

14 面试即正义第三期：刷题指南，熟能生巧

你好，我是卢誉声。

自从给出了动态规划的解题模板后，我们就一直沿着其既定的套路在处理各式各样的动归问题。这其实印证了我们在专栏开头所说的一句话：动态规划问题简直就是模板、套路届的典范。

学到今天，其实我们已经对动态规划进行了较为全面的经验式总结，也对技术面试会重点考察的题型进行了深入分析和讲解。这些经验总结在 90% 以上的情况下都是有效的，易于理解，而且十分适合用来应对面试。

诚然，我们在整个专栏中所掌握的解题思路、技巧和最重要的解题模板十分有效而且实用。但是，一定程度的练习还是有必要的，正所谓熟能生巧。而且，动态规划问题总共就那么多，只要稍加练习，相信你就能够轻松攻破技术面试中的动归难关。

在今天的课程里，我将对动态规划题目进行了分类，从难度、类型等维度进行总结。同时，给出刷题建议。话不多说，我们开始吧。

动态规划题目总结表格见链接（请关注题目链接的同学点击此处进行查看）：

<https://shimo.im/sheets/hrHvGxvRD3xxvGD/SZhqW>

线性问题

一般来说，线性问题是动态规划中最为基础的一类动态规划问题。这是因为其主要特点就是按照我们常说的动态规划概念，即按照问题的规模从小到达进行推导，较大的子问题依赖于较小的子问题 \rightarrow 当前决策得出。因此，这是一类非常凭直觉且容易求解的动归问题。

虽然我们这么说，但是基础并不意味着容易，我在文稿中的表格里列出了常考的题目。这些题目来源于互联网，主要是 LeetCode。对于题目难度的标注，是根据我个人刷题经验总结而来的，因此可能会跟网络上标注的难度有些出入，希望这份表格能对你有所帮助。

序号	题目名	中文名	难度	题目链接	备注
1	70 Climbing Stairs	爬楼梯	简单	Link	
2	746 Min Cost Climbing Stairs	最小花费爬楼梯	简单	Link	
3	198 House Robber198	打家劫舍 I	简单	Link	这是一类求解不相邻子序列的最大和问题。相邻子序列的最大和。
4	213 Robber II	打家劫舍 II	一般	Link	
5	337 House Robber III	打家劫舍 III	困难	Link	
6	790 Domino and Tromino Tiling	多米诺和托米诺平铺	一般	Link	有条件的进行排列组合。
7	139 Word Break	分词	一般	Link	这是一道求可行性类型的动态规划问题。
8	303 Range Sum Query - Immutable	区域和检索 - 数组不可变	简单	Link	子数组问题。需要注意的是，这里显然给出了“不可变”这一关键词，如果输入数据可变或可排序，那么就不需要动态规划来解决了。具体的分析可以回看到本专栏的第5课。
9	304 Range Sum Query 2D - Immutable	二维区域和检索 - 矩阵不可变	简单	Link	
10	525 Contiguous Array	连续数组	简单	Link	
11	53 Maximum Subarray	最大子数组和	简单	Link	
12	560 Subarray Sum Equals K	和为K的子数组	一般	Link	
13	152 Maximum Product Subarray	乘积最大子数组	一般	Link	
14	238 Product of Array Except Self	除自身以外数组的乘积	一般	Link	
15	724 Find Pivot Index	寻找数组的中心索引	一般	Link	
16	1477 Find Two Non-overlapping Sub-arrays Each With Target Sum	找两个和为目标值且不重叠的子数组	一般	Link	
17	713 Subarray Product Less Than K	乘积小于 K 的子数组	一般	Link	
18	1352 Product of the Last K Numbers	最后 K 个数的乘积	一般	Link	子序列问题。一般来说，由于求解的答案不一定是连续的，因此子序列问题比子数组问题要复杂一些。
19	801 Minimum Swaps To Make Sequences Increasing	序列递增的最小交换次数	一般	Link	
20	673 Number of Longest Increasing Subsequence	最长递增子序列的个数	困难	Link	
21	300 Longest Increasing Subsequence	最长上升子序列	困难	Link	
22	1143 Longest Common Subsequence	最长公共子序列	困难	Link	
23	115 Distinct Subsequences	不同的子序列 I	困难	Link	
24	940 Distinct Subsequences II	不同的子序列 II	困难	Link	
25	1425 Constrained Subsequence Sum	带限制的子序列和	困难	Link	
26	121 Best Time to Buy and Sell Stock	买卖股票的最佳时机 I	简单	Link	
27	122 Best Time to Buy and Sell Stock II	买卖股票的最佳时机 II	简单	Link	
28	309 714 Best Time to Buy and Sell Stock with Transaction	最佳买卖股票时机含冷冻期	一般	Link	买卖股票类型的问题，建议一起练习。在本专栏的加餐中，对买卖股票作出了深入讨论，并给出了通用的解题模板，在求解前不妨先阅读一下这篇加餐。
29	714 Best Time to Buy and Sell Stock with Transaction Fee	买卖股票的最佳时机含手续费	一般	Link	
30	123 Best Time to Buy and Sell Stock III	买卖股票的最佳时机 III	困难	Link	
31	188 Best Time to Buy and Sell Stock IV	买卖股票的最佳时机 IV	困难	Link	
32	873 Length of Longest Fibonacci Subsequence	最长的斐波那契子序列的长度	简单	Link	
33	1027 Longest Arithmetic Subsequence	最长等差数列	一般	Link	
34	1055 Shortest Way to Form String	组成字符串的最短路径	一般	Link	
35	368 Largest Divisible Subset	最大的整除子集	一般	Link	
36	413 Arithmetic Slices	划分等差数列	一般	Link	
37	91 Decode Ways	解码方法 I	一般	Link	解码类型的题目，建议放在一起练习。
38	639 Decode Ways II	解码方法 II	困难	Link	
39	338 Counting Bits	比特位计数	一般	Link	
40	801 Minimum Swaps To Make Sequences Increasing	让序列递增的最小交换次数	一般	Link	
41	583 Delete Operation for Two Strings	两个字符串的删除操作	一般	Link	
42	32 Longest Valid Parentheses	最长有效括号	困难	Link	
43	132 Palindrome Partitioning II	分割回文串 II	困难	Link	
44	871 Minimum Number of Refueling Stops	最低加油次数	困难	Link	
45	818 Race Car	赛车	困难	Link	
46	120 Triangle	三角形最小路径和	一般	Link	
47	64 Minimum Path Sum	最小路径和	一般	Link	
48	221 Maximal Square	最大正方形	一般	Link	
49	931 Minimum Falling Path Sum	下降路径最小和	一般	Link	
50	343 Integer Break	整数拆分	一般	Link	
51	85 Maximal Rectangle	最大矩形	困难	Link	
52	363 Max Sum of Rectangle No Larger Than K	矩形区域不超过 K 的最大数值和	困难	Link	

在初学动态规划时，通过不断练习线性类型的动态规划问题可以不断加深理解，之后再学习其它类型的动态规划问题就会变得容易许多。

区间问题

除了线性问题以外，还有一种特别容易在技术面试环节考察的题目类型：区间类型问题。事实上，我们在专栏的课程中已经对这类问题做了深入探讨，比如“最长回文子序列”问题就属于这类区间类型的问题。

那么什么是区间呢？从“最长回文子序列”问题就可以看出，当时我们使用了状态参数 i 和 j 共同定义了字符串或数组上的一个区间，通过算法计算游走于数组之上，最后根据状态转移方程完成整个问题的推导。因此，所谓区间问题，就是使用多个状态参数来约束数据结构访问的范围，其中区间用两个端点表示。

现在，让我们来看看有哪些区间类型的题目值得练习。

序号	题目名	中文名	难度	题目链接	备注
1	5 Longest Palindromic Substring	最长回文子串	简单	Link	子数组问题
2	647 Palindromic Substrings	回文子串的个数	一般	Link	
3	1000 Minimum Cost to Merge Stones	合并石头的最低成本	一般	Link	
4	516 Longest Palindromic Subsequence	最长回文子序列	困难	Link	子序列问题
5	1147 Longest Chunked Palindrome Decomposition	段式回文	困难	Link	
6	730 Count Different Palindromic Subsequences	计算不同回文子字符串的个数	困难	Link	
7	1312 Minimum Insertion Steps to Make a String Palindrome	让字符串成为回文串的最少插入次数	困难	Link	
8	312 Burst Balloons	戳气球	困难	Link	
9	546 Remove Boxes	移除盒子	困难	Link	
10	1039 Minimum Score Triangulation of Polygon	多边形三角剖分的最得分	困难	Link	

到这里你可能会有疑问，那就是单个字符串或数组的问题不也存在区间吗？比如在最长上升子序列问题中，我们明明就可以用一个 $DP[i]$ 来存储计算的子问题答案。其实，这里面是有显著区别的。

对于单个字符串或数组的问题来说，它其实隐含了区间的起始位置，因为每个子问题的起始位置都是 0，因此我们通过降维实现了只需要一个状态参数的计算。在这种情况下，我们可以考虑它是一个线性问题。

但对于区间类型问题来说，一般我们会定义 $DP[i][j]$ ，表示考虑 $[i..j]$ 范围内的元素，原问题的解增加 i 、减小 j 都可以得到更小规模的子问题，状态转移是按照区间长度由短到长进行推导的。比如“最长回文子序列问题”，其原问题的最终答案可能不是存储在以 0 为起始位置的子序列当中的，正因为我们需要明确的计算出该问题的起始位置，因此状态参数 i 不能被忽略。在这种情况下，我们需要将其考虑成区间类型的动态规划问题。

由于状态参数的增多，导致了状态存储，即备忘录的维度的增加，势必会提高写出算法代码的难度。通常来说，在填充高维备忘录时，你都需要小心**计算方向**这个东西。

背包问题

在专栏中，我曾耗费了不少篇幅深入讲解了背包问题，如果你对其有所遗忘，不妨回看第6课和第7课的内容。简单地说，背包问题是一种组合优化的 NP 完全问题。简单的背包问题包括硬币找零，而稍微复杂一些的就包括0-1背包问题、完全背包问题和多重背包问题。

几乎所有的背包问题都可以概括成这样一句话：给定一系列物品，每种物品都有自己的重量和价值两个参数。此时，给定一个有重量上限的背包，求如何选择才能使得物品的总价值最高。

只不过，在技术面试环节，即便考察了背包问题，往往考察的也是其变种。我们曾在第6课就提出过一个粉碎石头的问题，那就是一个背包问题的变种。

现在，我给出你值得关注和练习的背包问题。

序号	题目名	中文名	难度	题目链接	备注
1	322 Coin Change	零钱兑换	简单	Link	我们在专栏开篇就讲解了类似的问题。
2	518 Coin Change 2	零钱兑换	简单	Link	
3	1049 Last Stone Weight II	最后一块石头的重量 II	一般	Link	粉碎石头的问题。
4	474 Ones and Zeroes	一和零	一般474 Ones and Zeroes	Link	
5	494 Target Sum	目标和目标和	一般	Link	
6	377 Combination Sum IV	组合总和 IV	一般	Link	
7	416 Partition Equal Subset Sum	分割等和子集	一般	Link	
8	879 Profitable Schemes	盈利计划	困难	Link	

对于0-1背包、完全背包和多重背包的问题在 LintCode 上有所体现。我的建议是根据专栏的课程多加练习，如果你能在遇到这类问题时轻松写出状态转移方程，就更好了。

方案总数问题

现在，让我们来看看什么是方案数类型的问题。其实最容易想得到的就是我们曾在第10课中讲解的问题。其中“简单的路径规划”和“带障碍的路径规划”就属于方案数的问题，其原问题要求解的答案就可以是有多少种可行路径。

求方案总数的问题和求可行性（True或False）的问题是可以相互转化的。举个例子，在早期讨论的硬币找零问题，当时我们要求的是：最少需要几枚硬币凑出总金额。

但那个问题显然存在无法用硬币凑出解的情况（至于这种情况，原问题要求返回 -1）。因此，如果我们把原问题修改成：你能否用最少的硬币凑出总金额？这样就变成了一个求可行性问题了。

现在，我给出你值得关注和练习的问题。

序号	题目名	中文名	难度	题目链接	备注
1	62 Unique Paths	不同路径	简单	Link	
2	63 Unique Paths II	不同路径 II	一般	Link	
3	96 Unique Binary Search Trees	不同的二叉搜索树	一般	Link	
4	95 Unique Binary Search Trees II	不同的二叉搜索树 II	一般	Link	
5	1155 Number of Dice Rolls With Target Sum	掷骰子的 N 种方法	一般	Link	
6	940 Distinct Subsequences II	不同的子序列 II		Link	

复杂问题

最后，我列出了常考的较为复杂的动态规划面试问题，这些问题确实比较困难，但却常常出现在技术面试环节中。因此，你应该花一些时间仔细研读这几道题目，就足以应对复杂的动归技术面试了。

序号	题目名	中文名	难度	题目链接	备注
1	887 Super Egg Drop	扔鸡蛋之殇	困难	Link	这是一道非常漂亮的递归加迭代求解的动态规划问题，值得深入分析和理解。另外，如果改成仍玻璃珠，你还会吗？
2	1067 Digit Count in Range	一定范围内的数字计数	困难	无	
3	600 Non-negative Integers without Consecutive Ones	不含连续1的非负整数	困难	Link	
4	1012 Numbers With Repeated Digits	至少有 1 位重复的数字	困难	Link	

课程总结

在本课中，我列出了在技术面试环节常考的高频动态规划问题。你可以根据题目的类型，难易程度，逐步推进你的“刷题”进程。就像我在开头说的那样，虽然动态规划解题模板非常管用，但一定的练习还是有必要的。

这里列出的题目确实比较多，但我也不希望你去把每道题都解一遍。最重要的还是要掌握以及运用我们从专栏开课就给出的动态规划解题模板。学习并培养解题思路，养成思考的习惯，这才是本课最重要的目的。同时，你也可以将这节课列出的题目当作一个参考文档，它几乎就是你能看到的最全面的动态规划面试问题总结了。

我相信你能够通过这些练习，进一步加深对动态规划的理解，彻底攻破最难技术面试问题这一关！

课后思考

你是否遇到过本课中尚未列出的动态规划技术面试问题？如果有，请列出它，不妨与大家一起分享，共同进步。

期待你的留言。如果今天的内容对你有所帮助，也欢迎把文章分享给你身边的朋友，邀请他一起练习！

[上一页](#)

[下一页](#)