5月12日日报

本日学习内容

- 1. 学习了第六章字符串, 日期与时间
- 2. 复习了昨天没理清的类别中的部分内容
- 3. 每日算法题

今日算法题

题目1: 找出临界点之间的最小和最大距离

2058. 找出临界点之间的最小和最大距离

已解答 🕑

中等 ♥ 相关标签 A 相关企业 ② 提示 A A

链表中的 临界点 定义为一个 局部极大值点 或 局部极小值点。

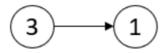
如果当前节点的值 严格大于 前一个节点和后一个节点,那么这个节点就是一个 局部极大值点。

如果当前节点的值 严格小于 前一个节点和后一个节点,那么这个节点就是一个 局部极小值点。

注意: 节点只有在同时存在前一个节点和后一个节点的情况下,才能成为一个 局部极大值点/极小值点。

给你一个链表 head ,返回一个长度为 2 的数组 [minDistance, maxDistance] ,其中 minDistance 是任意 两个不同临界点之间的最小距离, maxDistance 是任意两个不同临界点之间的最大距离。如果临界点少于两 个,则返回 [-1,-1]。

示例 1:



输入: head = [3,1]

输出: [-1,-1]

解释: 链表 [3,1] 中不存在临界点。

示例 2:

```
int* nodesBetweenCriticalPoints(struct ListNode* head, int* returnSize) {
   *returnSize = 2;
   struct ListNode* pre = head;
   struct ListNode* next;
```

```
struct ListNode* cur = head->next;
   int cnt = 1;
   int* arr = (int *)malloc(sizeof(int) * 2);
   int min = INT MAX;
   int prepoint = -1;
   int left = -1, right = -1;
   while (cur != NULL && cur->next != NULL) {
       next = cur->next;
       if ((pre->val < cur->val && next->val < cur->val)
        ||(pre->val > cur->val && next->val > cur->val)) {
            if (left == -1) {
               left = cnt;
            } else {
                if (min > cnt - prepoint) {
                   min = cnt - prepoint;
               right = cnt;
           prepoint = cnt;
       pre = cur;
       cur = next;
       cnt++;
   if (left == -1 | right == -1) {
        arr[0] = -1;
       arr[1] = -1;
   } else {
       arr[0] = min;
       arr[1] = right - left;
   return arr;
}
```

本日遇到的问题

1. 字符串与日期中函数太多导致容易遗忘

明日学习计划

- 1. 每日算法题
- 2. 第七章的复制与数组的部分内容