817. Linked List Components

Medium 301 812 Add to List Share

We are given `head`, the head node of a linked list containing **unique integer values**.

We are also given the list `G`, a subset of the values in the linked list.

Return the number of connected components in `G`, where two values are connected if they appear consecutively in the linked list.

Input:
head: 0->1->2->3
G = [0, 1, 3]
Output: 2
Explanation:
0 and 1 are connected, so [0, 1] and [3] are the two connected components.

Input:
head: 0->1->2->3->4
G = [0, 3, 1, 4]
Output: 2
Explanation:
0 and 1 are connected, 3 and 4 are connected, so [0, 1] and [3, 4] are the two connected components.

agar mai hashMap mai present hu and mere sath wala nahi hai to mai ek component hu so count++ krna hai and next node ke liye move krjao.

```
int numComponents(ListNode *head, vector<int> &G)
{
    unordered_set<int> map;
    for (const int &ele : G)
        map.insert(ele);

    int count = 0;
    while (head != nullptr)
    {
        if (map.find(head->val) != map.end() &&
            (head->next == nullptr || map.find(head->next->val) == map.end()))
            count++;
        head = head->next;
    }
    return count;
}
```

agar mai present hu

next nahi to count ko increase krdo. and count se start hai to null manage hai.