82 remove duplicates from sorted list II

Created	@March 9, 2022 4:02 PM
Tags	Medium
link	https://leetcode.com/problems/remove-duplicates-from-sorted-list-ii/
Problem Number	82
companies	amazon google

Description

Given the `head` of a sorted linked list, *delete all nodes that have duplicate numbers, leaving only distinct numbers from the original list*. Return *the linked list **sorted** as well*.

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def get_count(self, head):
        if head == None:
            return 0, None

        if head.next is None:
            return 1, None

        count = 0
        val = head.val
        while head is not None and head.val == val:
            count += 1
            head = head.next
        return count, head

    def deleteDuplicates(self, head: Optional[ListNode]) -> Optional[ListNode]:
        print ("82. Remove Duplicates from Sorted List II".lower())
        if head is None:
            return head

        copy_head = head
        prev = None

        while copy_head is not None:
            count, next_head = self.get_count(copy_head)
            print (f'count {count} next_head {next_head}')

            if count == 1:
                if prev is None:
                    head = copy_head
                    prev = head
                else:
                    prev.next = copy_head
                    prev = prev.next

            copy_head = next_head
        if prev is not None:
            prev.next = None

        return head if prev is not None else None
```

Note: Using a “Sentinal node” as head makes the code much easier