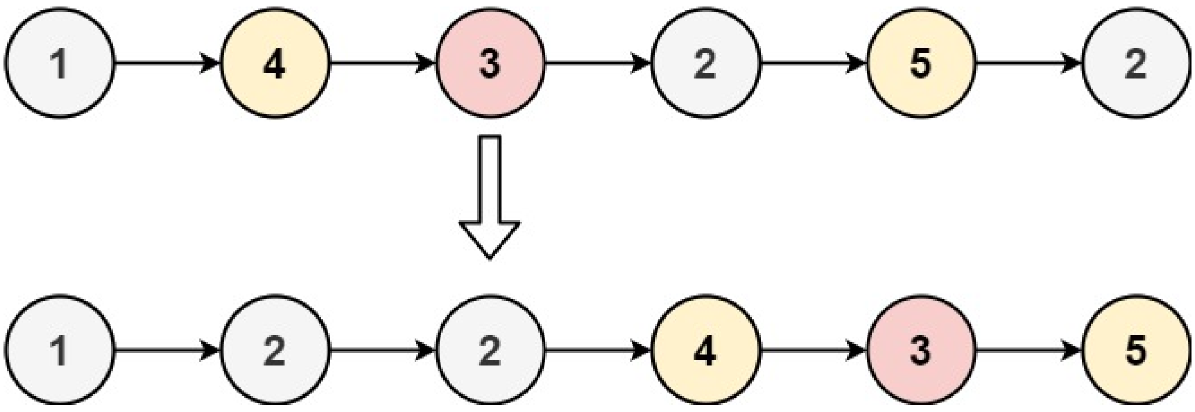
86 分隔链表

```
Label：链表

给你一个链表的头节点 head 和一个特定值 x ，请你对链表进行分隔，使得所有 小于 x 的节点都出现在
大于或等于 x 的节点之前。

你应当 保留 两个分区中每个节点的初始相对位置。
```



- 双端队列 创建新链表

```java
class Solution {
    public ListNode partition(ListNode head, int x) {

        Deque<Integer> small = new ArrayDeque<>();
        Deque<Integer> big = new ArrayDeque<>();
        ListNode curr = head;
        while (curr != null) {
            if (curr.val < x) {
                small.addLast(curr.val);
            }else {
                big.addLast(curr.val);
            }
            curr = curr.next;
        }

        // 还原链表
        ListNode newHead = new ListNode();
        curr = newHead;
        while (!small.isEmpty()) {
            curr.next = new ListNode(small.pollFirst());
            curr = curr.next;
        }
        while (!big.isEmpty()) {
            curr.next = new ListNode(big.pollFirst());
            curr = curr.next;
        }
        return newHead.next;
    }
}
```

- 双链表 虚拟头结点

```java
class Solution {
    public ListNode partition(ListNode head, int x) {
        // 直接定义两个虚拟头结点
        ListNode small = new ListNode(0);
        ListNode smallHead = small;
        ListNode large = new ListNode(0);
        ListNode largeHead = large;

        while (head != null) {
            if (head.val < x) {
                small.next = head;
                small = small.next;
            } else {
                large.next = head;
                large = large.next;
            }
            head = head.next;
        }
        // 切断两个链表的尾结点，防止出现循环链表
        large.next = null;
        small.next = largeHead.next;
        return smallHead.next;
    }
}
```