「动态规划」思考问题方向

- 理论知识
  - 用于：解决多阶段决策问题
  - 特点1：重复子问题：因为存在大量重复子问题，才需要记录之前计算的结果
  - 特点2：最优子结构：不同规模问题之间的关系
  - 特点3：无后效性：只记录阶段结果，而不关心这个结果是怎么来的，一般而言，状态定义越仔细，就能消除后效性（理论部分比较难理解，需要做题体会）
  - 思路方向2：「自底向上」递推求解：绝大多数问题都可以这样做，需要习惯这种思考模式
- 1、状态
  - 状态表示了求解问题的某个阶段
  - 先看看题目问的能不能作为状态
  - 什么状态好转移，就用什么状态，状态定义应该为转移方便而服务
- 2、状态转移方程
  - 「分类讨论」。状态转移很多时候就是在做「分类讨论」。把当前问题分成几个小问题，这些小问题的最优解构成了当前问题的最优解。尝试思考大问题怎么分类。
  - 掌握经典问题的状态设置以及状态转移方程，有的问题很有技巧，只能靠多做题，多总结
  - 经典问题有
    - 打家劫舍（198、213、337）——请留意第337题是树形dp问题
    - 第279题：完全平方数、第343题整数拆分
    - 0-1背包问题（416、474、494），完全背包问题（322、518）——必须掌握优化空间的写法，知道在一维表格下「01背包」逆向、「完全背包」正向填表的原因
    - 股票问题（121、122、123、188、309、714）——其实不难，一个约束对应一个状态维度，将状态定义具体，就方便转移了
    - （典型问题）第300题：最长上升子序列——在设计了更贴合问题场景的「状态」定义以后，才可以使用贪心算法和二分查找
    - 第1143题：最长公共子串、第72题：编辑距离、第10题：正则表达式、第44题：通配符匹配（选做）
    - 第120题：三角形最小路径和
    - （典型问题）第53题：最大子序和、第152题：乘积最大的子数组
    - （典型问题）第5题：最长回文子串——填表顺序很重要
    - 第1371题：每个元音包含偶数次的最长子字符串——状态压缩dp问题，知识点很密集、涉及异或、前缀和、哈希表
- 3、初始化
  - 初始化过程同样重要，请务必重视
  - 直接从语义出发定义初始化
  - 最小的子问题一般都比较好想，但也有例外。有些初始化状态可能不符合语义，但是可以被后来的状态所参考
  - 从状态转移方程的下标思考初始化状态，注意数组下标不能越界，或者思考是否可以通过给状态数组（矩阵）多加一行（一列），从而避免复杂的初始化讨论
  - 不同定义下初始化的值不一样，这种差别是很细微的，只能通过多做问题多总结加深体会
- 4、输出
  - 有的时候，题目要的不是最后一个状态，这一点容易被忽略
- 5、考虑是否可以优化空间
  - 「优化空间」即「表格复用」
  - 在写对之前代码的前提下，看一看状态转移的过程中，是不是有一些状态使用过了以后再也用不到，因此考虑「复用表格」已解决规模更大的问题
  - 写「优化空间」的代码在一定程度上会降低代码的可读性，不易于理解，如果看到了不太好理解的代码，很可能这一版代码是优化过的，需要从未优化的版本去理解。
  - 必须掌握的「优化空间」的问题：「0-1背包的问题」（理解以为数组逆序填表的合理性）和「完全背包」（理解一维数组顺序填表的合理性）

图例

- Priority 1
- Red
- Red
- Red