3063. Linked List Frequency Premium

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 **10.7K** | Submissions **12.5K** | Acceptance Rate **85.7%**

Topics

Hash Table Linked List Counting

Hint 1

Traverse the linked list and keep the number of occurrences of values using a HashMap.

Similar Questions

Top K Frequent Elements Medium

Discussion (10)