" data of this node correctly for deleting
        pointer2Node.next = pointer2Node.next.next

        return nHead.next
```