

6月8日日报

本日学习内容

1. 继续完成ZARA仿写，完成自定义cell部分的“我的”内容
2. 继续总结博客

今日算法题

题目1: [300. 最长递增子序列](#)

300. 最长递增子序列

已解答 ✓

中等 相关标签 相关企业 Aa

给你一个整数数组 `nums`，找到其中最长严格递增子序列的长度。

子序列 是由数组派生而来的序列，删除（或不删除）数组中的元素而不改变其余元素的顺序。例如，`[3,6,2,7]` 是数组 `[0,3,1,6,2,2,7]` 的**子序列**。

示例 1：

输入：nums = [10,9,2,5,3,7,101,18]

输出：4

解释：最长递增子序列是 [2,3,7,101]，因此长度为 4。

示例 2：

输入：nums = [0,1,0,3,2,3]

输出：4

示例 3：

输入：nums = [7,7,7,7,7,7,7]

输出：1

提示：

- `1 <= nums.length <= 2500`
- `-104 <= nums[i] <= 104`

进阶：

- 你能将算法的时间复杂度降低到 `O(n log(n))` 吗？

```
class Solution {
public:
    int lengthOfLIS(vector<int>& nums) {
        int len = 1, n = (int)nums.size();
        if (n == 0) {
            return 0;
        }
    }
```

```

    }
    vector<int> d(n + 1, 0);
    d[len] = nums[0];
    for (int i = 1; i < n; ++i) {
        if (nums[i] > d[len]) {
            d[++len] = nums[i];
        } else {
            int l = 1, r = len,
                pos = 0; // 如果找不到说明所有的数都比 nums[i]
                        // 大, 此时要更新 d[1], 所以这里将 pos 设为 0
            while (l <= r) {
                int mid = (l + r) >> 1;
                if (d[mid] < nums[i]) {
                    pos = mid;
                    l = mid + 1;
                } else {
                    r = mid - 1;
                }
            }
            d[pos + 1] = nums[i];
        }
    }
    return len;
}
};

```

题目2: [322. 零钱兑换](#)

322. 零钱兑换

已解答 

中等 相关标签 相关企业 Aa

给你一个整数数组 `coins`，表示不同面额的硬币；以及一个整数 `amount`，表示总金额。

计算并返回可以凑成总金额所需的 **最少的硬币个数**。如果没有任何一种硬币组合能组成总金额，返回 `-1`。

你可以认为每种硬币的数量是无限的。

示例 1:

输入: `coins = [1, 2, 5]`, `amount = 11`
输出: `3`
解释: $11 = 5 + 5 + 1$

示例 2:

输入: `coins = [2]`, `amount = 3`
输出: `-1`

示例 3:

输入: `coins = [1]`, `amount = 0`
输出: `0`

提示:

- $1 \leq \text{coins.length} \leq 12$
- $1 \leq \text{coins}[i] \leq 2^{31} - 1$
- $0 \leq \text{amount} \leq 10^4$

```
class Solution {
public:
    int coinChange(vector<int>& coins, int amount) {
        const int INF = amount + 1;
        vector<int> dp(amount + 1, INF);
        dp[0] = 0;

        for (int coin : coins) {
            for (int i = coin; i <= amount; ++i) {
                dp[i] = min(dp[i], dp[i - coin] + 1);
            }
        }

        return dp[amount] == INF ? -1 : dp[amount];
    }
};
```

本日遇到的问题

1. cell部分掌握不熟练，还需要再看看

明日学习计划

1. 继续完善项目，增加换头像改昵称功能
2. 继续思考之前的轮播代码，做到能掌握