

```

1  /*
2  给定一个排序链表，删除所有重复的元素，使得每个元素只出现一次。
3
4  示例 1:
5
6  输入：1->1->2
7  输出：1->2
8
9  示例 2:
10
11 输入：1->1->2->3->3
12 输出：1->2->3
13
14 来源：力扣（LeetCode）
15 链接：https://leetcode-cn.com/problems/remove-duplicates-from-sorted-list
16 著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
17 */

```

分析:

- 设置两个节点的滑动窗(窗头和窗尾,窗头的next指针始终指向窗尾节点)来依次遍历所有节点,直到窗尾首先为NULL才停止.
- 在遍历过程中,若窗头节点的val等于窗尾节点的val,就先保存窗尾节点的指针到temp,窗尾指针指向本身的next,窗头的next指向窗尾,最后释放temp指向的空间.
- 在遍历过程中,若窗口节点的val不等于窗尾节点的val,窗口节点的移动到窗尾节点,窗尾节点移动到窗尾节点的next.

方法一:C\_窗口遍历法

```

1  /**
2   * Definition for singly-linked list.
3   * struct ListNode {
4   *     int val;
5   *     struct ListNode *next;
6   * };
7   */
8
9
10 struct ListNode* deleteDuplicates(struct ListNode* head)
11 {
12
13     struct ListNode* frame_head = NULL;
14     struct ListNode* frame_tail = NULL;
15     struct ListNode* temp        = NULL;
16
17     do{
18
19         if( (head == NULL)
20            || (head->next == NULL)
21        )
22        {
23            break;

```

```

24     }
25     else
26     {
27         frame_head = head        ;
28         frame_tail = head->next   ;
29
30         while(frame_tail!=NULL)
31         {
32             if(frame_head->val == frame_tail->val)
33             {
34                 temp = frame_tail;
35                 frame_tail = frame_tail->next;
36                 frame_head->next = frame_tail;
37                 free(temp);
38             }
39             else
40             {
41                 frame_head = frame_tail;
42                 frame_tail = frame_tail -> next;
43             }
44         }
45     }
46 }
47
48 }while(0);
49
50 return head;
51 }
52
53
54 /*
55 执行结果:
56 通过
57 显示详情
58 执行用时 :8 ms, 在所有 C 提交中击败了86.25% 的用户
59 内存消耗 :7.8 MB, 在所有 C 提交中击败了5.21%的用户
60 */

```