

## 82. [Remove duplicates from sorted lists 2](#)

```
class Solution {
public:
    ListNode* deleteDuplicates(ListNode* head) {
        ListNode dummy(0);
        dummy.next = head;
        ListNode* prev = &dummy;

        while (head) {
            if (head->next && head->val == head->next->val) {
                while (head->next && head->val == head->next->val) {
                    head = head->next;
                }
                prev->next = head->next;
            } else {
                prev = prev->next;
            }
            head = head->next;
        }

        return dummy.next;
    }
};
```

← → ↺ 🏠 leetcode.com/problems/remove-duplicates-from-sorted-list-ii/ 🔍 ☆ 🔄 📄 1.00 📁 ⬇️ 👤 ⋮

🔗 Problem List < > 🔗

👤 Run 📄 Submit 🕒 📄

Description Accepted × Editorial 👤 Solutions 🔄 Submissions

← All Submissions 🔗

Accepted 166 / 166 testcases passed

👤 aashima\_narula submitted at Feb 13, 2025 22:07

📄 Editorial 📄 Solution


🕒 Runtime

0 ms | Beats 100.00% 🏆

📄 Analyze Complexity

📄 Memory

15.52 MB | Beats 92.77% 🏆



Runtime (ms)	Beats (%)
0	100.00
1	~0
2	~0
3	~0
4	~0

Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
```

🔗 View more

More challenges

Code

C++ v Auto

```
5  *   ListNode *next;
6  *   ListNode() : val(0), next(nullptr) {}
7  *   ListNode(int x) : val(x), next(nullptr) {}
8  *   ListNode(int x, ListNode *next) : val(x), next(next) {}
9  * };
10 */
11 class Solution {
12 public:
13     ListNode* deleteDuplicates(ListNode* head) {
14         ListNode dummy(0);
15         dummy.next = head;
16         ListNode* prev = &dummy;
17     }
18 }
```

Saved Ln 31, Col 6

📄 Testcase 🔄 Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

head =

[1,2,3,3,4,4,5]

Output

[1,2,5]

Expected

[1,2,5]

🔗 Contribute a testcase