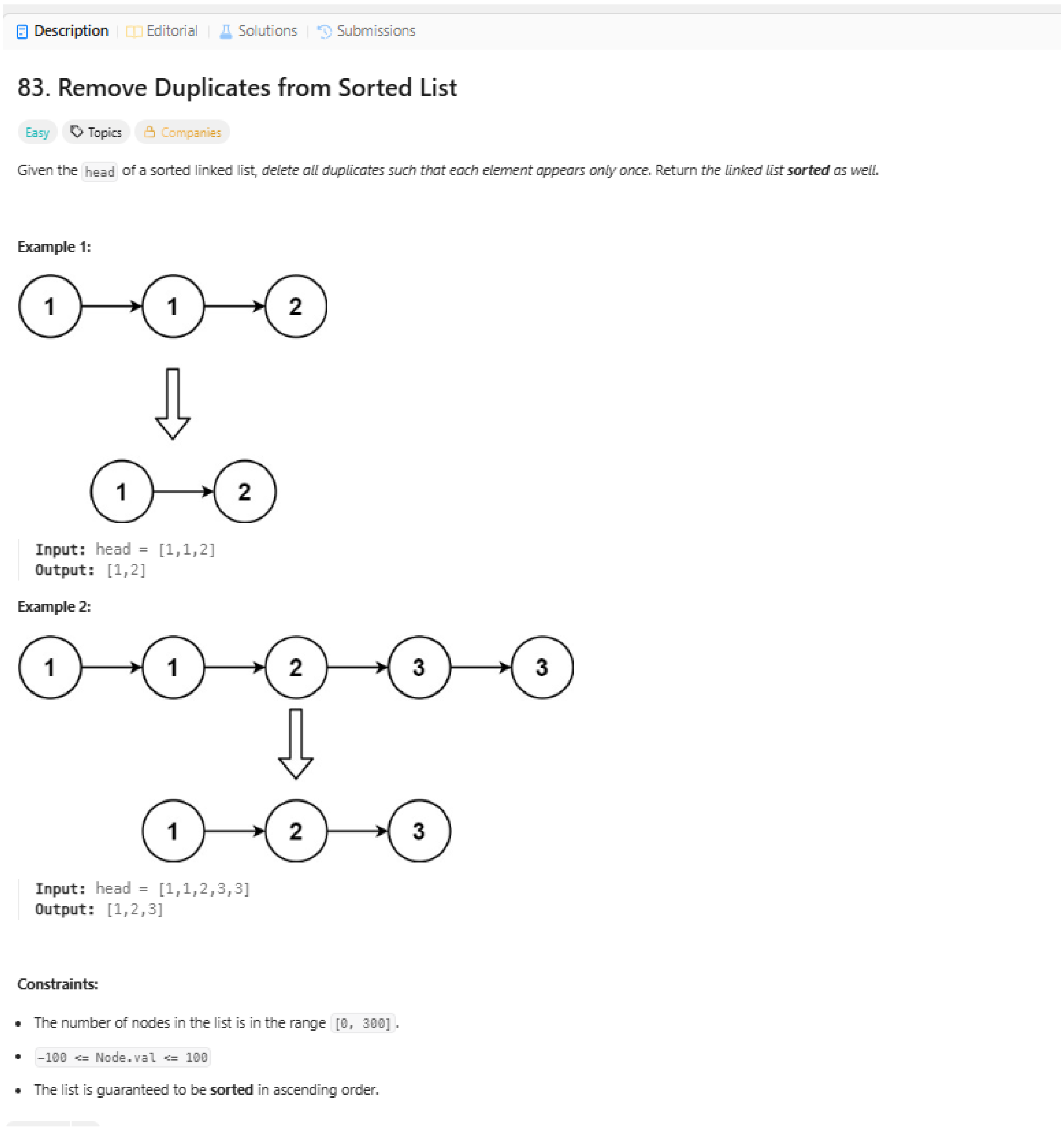
LEETCODE PROBLEM – 83



83. 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:

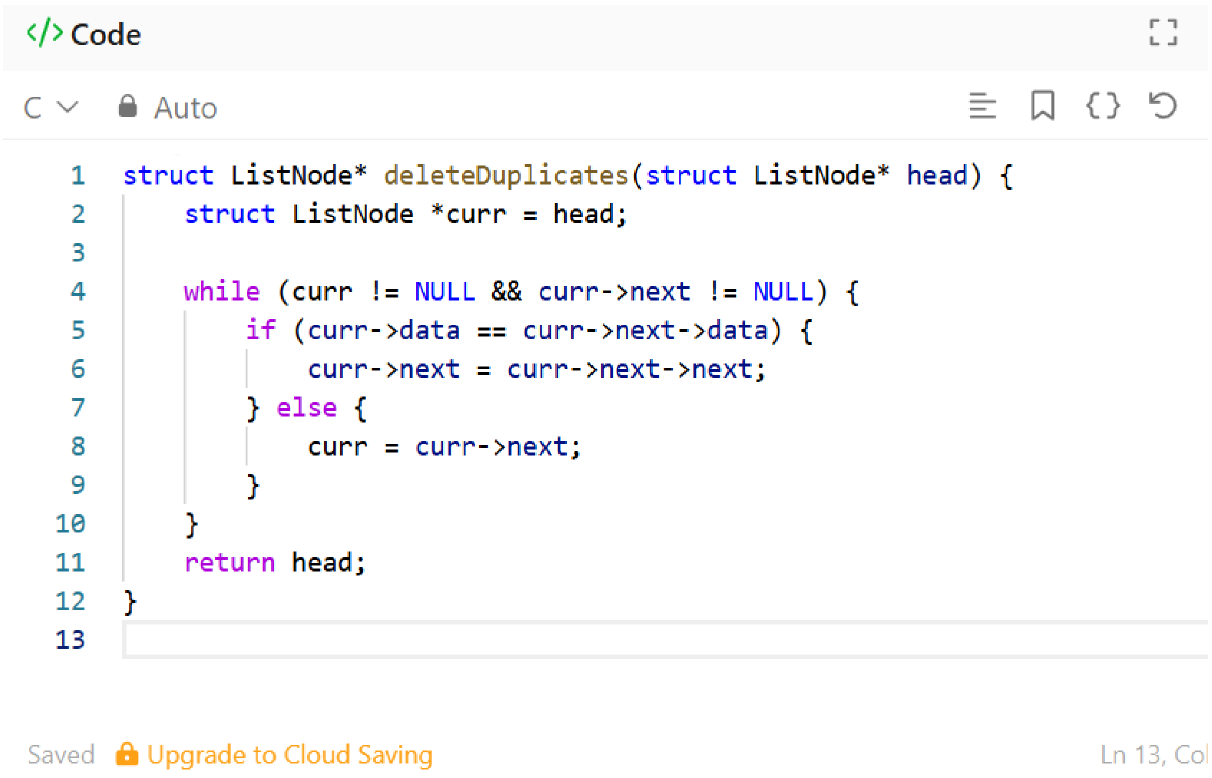
```
Input: head = [1,1,2]
Output: [1,2]
```

Example 2:

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

Constraints:

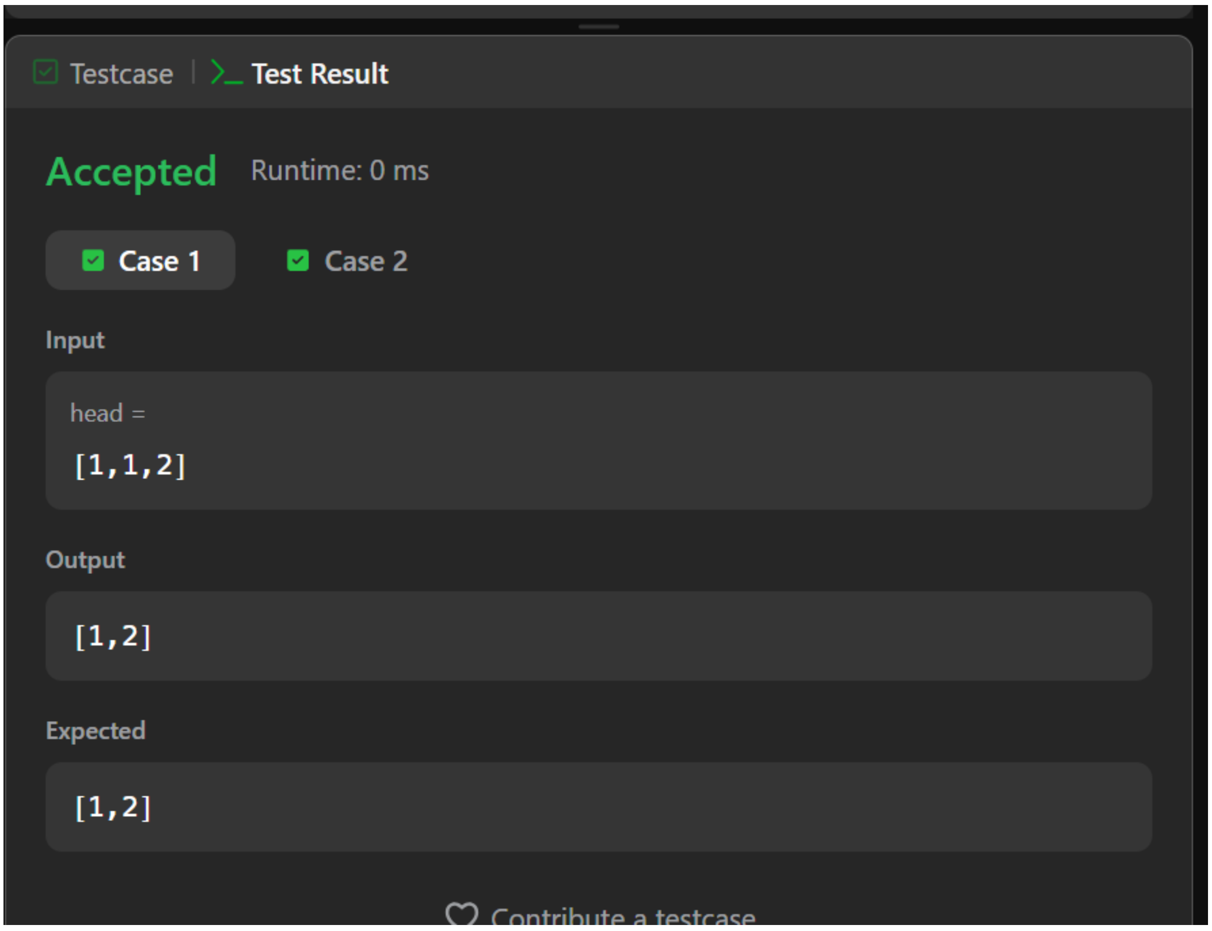
- The number of nodes in the list is in the range `[0, 300]`.
- `-100 <= Node.val <= 100`
- The list is guaranteed to be **sorted** in ascending order.



```c
struct ListNode* deleteDuplicates(struct ListNode* head) {
    struct ListNode *curr = head;

    while (curr != NULL && curr->next != NULL) {
        if (curr->data == curr->next->data) {
            curr->next = curr->next->next;
        } else {
            curr = curr->next;
        }
    }
    return head;
}

```



Accepted Runtime: 0 ms

Input

head =

```
[1,1,2]
```

Output

```
[1,2]
```

Expected

```
[1,2]
```