

4月28日报

本日学习内容

1. 学习oc面向对象中类与对象的使用，self关键字，id类型，成员变量，模拟类变量部分
2. 继续完成hot 100动态规划部分

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

Ax

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 = [7,7,7,7,7,7,7,7]`

输出: 1

提示:

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

```
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;
};
```

每日问题

1. 最初未能理解面向对象和面向过程设计思想的区别
2. 初学时不了解oc语法, 最开始看书有些看不懂代码, 自己写也有些困难

明日学习计划

1. 学习oc面向对象上单例模式, 隐藏与封装, 对象初始化部分
2. 继续学习git操作, 分离head