Remove Duplicates from Sorted List II

Given a sorted linked list, delete all nodes that have duplicate numbers, leaving only *distinct* numbers from the original list.

For example,
Given `1->2->3->3->4->4->5`, return `1->2->5`.
Given `1->1->1->2->3`, return `2->3`.

Solution 1

```
public ListNode deleteDuplicates(ListNode head) {
        if(head==null) return null;
        ListNode FakeHead=new ListNode(0);
        FakeHead.next=head;
        ListNode pre=FakeHead;
        ListNode cur=head;
        while(cur!=null){
            while(cur.next!=null&&cur.val==cur.next.val){
                cur=cur.next;
            }
            if(pre.next==cur){
                pre=pre.next;
            }
            else{
                pre.next=cur.next;
            }
            cur=cur.next;
        }
        return FakeHead.next;
    }
```

written by snowfish original link here

Solution 2

```
public ListNode deleteDuplicates(ListNode head) {
    if (head == null) return null;

    if (head.next != null && head.val == head.next.val) {
        while (head.next != null && head.val == head.next.val) {
            head = head.next;
        }
        return deleteDuplicates(head.next);
    } else {
        head.next = deleteDuplicates(head.next);
    }
    return head;
}
```

if current node is not unique, return deleteDuplicates with head.next. If current node is unique, link it to the result of next list made by recursive call. Any improvement?

written by totalheap original link here

Solution 3

```
class Solution {
public:
    ListNode *deleteDuplicates(ListNode *head) {
        ListNode **runner = &head;

        if(!head || !head->next)return head;

        while(*runner)
        {
            if((*runner)->next && (*runner)->next->val == (*runner)->val)
            {
                ListNode *temp = *runner;
                while(temp && (*runner)->val == temp->val)
                    temp = temp->next;

                *runner = temp;
            }
            else
                runner = &((*runner)->next);
        }

        return head;
    }
};
```

written by tommy1122337 original link here

From Leetcoder.