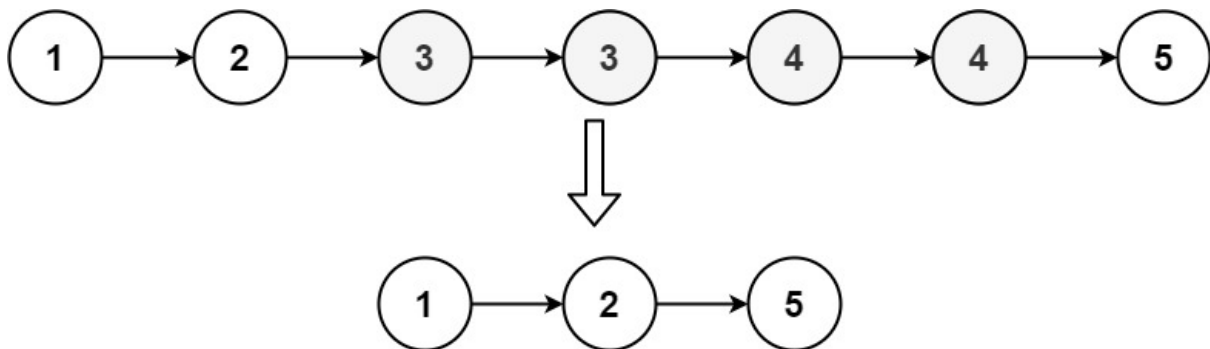


## 82. Remove Duplicates from Sorted List II

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.

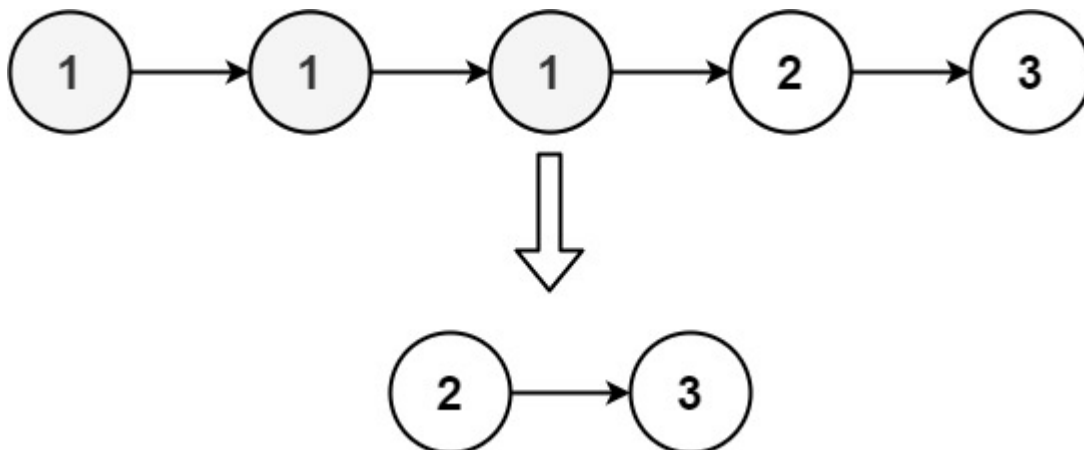
**Example 1:**



**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].
- $-100 \leq \text{Node.val} \leq 100$
- The list is guaranteed to be **sorted** in ascending order.

```
# Definition for singly-linked list.
# class ListNode(object):
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution(object):
    def deleteDuplicates(self, head):
        """
        :type head: ListNode
        :rtype: ListNode
        """
        if not head:
            return head

        output = ListNode(0, head)
        ptr = output

        while head:
            if head.next and head.val == head.next.val:
                while head.next and head.val == head.next.val:
                    head = head.next
                ptr.next = head.next
            else:
                ptr = ptr.next
            head = head.next

        return output.next
```