

143. Reorder List

Description

Hints

Submissions

Discuss

Solution

Pick One

Given a singly linked list $L: L_0 \rightarrow L_1 \rightarrow \dots \rightarrow L_{n-1} \rightarrow L_n$,
reorder it to: $L_0 \rightarrow L_n \rightarrow L_1 \rightarrow L_{n-1} \rightarrow L_2 \rightarrow L_{n-2} \rightarrow \dots$

You may **not** modify the values in the list's nodes, only nodes itself may be changed.

Example 1:

Given 1->2->3->4, reorder it to 1->4->2->3.

Example 2:

Given 1->2->3->4->5, reorder it to 1->5->2->4->3.

```
* 这是个链表反转的问题
* 1.找到中间节点
* 2.将链表从中间断开
* 3.把后链表reverse一下,
* 4.再合并两个链表
*/
public class L143 {
    public class ListNode {
        int val;
        ListNode next;
        ListNode(int x) { val = x; }
    }

    public void reorderList(ListNode head) {
        if (head == null || head.next == null || head.next.next == null)
            return;
        ListNode slow = head;
        ListNode fast = head;
        //找到中间节点
        while (fast.next != null && fast.next.next != null) {
            fast = fast.next.next;
            slow = slow.next;
        }
        ListNode second = slow.next;
        //注意置空, 分为两个链表, 第一个链表的长度大于 (+1) 等于第二个链表的长度
        slow.next = null;
        //反转后的链表
        second = reverseList(second);
        ListNode first = head;
        //合并两个链表, 画图模拟, 把第二个链表插在第一个链表中
        while (second != null) {
            //暂存第一个后续节点
            ListNode next = first.next;
            first.next = second;
            second = second.next;
            first = first.next;
            first.next = next;
            first = first.next;
        }
    }

    //重点, 翻转链表
    private ListNode reverseList(ListNode head) {
        ListNode prev = null;
        ListNode next = null;
        while (head != null) {
            next = head.next;
            head.next = prev;
            prev = head;
            head = next;
        }
        return prev;
    }
}
```



合并链表
first $\square \rightarrow \square \rightarrow$

second $\square \rightarrow \square \rightarrow$

1. next = first

first $\square \rightarrow \square \xrightarrow{\text{next}} \square \rightarrow$

2. first.next = second

first $\square \downarrow$
second $\square \rightarrow \square \xrightarrow{\text{next}} \square \rightarrow$

3. first.next =

first $\square \downarrow$
second $\square \rightarrow \square \xrightarrow{\text{next}} \square \rightarrow$

3. second = second.next

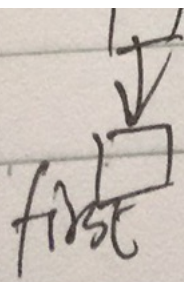
4. $\square \rightarrow \square \rightarrow$
second

4. first = first.next

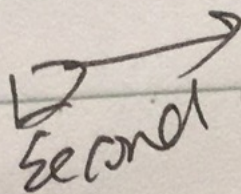
$\square \rightarrow \square \xrightarrow{\text{next}} \square \rightarrow$

5. first = first.next
first $\square \downarrow$
second $\square \rightarrow \square \xrightarrow{\text{next}} \square \rightarrow$

first

A hand-drawn diagram consisting of a vertical rectangle. Above the rectangle is a downward-pointing arrow, formed by two short vertical lines and a horizontal line connecting them at the top.

second

A hand-drawn diagram consisting of a horizontal rectangle. Above the rectangle is a rightward-pointing arrow, formed by two short horizontal lines and a vertical line connecting them at the right end.