

# 0019. 删除链表的倒数第 N 个结点

👤 ITCharge ⌚ 大约 2 分钟

- 标签：链表、双指针
- 难度：中等

## 题目链接

- [0019. 删除链表的倒数第 N 个结点 - 力扣](#)

## 题目大意

**描述：** 给定一个链表的头节点 `head` 。

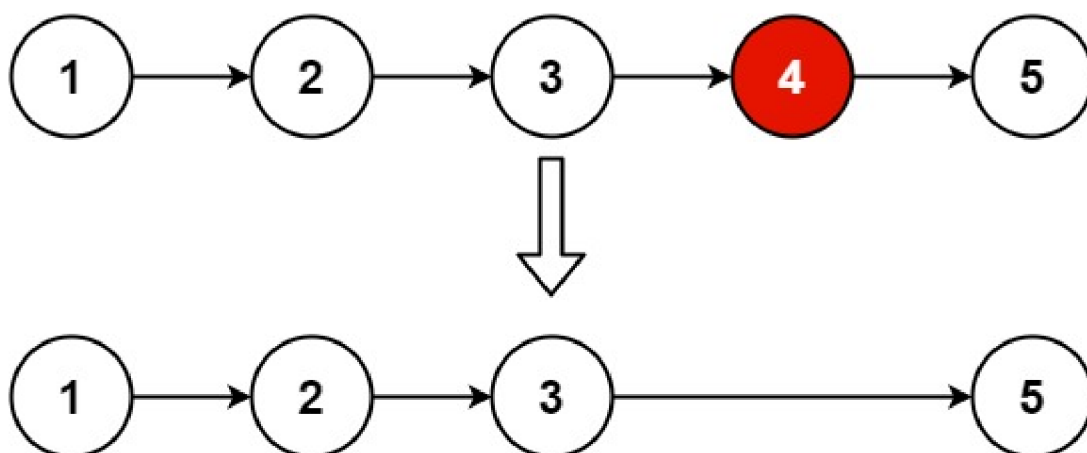
**要求：** 删除链表的倒数第 `n` 个节点，并且返回链表的头节点。

**说明：**

- 要求使用一次遍历实现。
- 链表中结点的数目为 `sz` 。
- $1 \leq sz \leq 30$ 。
- $0 \leq Node.val \leq 100$ 。
- $1 \leq n \leq sz$ 。

**示例：**

- 示例 1：



py

输入: head = [1,2,3,4,5], n = 2

输出: [1,2,3,5]

- 示例 2:

py

输入: head = [1], n = 1

输出: []

## 解题思路

### 思路 1: 快慢指针

常规思路是遍历一遍链表, 求出链表长度, 再遍历一遍到对应位置, 删除该位置上的节点。

如果用一次遍历实现的话, 可以使用快慢指针。让快指针先走  $n$  步, 然后快慢指针、慢指针再同时走, 每次一步, 这样等快指针遍历到链表尾部的时候, 慢指针就刚好遍历到了倒数第  $n$  个节点位置。将该位置上的节点删除即可。

需要注意的是要删除的节点可能包含头节点。我们可以考虑在遍历之前, 新建一个头节点, 让其指向原来的头节点。这样, 最终如果删除的是头节点, 则删除原头节点即可。返回结果的时候, 可以直接返回新建头节点的下一位节点。

### 思路 1: 代码

py

```
class Solution:
    def removeNthFromEnd(self, head: ListNode, n: int) -> ListNode:
        newHead = ListNode(0, head)
        fast = head
        slow = newHead
        while n:
            fast = fast.next
            n -= 1
```

```
while fast:
    fast = fast.next
    slow = slow.next
slow.next = slow.next.next
return newHead.next
```

## 思路 1：复杂度分析

- 时间复杂度： $O(n)$ 。
- 空间复杂度： $O(1)$ 。