链表分割段数\_LeetCode\_817\_LinkedListComponents\_Medium

# 链表分割段数\_LeetCode\_817\_LinkedListComponents\_Medium

## 题目介绍

\* 链表分割段数\_LeetCode\_817\_LinkedListComponents\_Medium

\* 难度：Medium

\* DateTime:2018-10-15

\* https://leetcode.com/problems/linked-list-components/description/

\* <p>

\* 题目介绍：

\* 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.

\* <p>

\* Example 1:

\* 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.

\* <p>

\* Example 2:

\* 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.

\* <p>

\* Note:

\* If N is the length of the linked list given by head, 1 <= N <= 10000.

\* The value of each node in the linked list will be in the range [0, N - 1].

\* 1 <= G.length <= 10000.

\* G is a subset of all values in the linked list.

## 思路分析

\* 思路分析：利用HashSet+连续标志实现

\* S1. 将数组转换为HashSet；

\* S2. 遍历链表，统计段数； 设置一个标志位isContinuous表示是否与上一个值连续。

\* while循环流程：如果set集合中包含当前节点的值，若与上一节点的值不连续，则设置isContinuous为true,并count++；

\* 如果集合set中不包含当前节点的值，且isContinuous是true,此时需要设置为false;从集合set中删除该值，并继续下一个节点。

## Java代码

public int **numComponents**(ListNode head, int[] G) {

if (head == null) return 0;

//S1. 将数组转换为HashSet

HashSet<Integer> set = new HashSet<Integer>();

for (int item : G)

set.add(item);

//S2. 遍历链表，统计段数

int count = 0;

boolean isContinuous = false;

while (head != null) {

if (set.contains(head.val)) {

if (!isContinuous) {

isContinuous = true;

count++;

}

} else {

if (isContinuous) {

isContinuous = false;

}

}

set.remove(head.val);

head = head.next;//不要忘记

}

return count;

}