

6月2日报

本日学习内容

- 1. 继续学习ui，学习多界面传值的几种方法
- 2. 继续总结博客

今日算法题

题目1: 148. 排序链表

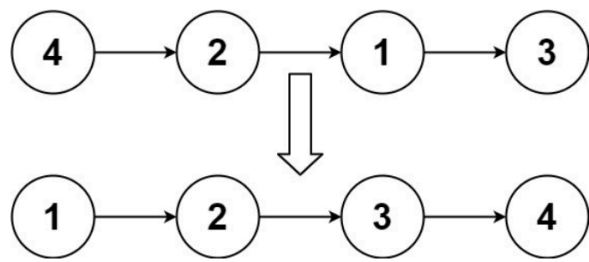
148. 排序链表

已解答

中等 相关标签 相关企业 Aa

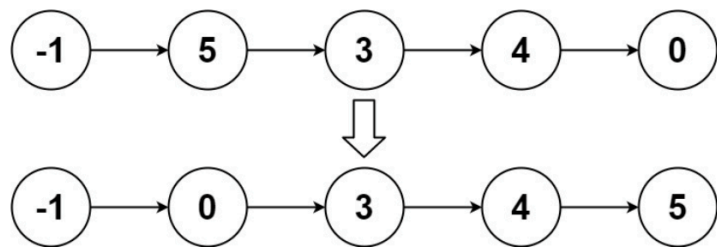
给你链表的头结点 `head`，请将其按升序排列并返回排序后的链表。

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`

进阶: 你可以在 `O(n log n)` 时间复杂度和常数级空间复杂度下, 对链表进行排序吗?

```
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;
    }
};

```

题目2: [94. 二叉树的中序遍历](#)

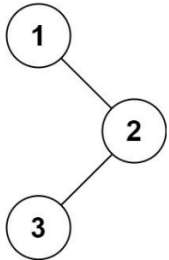
94. 二叉树的中序遍历

已解答 

简单 相关标签 相关企业 Az

给定一个二叉树的根节点 `root`，返回 它的 *中序* 遍历。

示例 1:



输入: `root = [1,null,2,3]`
输出: `[1,3,2]`

示例 2:

输入: `root = []`
输出: `[]`

示例 3:

输入: `root = [1]`
输出: `[1]`

提示:

- 树中节点数目在范围 `[0, 100]` 内
- `-100 <= Node.val <= 100`

进阶: 递归算法很简单, 你可以通过迭代算法完成吗?

面试中遇到过这道题? 1/5

是 否

通过次数 1,811,279/2.3M | 通过率 78.2 %

相关标签

```
class Solution {
public:
    vector<int> inorderTraversal(TreeNode* root) {
        vector<int> result;
        stack<TreeNode*> st;
        TreeNode* cur = root;
        while (cur != NULL || !st.empty()) {
            if (cur != NULL) { // 指针来访问节点，访问到最底层
                st.push(cur); // 将访问的节点放进栈
                cur = cur->left;
            } else {
                cur = st.top(); // 从栈里弹出的数据，就是要处理的数据（放进result数组里的数据）
                st.pop();
                result.push_back(cur->val);
                cur = cur->right;
            }
        }
        return result;
    }
};
```

```
        }  
    }  
    return result;  
}  
};
```

本日遇到的问题

1. UI导航栏和之前的部分内容还不能完全掌握，写程序时还需要看之前写过的程序或笔记
2. 初学多种传值还是有点模糊，需要再深入研究一下

明日学习计划

1. 复习之前学过的控件，继续向后学习UITableView