083 Remove Duplicates from Sorted List

2018年3月30日 13:37

Question:

Given a sorted linked list, delete all duplicates such that each element appear only *once*.
For example,
Given 1->1->2, return 1->2.
Given 1->1->2->3->3, return 1->2->3.

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

给定一个排序链表，删除所有重复的元素使得每个元素只留下一个。

案例:

给定 1->1->2，返回 1->2

给定 1->1->2->3->3，返回 1->2->3

Solution for Python3:

```python
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, x):
#         self.val = x
#         self.next = None

class Solution1:
    def deleteDuplicates(self, head):
        """
        :type head: ListNode
        :rtype: ListNode
        """
        l, r = head, head
        while l and r:
            r = r.next
            while r and r.val == l.val:
                r = r.next
            l.next = r
            l = r
        return head

class Solution2:
    def deleteDuplicates(self, head):
        """
        :type head: ListNode
```

```
        :rtype: ListNode
        """
        c = head
        while c and c.next:
            if c.next.val == c.val:
                c.next = c.next.next
            else:
                c = c.next
        return head
```

Solution for C++:

```
class Solution {
public:
    ListNode* deleteDuplicates(ListNode* head) {
        ListNode* c = head;
        while (c && c->next) {
           if (c->next->val == c->val) {
               c->next = c->next->next;
           } else {
               c = c->next;
           }
        }
        return head;
    }
};
```