Remove Duplicates from Sorted List

Abbos Aliboev 2023041080

2025.01.09

Problem Definition (1)

- Source: Leetcode - 83
- Title: Remove Duplicates from Sorted List
- Difficulty: Easy

Linked List, Data Structer

Problem Definition (2)

Given the head of a sorted linked list, delete all duplicates such that each element appears only once. Return the linked list sorted as well.

Example 1:
Input: head = [1,1,2]
Output: [1,2]

Example 2:
Input: head = [1,1,2,3,3]
Output: [1,2,3]

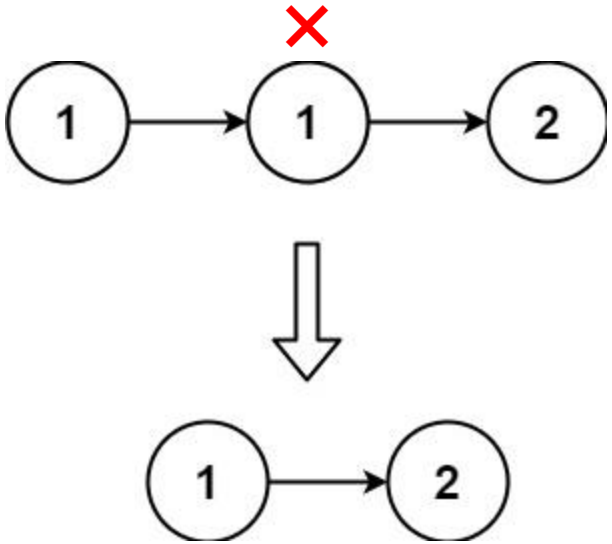
Constraints:

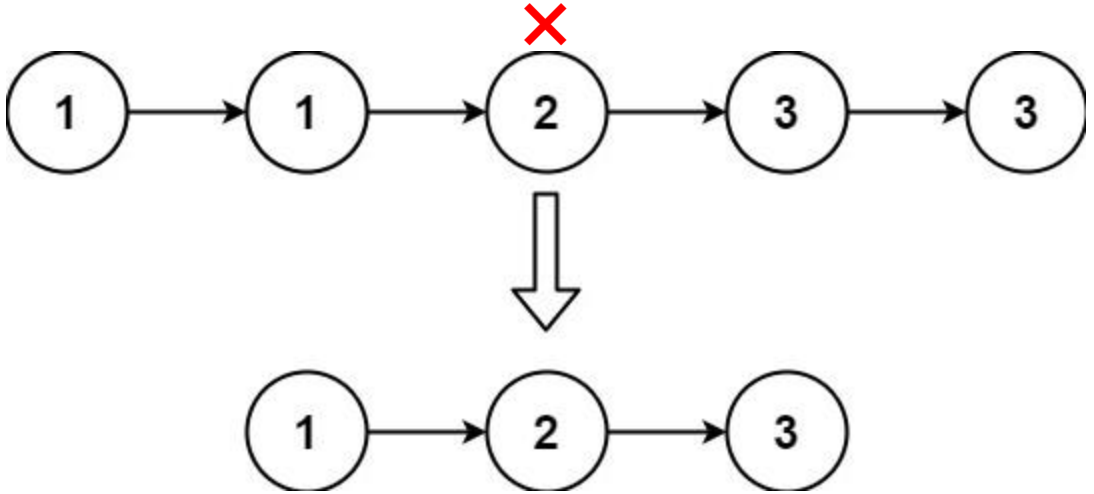
- The number of nodes in the list is in the range [0, 300].
- -100 <= Node.val <= 100
- The list is guaranteed to be sorted in ascending order.

Solution (1-0)

1. **Check if the list is empty:** If `head` is `nullptr`, return it immediately.
2. **Traverse the list:** Use `current = head` and move through the list while `current->next` is not `nullptr`.
3. **Remove duplicates:** If `current->val` is the same as `current->next->val`, then link `current->next` to `current->next->next` (skipping the duplicate). Otherwise, just move to the next node.
4. **Return the updated list:** Once done, return `head`.

Solution (1-1)





Solution (2-0)

Code

C++ Auto

```cpp
/**
 * 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) {}
 * };
 */

class Solution {
public:
    ListNode* deleteDuplicates(ListNode* head) {
        if (head == NULL) return NULL;

        ListNode* current = head;

        while(current->next != NULL){
            if(current->val == current->next->val){
                current->next = current->next->next;
            }
            else{
                current = current->next;
            }
        }
        return head;
    }
};
```

Testcase | Test Result

Case 1 Case 2 +

head =

[1,1,2]

Source

Testcase | Test Result

Case 1 Case 2 +

head =

[1,1,2,3,3]

Source

What you have learned

1. A **linked list** is a **data structure** where each node holds a value and a pointer to the next node.

2. Unlike arrays, **linked lists** allow efficient insertion/deletion by just adjusting pointers.

3. I learned how to **remove duplicates** from a sorted linked list by traversing and skipping duplicate nodes.

4. This practice deepened our understanding of pointer manipulation and fundamental **data structure** operations.

Questions and Answers

Greetings