

# LeetCode 第 19 号问题：删除链表的倒数第 N 个节点

## 题目描述

给定一个链表，删除链表的倒数第  $n$  个节点，并且返回链表的头结点。

示例：

给定一个链表：1→2→3→4→5，和  $n = 2$ 。

当删除了倒数第二个节点后，链表变为 1→2→3→5。

说明：

给定的  $n$  保证是有效的。

进阶：

你能尝试使用一趟扫描实现吗？

## 题目解析

采取双重遍历肯定是可以解决问题的，但题目要求我们一次遍历解决问题，那我们的思路得发散一下。

我们可以设想假设设定了双指针  $p$  和  $q$  的话，当  $q$  指向末尾的 `NULL`， $p$  与  $q$  之间相隔的元素个数为  $n$  时，那么删除掉  $p$  的下一个指针就完成了要求。

- 设置虚拟节点 `dummyHead` 指向 `head`
- 设定双指针  $p$  和  $q$ ，初始都指向虚拟节点 `dummyHead`
- 移动  $q$ ，直到  $p$  与  $q$  之间相隔的元素个数为  $n$
- 同时移动  $p$  与  $q$ ，直到  $q$  指向的为 `NULL`
- 将  $p$  的下一个节点指向下下个节点

## 动画描述

## 19. Remove Nth Node From End of List

$$n = 2$$



公众号：猿了个猴

### 代码实现

```
class Solution {
public:
    ListNode* removeNthFromEnd(ListNode* head, int n) {
        ListNode* dummyHead = new ListNode(0);
        dummyHead->next = head;

        ListNode* p = dummyHead;
        ListNode* q = dummyHead;
        for( int i = 0 ; i < n + 1 ; i ++ ){
            q = q->next;
        }

        while(q){
            p = p->next;
            q = q->next;
        }

        ListNode* delNode = p->next;
        p->next = delNode->next;
        delete delNode;

        ListNode* retNode = dummyHead->next;
        delete dummyHead;

        return retNode;
    }
};
```

```
}  
};
```

## 19. Remove Nth Node From End of List

$$n = 2$$

