Description | Editorial | Solutions | Submissions

3063. Linked List Frequency

Premium

Solved

Easy | Topics | Hint

Given the `head` of a linked list containing `k` **distinct** elements, return *the head to a linked list of length* `k` *containing the frequency of each* ***distinct*** *element in the given linked list in* ***any order***.

Example 1:

> **Input:** head = [1,1,2,1,2,3]
>
> **Output:** [3,2,1]
>
> **Explanation:** There are `3` distinct elements in the list. The frequency of `1` is `3`, the frequency of `2` is `2` and the frequency of `3` is `1`. Hence, we return `3 -> 2 -> 1`.
>
> Note that `1 -> 2 -> 3`, `1 -> 3 -> 2`, `2 -> 1 -> 3`, `2 -> 3 -> 1`, and `3 -> 1 -> 2` are also valid answers.

Example 2:

> **Input:** head = [1,1,2,2,2]
>
> **Output:** [2,3]
>
> **Explanation:** There are `2` distinct elements in the list. The frequency of `1` is `2` and the frequency of `2` is `3`. Hence, we return `2 -> 3`.

Example 3:

> **Input:** head = [6,5,4,3,2,1]
>
> **Output:** [1,1,1,1,1,1]
>
> **Explanation:** There are `6` distinct elements in the list. The frequency of each of them is `1`. Hence, we return `1 -> 1 -> 1 -> 1 -> 1 -> 1`.

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 8.3K | Submissions 9.6K | Acceptance Rate 86.7%

Topics

Hint 1

Similar Questions

Discussion (10)

33 | 10

Code

Python3 • Auto

```python
# class ListNode:
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
class Solution:
    def frequenciesOfElements(self, head: Optional[ListNode]) -> Optional[ListNode]:

        # Dictionary to store the frequency of each element
        frequency_map = collections.defaultdict(int)

        # Traverse the original linked list and count the frequency of each element
        current_node = head

        while current_node:
            frequency_map[current_node.val] += 1
            current_node = current_node.next

        # Create a new linked list to store the frequencies
        dummy = ListNode()
        new_linked_list = dummy
        # Iterate through the frequency map and add each frequency to the new linked list
        for val, freq in frequency_map.items():
            new_linked_list.next = ListNode(freq)
            new_linked_list = new_linked_list.next

        # Return the head of the new linked list
        return dummy.next

```

Saved

Ln 12, Col 70

Testcase | Test Result

Accepted Runtime: 25 ms

Case 1

Input

head =
[1,1,2,1,2,3]

Output