从递归入手一维动态规划

前置知识:
需要熟悉递归，如果不熟悉如下课程都涉及递归，
讲解017、讲解020、讲解021、讲解022、讲解023
讲解036、讲解037、讲解038、讲解039、讲解040
关于取模，
讲解041-同余原理

本节课我们从基本递归入手，来了解一维动态规划

注意：
重叠子问题？最优子结构？无后效性？
此时谈这些太早，【必备】阶段动态规划的大总结，将在动态规划专题结束时进行

左程云

从递归入手一维动态规划

动态规划：用空间代替重复计算，包含一整套原理和技巧的总和，课程会用非常大的篇幅来全盘介绍

知道怎么算的算法 vs 知道怎么试的算法

有些递归在展开计算时，总是重复调用同一个子问题的解，这种重复调用的递归变成动态规划很有收益
如果每次展开都是不同的解，或者重复调用的现象很少，那么没有改动态规划的必要
下节课会举例，哪些递归没有必要改动态规划的必要

任何动态规划问题都一定对应着一个有重复调用行为的递归
所以任何动态规划的题目都一定可以从递归入手，逐渐实现动态规划的方法
题目1到题目4，都从递归入手，逐渐改出动态规划的实现

尝试策略 就是 转移方程，完全一回事！
推荐从尝试入手，因为代码好写，并且一旦发现尝试错误，重新想别的递归代价轻！

从递归入手一维动态规划

当熟悉了从递归到动态规划的转化过程
那么就可以纯粹用动态规划的视角来分析问题了
题目5到题目8，都是纯粹用动态规划的视角来分析、优化的

如果不熟悉这个过程，直接一上来就硬去理解状态转移方程
那么往往会步履维艰、邯郸学步、东施效颦
这是多年教学看到的真实情况

很多极为优异的想法、设计和优化 来自 努力 or 天赋

建议脚踏实地，真正做好从递归到动态规划的练习
接下来的几节课也都会从最基本递归入手，逐渐写出动态规划的版本

左程云

从递归入手一维动态规划

动态规划的大致过程：
想出设计优良的递归尝试(方法、经验、固定套路很多)，有关尝试展开顺序的说明
-> 记忆化搜索(从顶到底的动态规划)，如果每个状态的计算枚举代价很低，往往到这里就可以了
-> 严格位置依赖的动态规划(从底到顶的动态规划)，更多是为了下面说的 进一步优化枚举做的准备
-> 进一步优化空间（空间压缩），一维、二维、多维动态规划都存在这种优化
-> 进一步优化枚举也就是优化时间（本节没有涉及，但是后续巨多内容和这有关）

解决一个问题，可能有很多尝试方法
众多的尝试方法中，可能若干的尝试方法有重复调用的情况，可以转化成动态规划
若干个可以转化成动态规划的方法中，又可能有优劣之分
判定哪个是最优的动态规划方法，依据来自题目具体参数的数据量
最优的动态规划方法实现后，后续又有一整套的优化技巧

本系列课程从【必备】到【扩展】到【挺难】都会讲动态规划，会把这一话题做全面的讲述

从递归入手一维动态规划

题目1
斐波那契数
斐波那契数 （通常用 F(n) 表示）形成的序列称为 斐波那契数列
该数列由 0 和 1 开始，后面的每一项数字都是前面两项数字的和。
也就是：F(0) = 0，F(1) = 1
F(n) = F(n - 1) + F(n - 2)，其中 n > 1
给定 n ，请计算 F(n)
测试链接 : https://leetcode.cn/problems/fibonacci-number/

注意：
斐波那契数问题最经典，本节课讲述的方法时间复杂度O(n)
但是最优解来自矩阵快速幂，时间复杂度可以做到O(log n)
矩阵快速幂后续课程一定会讲述！本节课不再展开

从递归入手一维动态规划

题目2
最低票价
在一个火车旅行很受欢迎的国度，你提前一年计划了一些火车旅行
在接下来的一年里，你要旅行的日子将以一个名为 days 的数组给出
每一项是一个从 1 到 365 的整数
火车票有 三种不同的销售方式
一张 为期1天 的通行证售价为 costs[0] 美元
一张 为期7天 的通行证售价为 costs[1] 美元
一张 为期30天 的通行证售价为 costs[2] 美元
通行证允许数天无限制的旅行
例如，如果我们在第 2 天获得一张 为期 7 天 的通行证
那么我们可以连着旅行 7 天(第2~8天)
返回 你想要完成在给定的列表 days 中列出的每一天的旅行所需要的最低消费
测试链接：https://leetcode.cn/problems/minimum-cost-for-tickets/

从递归入手一维动态规划

题目3
解码方法
一条包含字母 A-Z 的消息通过以下映射进行了 编码 ：
'A' -> "1"、'B' -> "2" ...'Z' -> "26"
要 解码 已编码的消息，所有数字必须基于上述映射的方法，反向映射回字母（可能有多种方法）
例如，"11106" 可以映射为："AAJF"、"KJF"
注意，消息不能分组为(1 11 06)，因为 "06" 不能映射为 "F"
这是由于 "6" 和 "06" 在映射中并不等价
给你一个只含数字的 非空 字符串 s ，请计算并返回 解码 方法的 总数
题目数据保证答案肯定是一个 32位 的整数
测试链接：https://leetcode.cn/problems/decode-ways/

从递归入手一维动态规划

题目4
解码方法 II
一条包含字母 A-Z 的消息通过以下的方式进行了 编码 ：
'A' -> "1"、'B' -> "2" ...'Z' -> "26"
要 解码 一条已编码的消息，所有的数字都必须分组
然后按原来的编码方案反向映射回字母，可能存在多种方式。例如"11106" 可以映射为："AAJF"、"KJF"
注意，像 (1 11 06) 这样的分组是无效的，"06"不可以映射为'F'
除了上面描述的数字字母映射方案，编码消息中可能包含 '*' 字符
可以表示从 '1' 到 '9' 的任一数字（不包括 '0'）
例如，"1*" 可以表示 "11"、"12"、"13"、"14"、"15"、"16"、"17"、"18" 或 "19"
对 "1*" 进行解码，相当于解码该字符串可以表示的任何编码消息
给你一个字符串 s ，由数字和 '*' 字符组成，返回 解码 该字符串的方法 数目
由于答案数目可能非常大，返回10^9 + 7的模
测试链接 : https://leetcode.cn/problems/decode-ways-ii/

左程云

从递归入手一维动态规划

题目5
丑数 II
丑数 就是只包含质因数 2、3 或 5 的正整数
默认第1个丑数是1，前几项丑数为:
1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20,
24, 25, 27, 30, 32, 36, 40, 45, 48, 50, 54, 60,
64, 72, 75, 80, 81, 90, 96, 100, 108, 120, 125..
给你一个整数n ，请你找出并返回第n个丑数
比如，n = 37，返回125
测试链接：https://leetcode.cn/problems/ugly-number-ii/

左程云

从递归入手一维动态规划

题目6
最长有效括号
给你一个只包含 '(' 和 ')' 的字符串
找出最长有效（格式正确且连续）括号子串的长度。
测试链接：https://leetcode.cn/problems/longest-valid-parentheses/

从递归入手一维动态规划

题目7
环绕字符串中唯一的子字符串
定义字符串 base 为一个 "abcdefghijklmnopqrstuvwxyz" 无限环绕的字符串
所以 base 看起来是这样的：
"..zabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcd.."
给你一个字符串 s ，请你统计并返回 s 中有多少 不同、非空子串 也在 base 中出现
测试链接：https://leetcode.cn/problems/unique-substrings-in-wraparound-string/

从递归入手一维动态规划

题目8
不同的子序列 II
给定一个字符串 s，计算 s 的 不同非空子序列 的个数
因为结果可能很大，所以返回答案需要对 10^9 + 7 取余
字符串的 子序列 是经由原字符串删除一些（也可能不删除）
字符但不改变剩余字符相对位置的一个新字符串
例如，"ace" 是 "abcde" 的一个子序列，但 "aec" 不是
测试链接：https://leetcode.cn/problems/distinct-subsequences-ii/