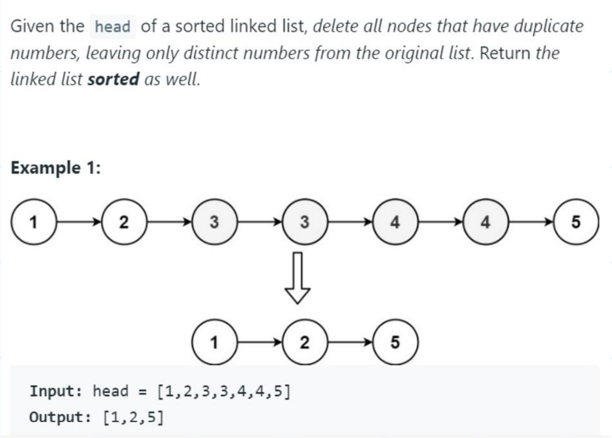
82. Remove Duplicates from Sorted List II

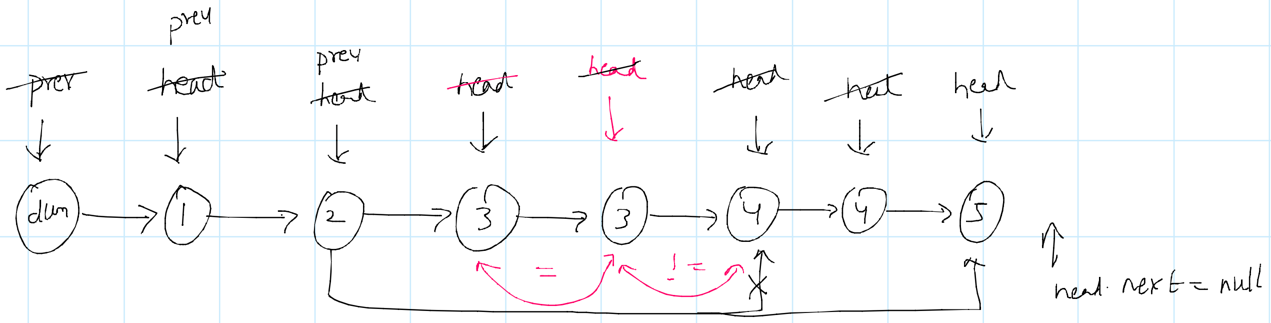
09 March 2022 09:33 AM



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]
```

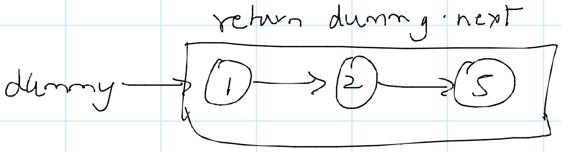


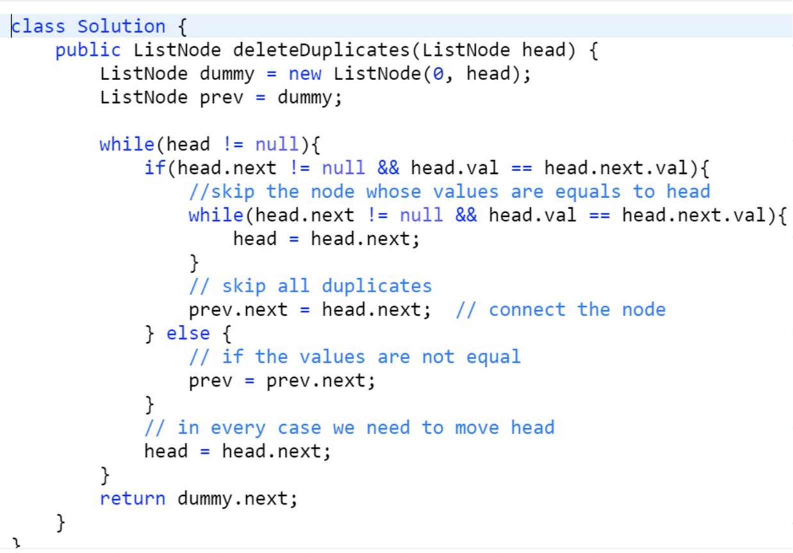
we compare the values of head and head - next

if the value are equal we skip, so we move head, so we point nex.head to next.prev

if the values are not equal we move head and prev

Finall we return dummy.next





```
class Solution {
    public ListNode deleteDuplicates(ListNode head) {
        ListNode dummy = new ListNode(0, head);
        ListNode prev = dummy;

        while(head != null){
            if(head.next != null && head.val == head.next.val){
                //skip the node whose values are equals to head
                while(head.next != null && head.val == head.next.val){
                    head = head.next;
                }
                // skip all duplicates
                prev.next = head.next;  // connect the node
            } else {
                // if the values are not equal
                prev = prev.next;
            }
            // in every case we need to move head
            head = head.next;
        }
        return dummy.next;
    }
}
```