



链滴

## 算法 (6) 合并两个有序链表

作者: [ws](#)

原文链接: <https://ld246.com/article/1541646482561>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

将两个有序链表合并为一个新的有序链表并返回。新链表是通过拼接给定的两个链表的所有节点组成

。

### 示例:

输入: 1->2->4, 1->3->4

输出: 1->1->2->3->4->4

### 答案

```
class ListNode {
    int val;
    ListNode next;
    ListNode(int x) { val = x; }
}

class Solution {
    public static ListNode mergeTwoLists(ListNode l1, ListNode l2) {
        // 把原两个链表的内容放到第三个链表, 最后返回第三个链表的头
        ListNode l3 = new ListNode(0);
        ListNode head = l3;
        while (l1 != null && l2 != null) {
            if (l1.val <= l2.val) {
                ListNode newNode = new ListNode(l1.val);
                l3.next = newNode;
                l3 = l3.next;
                l1 = l1.next;
            } else {
                ListNode newNode = new ListNode(l2.val);
                l3.next = newNode;
                l3 = l3.next;
                l2 = l2.next;
            }
        }
        // 可能存在l1、l2偏大, 某一个链表没有比较完, 需要把剩下的加入到l3
        while (l1 != null) {
            ListNode newNode = new ListNode(l1.val);
            l3.next = newNode;
            l3 = l3.next;
            l1 = l1.next;
        }
        while (l2 != null) {
            ListNode newNode = new ListNode(l2.val);
            l3.next = newNode;
            l3 = l3.next;
            l2 = l2.next;
        }
        return head.next;
    }
}

public class Main {
    public static void main(String[] args) {
```

```
ListNode listNode1 = new ListNode(0);
ListNode listNode2 = new ListNode(1);
ListNode listNode3 = new ListNode(2);
ListNode listNode4 = new ListNode(3);
listNode1.next = listNode2;
listNode2.next = listNode3;
listNode3.next = listNode4;

ListNode listNode = Solution.mergeTwoLists(listNode2, listNode1);

while(listNode.next != null){
    System.out.println(listNode.val);
    listNode = listNode.next;
}
}
```