19. Remove Nth Node From End of List

Description | Hints | Submissions | Discuss | Solution

Pick One

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

Example:

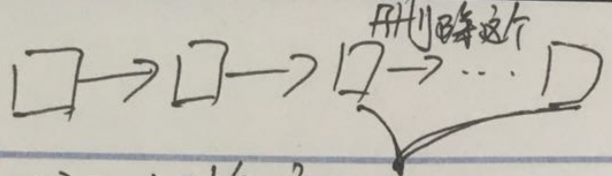
```
Given linked list: 1->2->3->4->5, and n = 2.

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

Note:

Given *n* will always be valid.

Follow up:

Could you do this in one pass?



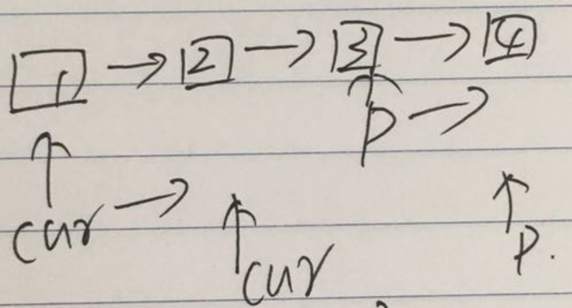
第一步：找到相等长度之点..

如 1 → 2 → 3 → 4　n=2，即删除3

找到 ← n+1 → = p（为什么n+1）。因为要记住3之前一个位置

cur

第二步。将cur与p同时后移。



如果 p.next = null，停。cur.next = cur.next.ne

但其中有 1 → 2　n=2　，结果应为2.

则如果第一步后 p=null.

则 return head.next

```java
public class L19 {
    public class ListNode {
              int val;
              ListNode next;
              ListNode(int x) { val = x; }
          }

    public ListNode removeNthFromEnd(ListNode head, int n) {
        if(head == null)
            return null;
           ListNode p = head;
           for(int i = 0; i < n; i ++) {
               p = p.next;
               if(p == null) {
                   if(i == n-1) {
                       return head.next;
                   }else {
                       return null;
                   }
               }
           }
           ListNode cur = head;
           while (p.next != null) {
               p = p.next;
               cur = cur.next;
           }
           cur.next = cur.next.next;
           return head;

    }
```