```
/**
 * Definition for singly-linked list.
 * class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode(int x) {
 *         val = x;
 *         next = null;
 *     }
 * }
 */
public class Solution {
    public ListNode sortList(ListNode head) {
                if (head == null || head.next == null)
                        return head;

                // count total number of elements
                int count = 0;
                ListNode p = head;
                while (p != null) {
                        count++;
                        p = p.next;
                }

                // break up to two list
                int middle = count / 2;

                ListNode l = head, r = null;
                ListNode p2 = head;
                int countHalf = 0;
                while (p2 != null) {
                        countHalf++;
                        ListNode next = p2.next;

                        if (countHalf == middle) {
                                p2.next = null;
                                r = next;
                        }
                        p2 = next;
                }

                // now we have two parts l and r, recursively sort them
                ListNode h1 = sortList(l);
                ListNode h2 = sortList(r);

                // merge together
                ListNode merged = merge(h1, h2);
```

```
                return merged;
    }
    public static ListNode merge(ListNode l, ListNode r) {
                ListNode p1 = l;
                ListNode p2 = r;

                ListNode fakeHead = new ListNode(100);
                ListNode pNew = fakeHead;

                while (p1 != null || p2 != null) {

                        if (p1 == null) {
                                pNew.next = new ListNode(p2.val);
                                p2 = p2.next;
                                pNew = pNew.next;
                        } else if (p2 == null) {
                                pNew.next = new ListNode(p1.val);
                                p1 = p1.next;
                                pNew = pNew.next;
                        } else {
                                if (p1.val < p2.val) {
                                        // if(fakeHead)
                                        pNew.next = new ListNode(p1.val);
                                        p1 = p1.next;
                                        pNew = pNew.next;
                                } else if (p1.val == p2.val) {
                                        pNew.next = new ListNode(p1.val);
                                        pNew.next.next = new ListNode(p1.val);
                                        pNew = pNew.next.next;
                                        p1 = p1.next;
                                        p2 = p2.next;

                                } else {
                                        pNew.next = new ListNode(p2.val);
                                        p2 = p2.next;
                                        pNew = pNew.next;
                                }
                        }
                }

                // printList(fakeHead.next);
                return fakeHead.next;
        }
}
```