Problem 1836: Remove Duplicates From an Unsorted Linked List

Problem Information

Difficulty: Medium
Acceptance Rate: 75.58%
Paid Only: Yes
Tags: Hash Table, Linked List

Problem Description

Given the `head` of a linked list, find all the values that appear **more than once** in the list and delete the nodes that have any of those values.

Return _the linked list after the deletions._

Example 1:

Input: head = [1,2,3,2] **Output:** [1,3] **Explanation:** 2 appears twice in the linked list, so all 2's should be deleted. After deleting all 2's, we are left with [1,3].

Example 2:

Input: head = [2,1,1,2] **Output:** [] **Explanation:** 2 and 1 both appear twice. All the elements should be deleted.

Example 3:

Input: head = [3,2,2,1,3,2,4] **Output:** [1,4] **Explanation:** 3 appears twice and 2 appears three times. After deleting all 3's and 2's, we are left with [1,4].

Constraints:

* The number of nodes in the list is in the range `[1, 105]` * `1 <= Node.val <= 105`

Code Snippets

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) {}
* };
*/
class Solution {
public:
ListNode* deleteDuplicatesUnsorted(ListNode* head) {

}
};
```

Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
```

```
public ListNode deleteDuplicatesUnsorted(ListNode head) {

}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
# def __init__(self, val=0, next=None):
# self.val = val
# self.next = next
class Solution:
def deleteDuplicatesUnsorted(self, head: ListNode) -> ListNode:
```