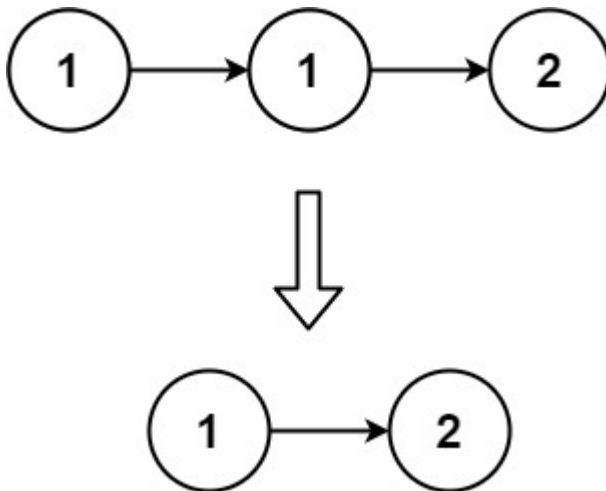


## Leetcode Problem 1. (Easy)

### Remove Duplicates from Sorted List

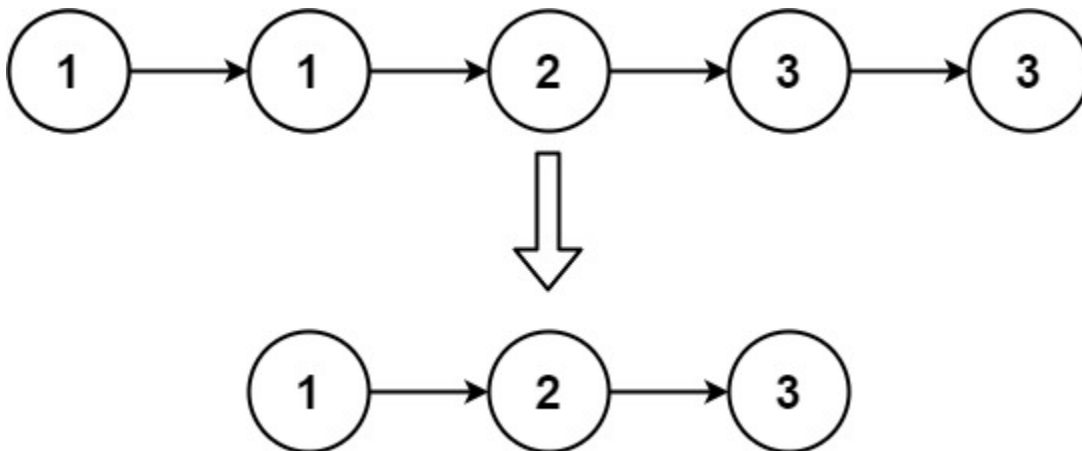
Given the **head** of a sorted linked list, *delete all duplicates such that each element appears only once*. Return the linked list **sorted** as well.

#### Example 1:



**Input:** head = [1,1,2]  
**Output:** [1,2]

#### Example 2:



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

#### 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.

Link: <https://leetcode.com/problems/remove-duplicates-from-sorted-list/>

```
class Solution {
    public ListNode deleteDuplicates(ListNode head) {

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

The screenshot shows the LeetCode interface for the problem "82. Remove Duplicates from Sorted List II". The solution is marked as "Accepted". The performance metrics are as follows:

Metric	Value
Runtime	0 ms
Beats	100%
Memory	42.2 MB
Beats	54.85%

The code is written in Java and defines a singly-linked list structure and a solution class.

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode() {}
 *     ListNode(int val) { this.val = val; }
 *     ListNode(int val, ListNode next) { this.val = val; this.next = next; }
 * }
 */
class Solution {
    public ListNode deleteDuplicates(ListNode head) {
```