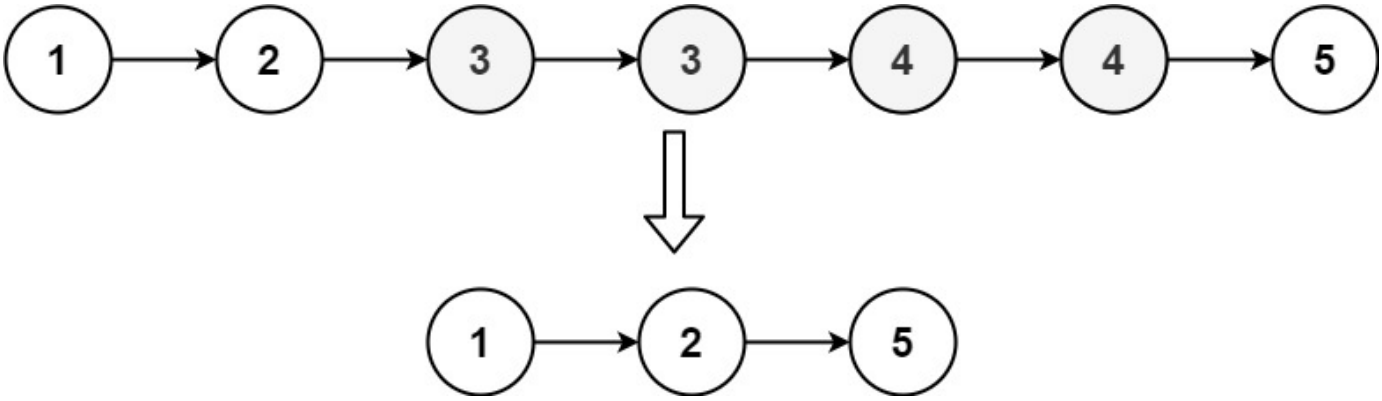
# 82. Remove Duplicates from Sorted List II

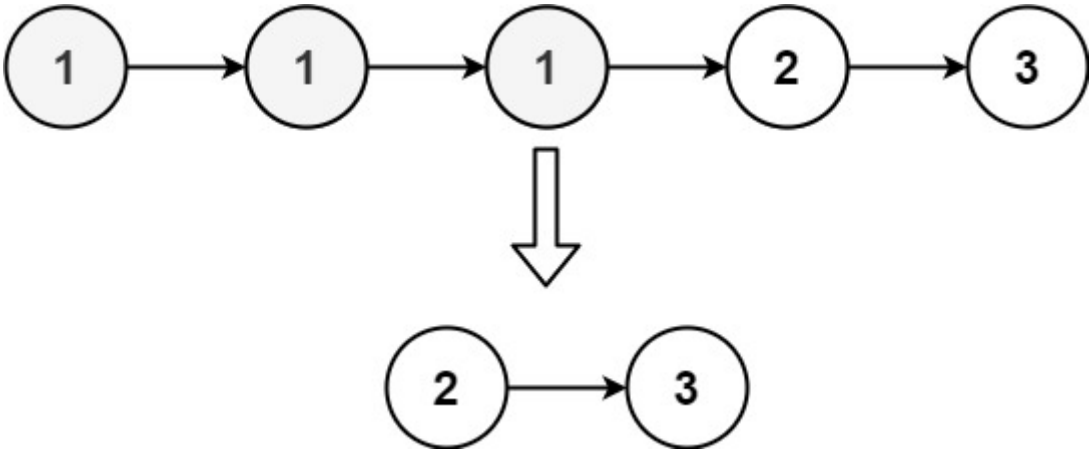
Given the `head` of a sorted linked list, *delete all nodes that have duplicate numbers, leaving only distinct numbers from the original list*. Return *the linked list* ***sorted*** *as well*.

**Example 1:**




```
Input: head = [1,2,3,3,4,4,5]
Output: [1,2,5]
```

**Example 2:**




```
Input: head = [1,1,1,2,3]
Output: [2,3]
```

**Constraints:**

- The number of nodes in the list is in the range `[0, 300]`.
- `-100 <= Node.val <= 100`
- The list is guaranteed to be **sorted** in ascending order.

```python
def deleteDuplicates(self, head: Optional[ListNode]) -> Optional[ListNode]:
        if head is None or head.next is None:
            return head
        curr = head
        dummyHead = ListNode(-1)
        prev = dummyHead

        while curr!=None and curr.next!=None:
            if curr.val!=curr.next.val:
                prev.next = curr
                prev = curr
                curr = curr.next
            else:
                temp = curr.val
                while curr!=None and curr.val==temp:
                    curr = curr.next
        if curr!=None:
            prev.next = curr
            prev = curr
        if curr is None:
            prev.next = None
        return dummyHead.next
```