Problem 86: Partition List

Problem Information

Difficulty: Medium
Acceptance Rate: 60.04%
Paid Only: No
Tags: Linked List, Two Pointers

Problem Description

Given the `head` of a linked list and a value `x`, partition it such that all nodes **less than** `x` come before nodes **greater than or equal** to `x`.

You should **preserve** the original relative order of the nodes in each of the two partitions.

Example 1:

Input: head = [1,4,3,2,5,2], x = 3 **Output:** [1,2,2,4,3,5]

Example 2:

Input: head = [2,1], x = 2 **Output:** [1,2]

Constraints:

* The number of nodes in the list is in the range `[0, 200]`. * `-100 <= Node.val <= 100` * `-200 <= x <= 200`

Code Snippets

C++:

```
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode() : val(0), next(nullptr) {}
* ListNode(int x) : val(x), next(nullptr) {}
* ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
ListNode* partition(ListNode* head, int x) {

}
};
```

Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
public ListNode partition(ListNode head, int x) {

}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
# def __init__(self, val=0, next=None):
# self.val = val
# self.next = next
class Solution:
```

```
def partition(self, head: Optional[ListNode], x: int) -> Optional[ListNode]:
```