203. Remove Linked List Elements

想法：比較相等的時候 指向的next變成目前的next

# Definition for singly-linked list.

# class ListNode:

# def \_\_init\_\_(self, val=0, next=None):

# self.val = val

# self.next = next

class Solution:

def removeElements(self, head: Optional[ListNode], val: int) -> Optional[ListNode]:

count = 0

if head is None:

return head

pre = head

curr = pre.next

while pre.val == val and pre.next:

if count == 0:

pre = head.next

curr = pre.next

count += 1

else:

pre = curr

curr = curr.next

if pre.val == val:

return None

res = pre

while curr:

while curr.val == val and curr.next:

curr = curr.next

pre.next = curr

if curr.val == val and curr.next is None:

pre.next = None

pre = curr

curr = curr.next

return res

def removeElements(self, head: Optional[ListNode], val: int) -> Optional[ListNode]:

if head == None:

return

elif head.val == val:

head = self.removeElements(head.next,val)

else:

head.next = self.removeElements(head.next,val)

return head