

# 0082. 删除排序链表中的重复元素 II

👤 ITCharge 🕒 大约 2 分钟

- 标签：链表、双指针
- 难度：中等

## 题目链接

- [0082. 删除排序链表中的重复元素 II - 力扣](#)

## 题目大意

**描述：** 给定一个已排序的链表的头 *head*。

**要求：** 删除原始链表中所有重复数字的节点，只留下不同的数字。返回已排序的链表。

**说明：**

- 链表中节点数目在范围  $[0, 300]$  内。
- $-100 \leq Node.val \leq 100$ 。
- 题目数据保证链表已经按升序排列。

**示例：**

- 示例 1：

输入：head = [1,2,3,3,4,4,5]

输出：[1,2,5]

py

# 解题思路

## 思路 1：遍历

这道题的题意是需要保留所有不同数字，而重复出现的所有数字都要删除。因为给定的链表是升序排列的，所以我们要删除的重复元素在链表中的位置是连续的。所以我们可以对链表进行一次遍历，然后将连续的重复元素从链表中删除即可。具体步骤如下：

- 先使用哑节点 *dummy\_head* 构造一个指向 *head* 的指针，使得可以防止从 *head* 开始就是重复元素。
- 然后使用指针 *cur* 表示链表中当前元素，从 *head* 开始遍历。
- 当指针 *cur* 的下一个元素和下下一个元素存在时：
  - 如果下一个元素值和下下一个元素值相同，则我们使用指针 *temp* 保存下一个元素，并使用 *temp* 向后遍历，跳过所有重复元素，然后令 *cur* 的下一个元素指向 *temp* 的下一个元素，继续向后遍历。
  - 如果下一个元素值和下下一个元素值不同，则令 *cur* 向右移动一位，继续向后遍历。
- 当指针 *cur* 的下一个元素或者下下一个元素不存在时，说明已经遍历完，则返回哑节点 *dummy\_head* 的下一个节点作为头节点。

## 思路 1：代码

```
# Definition for singly-linked list.
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def deleteDuplicates(self, head: ListNode) -> ListNode:
        dummy_head = ListNode(-1)
        dummy_head.next = head

        cur = dummy_head
        while cur.next and cur.next.next:
            if cur.next.val == cur.next.next.val:
                temp = cur.next
                while temp and temp.next and temp.val == temp.next.val:
                    temp = temp.next
```

py

```
        temp = temp.next
    cur.next = temp.next
else:
    cur = cur.next
return dummy_head.next
```

## 思路 1：复杂度分析

- 时间复杂度： $O(n)$ 。其中  $n$  为链表长度。
- 空间复杂度： $O(1)$ 。