

剑指Offer（五十六）：删除链表中重复的结点

🕒 2018年1月25日 09:20:51 🗨 24 🌡 7,074 °C 📄 编辑



一、前言

本系列文章为《剑指Offer》刷题笔记。

刷题平台：[牛客网](#)

书籍下载：[共享资源](#)

二、题目

在一个排序的链表中，存在重复的结点，请删除该链表中重复的结点，重复的结点不保留，返回链表头指针。例如，链表1->2->3->3->4->4->5 处理后为 1->2->5。

1、思路

删除重复结点，只需要记录当前结点前的最晚访问过的不重复结点pPre、当前结点pCur、指向当前结点后面的结点pNext的三个指针即可。如果当前结点和它后面的几个结点数值相同，那么这些结点都要被剔除，然后更新pPre和pCur；如果不相同，则直接更新pPre和pCur。

需要考虑的是，如果第一个结点是重复结点我们该怎么办？这里我们分别处理一下就好，如果第一个结点是重复结点，那么就头指针pHead也更新一下。

2、代码

C++:

```
1  /*
2  struct ListNode {
3      int val;
4      struct ListNode *next;
5      ListNode(int x) :
6          val(x), next(NULL) {
7      }
8  };
9  */
10 class Solution {
11 public:
12     ListNode* deleteDuplication(ListNode* pHead)
13     {
14         if(pHead == NULL){
15             return NULL;
16         }
17         // 指向当前结点前最晚访问过的不重复结点
18         ListNode* pPre = NULL;
```


2021/5/12剑指Offer（五十六）：删除链表中重复的结点

```
19 // 指向当前处理的结点
20 ListNode* pCur = pHead;
21 // 指向当前结点后面的结点
22 ListNode* pNext = NULL;
23
24 while(pCur != NULL){
25     // 如果当前结点与下一个结点相同
26     if(pCur->next != NULL && pCur->val == pCur->next->val){
27         pNext = pCur->next;
28         // 找到不重复的最后一个结点位置
29         while(pNext->next != NULL && pNext->next->val == pCur->val){
30             pNext = pNext->next;
31         }
32         // 如果pCur指向链表中第一个元素, pCur -> ... -> pNext ->...
33         // 要删除pCur到pNext, 将指向链表第一个元素的指针pHead指向pNext->next。
34         if(pCur == pHead){
35             pHead = pNext->next;
36         }
37         // 如果pCur不指向链表中第一个元素, pPre -> pCur ->...->pNext ->...
38         // 要删除pCur到pNext, 即pPre->next = pNext->next
39         else{
40             pPre->next = pNext->next;
41         }
42         // 向前移动
43         pCur = pNext->next;
44     }
45     // 如果当前结点与下一个结点不相同
46     else{
47         pPre = pCur;
48         pCur = pCur->next;
49     }
50 }
51 return pHead;
52 }
53 };
```

Python：

Pytho

```
1 # -*- coding:utf-8 -*-
2 # class ListNode:
3 #     def __init__(self, x):
4 #         self.val = x
5 #         self.next = None
6 class Solution:
7     def deleteDuplication(self, pHead):
8         # write code here
9         pPre = None
10        pCur = pHead
11        pNext = None
12        while pCur != None:
13            if pCur.next != None and pCur.val == pCur.next.val:
14                pNext = pCur.next
15                while pNext.next != None and pNext.next.val == pCur.val:
16                    pNext = pNext.next
17                if pCur == pHead:
18                    pHead = pNext.next
19                else:
20                    pPre.next = pNext.next
21                pCur = pNext.next
22            else:
23                pPre = pCur
24                pCur = pCur.next
25        return pHead
```



微信公众号

分享技术，乐享生活：微信公众号搜索「JackCui-AI」关注一个在互联网摸爬滚打的潜行者。

不在沉默中爆发，就在沉默中灭亡。--- 鲁迅