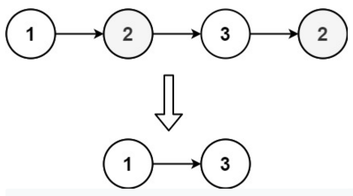
1836. Remove Duplicates From an Unsorted Linked List

Medium 125 4 Add to List Share

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.

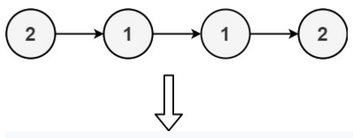
Return *the linked list after the deletions.*

Example 1:



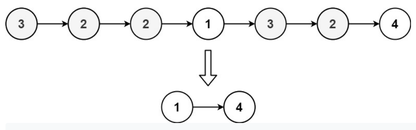
```
Input: head = [1,2,3,2]
Output: [1,3]
Explanation: 2 appears twice in the linked list, so all 2's should be deleted.
After deleting all 2's, we are left with [1,3].
```

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, 10^5]`
- `1 <= Node.val <= 10^5`

Accepted 7,289 | Submissions 10,506

Seen this question in a real interview before? Yes No

Companies

0 ~ 6 months 6 months ~ 1 year 1 year ~ 2 years

Goldman Sachs | 2

Related Topics

Hash Table Linked List

Similar Questions

Remove Duplicates from Sorted List II Medium

Remove Duplicates from Sorted List Easy

Hide Hint 1

Is there a way we can know beforehand which nodes to delete?

Hide Hint 2

Count the number of appearances for each number.

```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) {

    }
}
```