86. Partition List

Description | Hints | Submissions | Discuss | Solution

Pick One

Given 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:

```
Input: head = 1->4->3->2->5->2, x = 3
Output: 1->2->2->4->3->5
```

```java
public class L86 {

    class ListNode{
        int val;
        ListNode next;
        public ListNode(int val) {
            this.val = val;
        }
    }
    //这道题目的意思是给定一个x的值，小于x都放在大于等于x的前面，并且不改变链表之间node原始的相对位置
    //new两个新链表，一个用来创建所有大于等于x的链表，一个用来创建所有小于x的链表，遍历整个链表时，当当前node的val小于x时，接在小链表上
    //反之，接在大链表上，这样就保证了相对顺序没有改变，而仅仅对链表做了与x的比较判断。最后，把小链表接在大链表上，别忘了把大链表的结尾赋为null
      public ListNode partition(ListNode head, int x) {
          if(head==null || head.next == null)
              return head;
          ListNode small = new ListNode(-1);
          ListNode newsmallhead = small;
          ListNode big = new ListNode(-1);
          ListNode newbighead = big;
          while (head != null) {
            if(head.val < x) {
                small.next = head;
                small = small.next;
            }else {
                big.next = head;
                big = big.next;
            }
            head = head.next;
        }
          big.next = null;
          small.next = newbighead.next;
          return newsmallhead.next;
      }
    }
```