

# 动态规划中用观察优化枚举的技巧-上

前置知识:

讲解066 ~ 讲解081 - 动态规划基础

【必备】课程的动态规划大专题从讲解066开始，建议从头开始学习会比较系统

动态规划的问题中，已经写出了记忆化搜索的版本，还要写出严格位置依赖的版本，意义何在？

不仅可以进行空间压缩优化；关键还在于，很多时候通过进一步观察，可以优化枚举，让时间复杂度更好！  
这个话题讲分为讲解082、讲解083，上、下两节来讲述

注意：

优化枚举的技巧很多，讲解075-多重背包利用二进制分组、单调队列进行优化，本节课讲根据观察优化

【必备】课程中的优化手段是笔试、面试最常见的优化类型，此外动态规划的枚举行为还有如下的优化：  
单调栈、树状数组、线段树、预处理表、状态设计优化、单调性优化、斜率优化、四边形不等式 ..等等  
这些内容会在【扩展】、【挺难】课程里安排讲述

# 动态规划中用观察优化枚举的技巧-上

动态规划方法的复杂度大致可以理解为： $O(\text{状态数量} * \text{每个状态的枚举代价})$

当每个状态的枚举代价为 $O(1)$ ，那么写出记忆化搜索的版本，就是时间复杂度最好的实现了

但是当每个状态的枚举代价比较高的时候，记忆化搜索的版本可能不是最优解，可能存在进一步的优化  
之所以从记忆化搜索改出了严格位置依赖的版本，是为了建立空间感，让观察并优化枚举的分析变容易

通过观察优化枚举的技巧包括：

观察并优化转移方程（本节课、下节课）、观察并设计高效的查询结构（下节课）

本节课的题目4、题目7，是最能体现观察并优化转移方程技巧的题目

但题目4属于著名的买卖股票系列问题中的一个，所以索性把这个系列全讲了，请重点关注题目4、题目7

注意：

到目前为止讲过的题目，如果有复杂枚举行为，都无法进行优化，这是我精心设计过的题目讲述顺序  
讲过的解法都是最优解，放心！当然，斐波那契数列那个题，最优解是矩阵快速幂的解法，后续会讲述

# 动态规划中用观察优化枚举的技巧-上

## 题目1

买卖股票的最佳时机

给定一个数组 *prices*，它的第 *i* 个元素 *prices[i]* 表示一支给定股票第 *i* 天的价格

你只能选择 某一天 买入这只股票，并选择在 未来的某一个不同的日子 卖出该股票

设计一个算法来计算你能获取的最大利润

返回你可以从这笔交易中获取的最大利润

如果你不能获取任何利润，返回 0

测试链接：<https://leetcode.cn/problems/best-time-to-buy-and-sell-stock/>

# 动态规划中用观察优化枚举的技巧-上

## 题目2

买卖股票的最佳时机 II

给你一个整数数组 *prices*，其中 *prices[i]* 表示某支股票第 *i* 天的价格

在每一天，你可以决定是否购买和/或出售股票

你在任何时候 最多 只能持有 一股 股票

你也可以先购买，然后在 同一天 出售

返回 你能获得的 最大 利润

测试链接：<https://leetcode.cn/problems/best-time-to-buy-and-sell-stock-ii/>

# 动态规划中用观察优化枚举的技巧-上

## 题目3

买卖股票的最佳时机 III

给定一个数组，它的第  $i$  个元素是一支给定的股票在第  $i$  天的价格。

设计一个算法来计算你能获取的最大利润。你最多可以完成 两笔 交易

注意：你不能同时参与多笔交易，你必须在再次购买前出售掉之前的股票

测试链接：<https://leetcode.cn/problems/best-time-to-buy-and-sell-stock-iii>

# 动态规划中用观察优化枚举的技巧-上

## 题目4

买卖股票的最佳时机 IV

给你一个整数数组 *prices* 和一个整数 *k*，其中 *prices[i]* 是某支给定的股票在第 *i* 天的价格  
设计一个算法来计算你能获取的最大利润。你最多可以完成 *k* 笔交易

也就是说，你最多可以买 *k* 次，卖 *k* 次

注意：你不能同时参与多笔交易，你必须在再次购买前出售掉之前的股票

测试链接：<https://leetcode.cn/problems/best-time-to-buy-and-sell-stock-iv/>

# 动态规划中用观察优化枚举的技巧-上

## 题目5

买卖股票的最佳时机含手续费

给定一个整数数组 *prices*，其中 *prices[i]* 表示第 *i* 天的股票价格

整数 *fee* 代表了交易股票的手续费用

你可以无限次地完成交易，但是你每笔交易都需要付手续费

如果你已经购买了一个股票，在卖出它之前你就不能再继续购买股票了。

返回获得利润的最大值

注意：这里的一笔交易指买入持有并卖出股票的整个过程，每笔交易你只需要为支付一次手续费

测试链接：<https://leetcode.cn/problems/best-time-to-buy-and-sell-stock-with-transaction-fee/>



# 动态规划中用观察优化枚举的技巧-上

## 题目6

买卖股票的最佳时机含冷冻期

给定一个整数数组 $prices$ ，其中第  $prices[i]$  表示第  $i$  天的股票价格

设计一个算法计算出最大利润

在满足以下约束条件下，你可以尽可能地完成更多的交易（多次买卖一支股票）：

卖出股票后，你无法在第二天买入股票（即冷冻期为 1 天）

注意：你不能同时参与多笔交易（你必须在再次购买前出售掉之前的股票）

测试链接：<https://leetcode.cn/problems/best-time-to-buy-and-sell-stock-with-cooldown/>



# 动态规划中用观察优化枚举的技巧-上

## 题目7

DI序列的有效排列

给定一个长度为 $n$ 的字符串 $s$ ，其中 $s[i]$ 是：

"D"意味着减少，"I"意味着增加

有效排列是对有 $n+1$ 个在 $[0, n]$ 范围内的整数的一个排列 $perm$ ，使得对所有的 $i$ ：

如果  $s[i] == 'D'$ ，那么  $perm[i] > perm[i+1]$

如果  $s[i] == 'I'$ ，那么  $perm[i] < perm[i+1]$

返回有效排列的 $perm$ 的数量

因为答案可能很大，所以请返回你的答案对 $10^9+7$ 取余

测试链接：<https://leetcode.cn/problems/valid-permutations-for-di-sequence/>

这道题的尝试包含状态设计优化的内容，同时包含观察并优化转移方程，不过完全能听懂

状态设计优化在【扩展】课程阶段会有进一步的讲述