

5月16日报

本日学习内容

- 1. 开始复习学过的内容，复习类别与拓展，多态，协议与委托，深浅拷贝完成博客
- 2. 完成每日算法题

今日算法题

题目1: 148. 排序链表

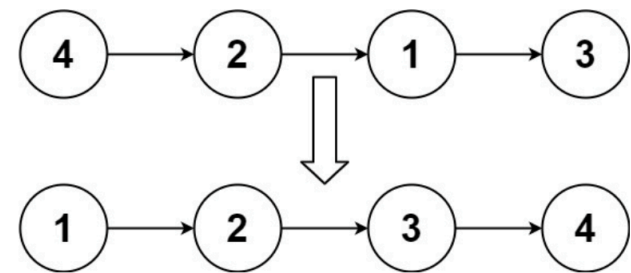
148. 排序链表

已解答

中等 相关标签 相关企业 Aa

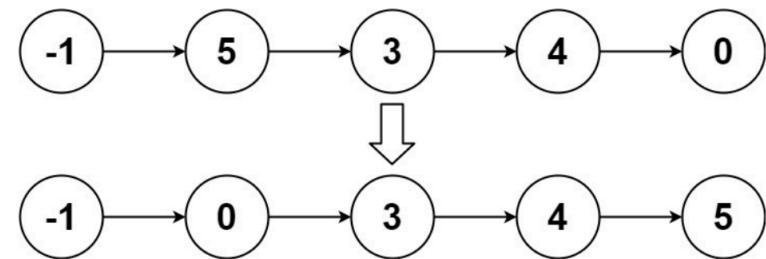
给你链表的头结点 `head`，请将其按 升序 排列并返回 排序后的链表。

示例 1:



输入: `head = [4,2,1,3]`
输出: `[1,2,3,4]`

示例 2:



输入: `head = [-1,5,3,4,0]`
输出: `[-1,0,3,4,5]`

示例 3:

输入: `head = []`
输出: `[]`

提示:

- 链表中节点的数目在范围 `[0, 5 * 104]` 内
- `-105 <= Node.val <= 105`

```
class Solution {
public:
```

```

ListNode* sortList(ListNode* head) {
    if (!head || !head->next)
        return head;

    // 找中点
    ListNode *slow = head, *fast = head->next;
    while (fast && fast->next) {
        slow = slow->next;
        fast = fast->next->next;
    }

    // 分割
    ListNode* mid = slow->next;
    slow->next = nullptr;

    // 递归排序左右部分
    ListNode* left = sortList(head);
    ListNode* right = sortList(mid);

    // 合并
    return merge(left, right);
}

ListNode* merge(ListNode* l1, ListNode* l2) {
    ListNode dummy(0), *tail = &dummy;
    while (l1 && l2) {
        if (l1->val < l2->val) {
            tail->next = l1;
            l1 = l1->next;
        } else {
            tail->next = l2;
            l2 = l2->next;
        }
        tail = tail->next;
    }
    tail->next = l1 ? l1 : l2;
    return dummy.next;
}
};

```

本日遇到的问题

1. 最初不太理解oc中协议与委托的存在意义

明日学习计划

1. 继续复习之前的内容，看学长学姐博客，查漏补缺
2. 将每日新学的内容整理为博客