

1-动态规划总结

1

动态规划

10

19

42

47

48 【动态规划】 【滑动窗口】

60

面试经典150题： 一维动态规划

打家劫舍

<https://leetcode.cn/problems/house-robber/description/?envType=study-plan-v2&envId=top-interview-150>

面试经典150题： 二维动态规划

三角形最小路径和（经典普通二维动规，但是有方法从下往上有点巧妙）

<https://leetcode.cn/problems/triangle/solutions/329143/san-jiao-xing-zui-xiao-lu-jing-he-by-leetcode-solu/?envType=study-plan-v2&envId=top-interview-150>

不同路径 II（）

<https://leetcode.cn/problems/unique-paths-ii/solutions/316968/bu-tong-lu-jing-ii-by-leetcode-solution-2/?envType=study-plan-v2&envId=top-interview-150>

编辑距离(好像在哪儿见过，很难)

<https://leetcode.cn/problems/edit-distance/description/?envType=study-plan-v2&envId=top-interview-150>

买卖股票（有个方法据说通用？） 从这里开始继续TODO 接雨水..等等完全搞懂2~3题

<https://leetcode.cn/problems/best-time-to-buy-and-sell-stock-iii/solutions/71209/tong-su-yi-dong-de-dong-tai-gui-hua-jie-fa-by-marc/?envType=study-plan-v2&envId=top-interview-150>

经典题型

【青蛙跳台阶】

【...】

TODO:网上随便找的DP大总结——

<https://github.com/youngyangyang04/leetcode-master/blob/master/problems/%E5%8A%A8%E6%80%81%E8%A7%84%E5%88%92%E6%80%BB%E7%BB%93%E7%AF%87.md>

<https://zhuanlan.zhihu.com/p/91582909>

b站教学视频

https://www.bilibili.com/video/BV1Yf4y1Z7Ac/?spm_id_from=pageDriver&vd_source=7d2082262d19abdd2e30ff5f60dc1fdf

相同算法，迭代比递归要更优化。

动态规划的实现方式有：记忆化搜索和递推。

用哈希数组或一些变量来记录运算结果，避免重复运算的方法就是记忆化搜索。

解题框架

<https://leetcode.cn/leetbook/read/illustration-of-algorithm/m5zfc1/>

动态规划特点

「分治」是算法中的一种基本思想，其通过将原问题分解为子问题，不断递归地将子问题分解为更小的子问题，并通过组合子问题的解来得到原问题的解。

类似于分治算法，「动态规划」也通过组合子问题的解得到原问题的解。不同的是，适合用动态规划解决的问题具有「重叠子问题」和「最优子结构」两大特性。

重叠子问题

（这是动态规划与dfs/回溯法的重要区别，如八皇后可以用dp，但没有重叠子问题，用了dp反而浪费空间。）

动态规划的子问题是有重叠的，即各个子问题中包含重复的更小子问题。若使用暴力法穷举，求解这些相同子问题会产生大量的重复计算，效率低下。

动态规划在第一次求解某子问题时，会将子问题的解保存；后续遇到重叠子问题时，则直接通过查表获取解，保证每个独立子问题只被计算一次，从而降低算法的时间复杂度。

最优子结构

如果一个问题的最优解可以由其子问题的最优解组合构成，并且这些子问题可以独立求解，那么称此问题具有最优子结构。（贪心算法?? 贪心要求每个子问题的最优解都是全局最优解的一部分）

动态规划从基础问题的解开始，不断迭代组合、选择子问题的最优解，最终得到原问题最优解。

（分析问题的状态时，不要分析整体，只分析最后一个阶段即可！因为动态规划问题都是划分为多个阶段的，各个阶段的状态表示都是一样，而我们的最终答案就是在最后一个阶段。）

二维数组的DP

据说80%的动态规划题目用二维数组。

据说90%的字符串问题都可以用动态规划解决，并且90%是采用二维数组。

<https://zhuanlan.zhihu.com/p/91582909>

背包问题——包含动态规划五部曲!!!

<https://leetcode.cn/circle/discuss/1Uki6J/>

01 背包问题

方法一：暴力法

每一件物品都有2种状态，取和不取。用回溯法，建立二叉树，深度为物品的数量，这样能遍历所有情况。时间 $O(2^n)$ ，指数级别、

方法二：动态规划

第一步：确定dp数组和下标的含义

第二步：确定递归公式

第三步：dp数组如何初始化

第四步：确定遍历顺序

第五步：举例推倒dp数组

第一步：dp[i][j]的含义，表述从下标为[0-i]的物品里任意取，放进容量为j的背包，价值总和最大是多少。

dp[i][j]

背包重量j:

	0	1	2	3	4
物品0:	0				
物品1:	0				
物品2:	0				


代码随想录

1

数位DP(有模版)

一般是统计一个区间[le,ri]内满足一些条件数的个数，在数的位数上做dp

https://www.bilibili.com/video/BV1rS4y1s721/?t=1581.4&vd_source=7d2082262d19abdd2e30ff5f60dc1fdf

(集合和二进制一一对应)

1

估计不太会考察的其他技巧：先不看

经典斐波那契数列还有一种矩阵快速幂解法，cao! (动态规划 $O(n)$, 快速幂 $O(\log n)$)

算法学习笔记(4): 快速幂 - Pecco的文章 - 知乎

<https://zhuanlan.zhihu.com/p/95902286>

