Description | Editorial | Submissions | Solutions | Accepted

1836. Remove Duplicates From an Unsorted Linked List

Premium

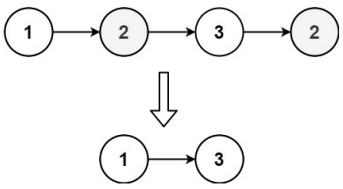
Solved

Medium | Topics | Companies | Hint

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.

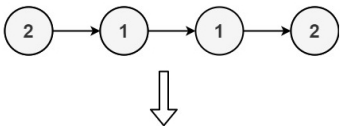
Return *the linked list after the deletions.*

Example 1:



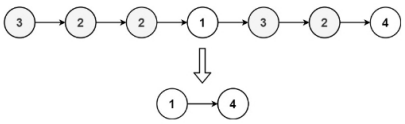

```
Input: head = [1,2,3,2]
Output: [1,3]
Explanation: 2 appears twice in the linked list, so all 2's should be deleted. After deleting all 2's, we are left with [1,3].
```

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

Seen this question in a real interview before? 1/5

Yes | No

Accepted 28.1K | Submissions 39.2K | Acceptance Rate 71.8%

Topics

Companies

Hint 1

Hint 2

Similar Questions

Discussion (6)

373 | 6

Code

Python3 | Auto

```python
# 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:

        dictx = {}

        curr = head

        while curr:
            val = curr.val
            if val not in dictx:
                dictx[val] = 1
            else:
                dictx[val] += 1
            curr = curr.next

        dummy = ListNode(0,head)
        slow = dummy
        curr = head

```

Saved | Ln 24, Col 1

Testcase | Test Result

Accepted Runtime: 42 ms

Case 1 | Case 2 | Case 3

Input

head =

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

Output

[1,4]