

5月2日报

本日学习内容

1. 每日算法题
2. 提交博客到git
3. 学习完成隐藏和封装的学习

今日算法题

题目1:[对链表进行插入排序](#)

147. 对链表进行插入排序

已解答

中等

🏷 相关标签

🔒 相关企业

Ax

给定单个链表的头 `head`，使用 **插入排序** 对链表进行排序，并返回 *排序后链表的头*。

插入排序 算法的步骤：

1. 插入排序是迭代的，每次只移动一个元素，直到所有元素可以形成一个有序的输出列表。
2. 每次迭代中，插入排序只从输入数据中移除一个待排序的元素，找到它在序列中适当的位置，并将其插入。
3. 重复直到所有输入数据插入完为止。

下面是插入排序算法的一个图形示例。部分排序的列表(黑色)最初只包含列表中的第一个元素。每次迭代时，从输入数据中删除一个元素(红色)，并就地插入已排序的列表中。

对链表进行插入排序。

3 5 6 1 8 7 2 4

```
void insertionSort(int* arr, int n) {  
    for (int i = 1; i < n; i++) {  
        int key = arr[i];
```

```

        int j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key;
    }
}

struct ListNode* insertionSortList(struct ListNode* head) {
    if (head == NULL) {
        return NULL;
    }
    int cnt = 0;
    struct ListNode* cur = head;
    while (cur != NULL) {
        cnt++;
        cur = cur->next;
    }

    int* arr = (int *)malloc(sizeof(int) * cnt);
    cur = head;
    for (int i = 0; i < cnt; i++) {
        arr[i] = cur->val;
        cur = cur->next;
    }
    insertionSort(arr, cnt);
    struct ListNode* newHead = NULL;
    struct ListNode* tail = NULL;
    for (int i = 0; i < cnt; i++) {
        struct ListNode* node = (struct ListNode *)malloc(sizeof(struct ListNode));
        node->val = arr[i];
        node->next = NULL
        if (newHead == NULL) {
            newHead = node;
            tail = node;
        } else {
            tail->next = node;
            tail = node;
        }
    }
    return newHead;
}

```

题目二： [最大子数组和](#)

53. 最大子数组和

已解答

中等

🏷 相关标签

🔒 相关企业

Aa

给你一个整数数组 `nums`，请你找出一个具有最大和的连续子数组（子数组最少包含一个元素），返回其最大和。

子数组是数组中的一个连续部分。

示例 1：

输入：nums = [-2,1,-3,4,-1,2,1,-5,4]
输出：6
解释：连续子数组 [4,-1,2,1] 的和最大，为 6 。

示例 2：

输入：nums = [1]
输出：1

示例 3：

输入：nums = [5,4,-1,7,8]
输出：23

提示：

```
#define MAX(a, b) ((a) > (b) ? (a) : (b))

int maxSubArray(int* nums, int numsSize) {
    int maxSum = INT_MIN;
    int sum = 0;

    for (int i = 0; i < numsSize; i++) {
        if (sum < 0) {
            sum = 0;
        }
        sum += nums[i];
        maxSum = MAX(maxSum, sum);
    }

    return maxSum;
}
```

本日遇到的问题

1. 没弄懂背包问题的一维滚动数组逻辑
2. 自己学oc各种函数脑子混乱，应看看学长博客理思路

明日学习计划

1. 每日两道题
2. 完成对象初始化与类的继承的学习，及多态的学习