82. Remove Duplicates from Sorted List II ★

Question | Editorial Solution

My Submissions (/problems/remove-duplicates-from-sorted-list-ii/submissions/)

Total Accepted: **84518**   Total Submissions: **301998**   Difficulty: **Medium**

Given a sorted linked list, delete all nodes that have duplicate numbers, leaving only *distinct* numbers from the original list.

For example,

Given `1->2->3->3->4->4->5`, return `1->2->5`.

Given `1->1->1->2->3`, return `2->3`.

Subscribe (/subscribe/) to see which companies asked this question

Show Tags

Have you met this question in a real interview? Yes   No

Discuss (https://leetcode.com/discuss/questions/oj/remove-duplicates-from-sorted-list-ii)

Pick One (/problems/random-one-question/)

C++   ⟳   </>

```cpp
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    ListNode* deleteDuplicates(ListNode* head) {
        if (!head) return head;
        ListNode dummy(head->val - 1); // 故意跟head值不同
        ListNode* newListCurNode = &dummy;
        while (head) {
            auto next = head->next;
            if (next && head->val != next->val) {
                newListCurNode->next = head;
                newListCurNode = head;
                head = next;
                newListCurNode->next = NULL;
            }
            else if(next && head->val == next->val) {
                while (next && next->val == head->val) {
```

```
25                delete head;
26                head = next;
27                next = next->next;
28            }
29            delete head;
30            head = next;
31        }
32        else {
33            newListCurNode->next = head;
34            newListCurNode = head;
35            newListCurNode->next = NULL;
36            head = NULL;
37        }
38    }
39    return dummy.next;
40    }
41 };
```

Custom Testcase ☐

Run Code   Submit Solution