

【干货】动态规划十问十答

2016-07-23 令狐老师 九章算法

问1 动态规划是个什么鸟蛋？

答：动态规划是一种通过“大而化小”的思路解决问题的算法。区别于一些固定形式的算法，如二分法，宽度优先搜索法，动态规划没有实际的步骤来规定第一步做什么第二步做什么。所以更加确切的说，动态规划是一种解决问题的思想。这种思想的本质是，一个规模比较大的问题（假如用2-3个参数可以表示），是通过规模比较小的若干问题的结果来得到的（通过取最大，取最小，或者加起来之类的运算）所以我们经常看到的动态规划的核心——状态转移方程都长成这样：

* $f[i][j] = f[i-1][j] + f[i][j-1]$

* $f[i] = \max\{f[j] \text{ if } j < i \text{ and } \dots\} + 1$

* $f[i][j] = f[0][j-1] \&\& \text{judge}(1,i) \parallel f[1][j-1] \&\& \text{judge}(2,i) \parallel \dots$

问2 动态规划面试考得多么？

答：多。并且越来越多。随着CS从业与求职者的增加，并伴随大家都是“有备而来”的情况下，一般简单的反转链表之类的题目已经无法再在面试中坚挺了。因此在求职者人数与招聘名额的比例较大的情况下，公司会倾向于出更难的问题。而动态规划就是一种比较具有难度，又比较“好出”的面试问题。相比其他的算法与数据结构知识来说，贪心法分治法太难出题了，搜索算法往往需要耗费求职者过长的程序编写时间一般也不倾向于出，二叉树链表等问题题目并没有那么多，而且求职者也都会着重准备这一块。因此动态规划这一类的问题，便越来越多的出现在了面试中。

问3 动态规划快在哪儿？

答：动态规划一般来说是“高效”的代名词，因为其解决的问题一般退而求其次的算法只有搜索了。以“数字三角形”一题为例（<http://www.lintcode.com/problem/triangle/>），在“三角矩阵”中找一条从上到下的路径，使得权值之和最小。如果使用暴力搜索的算法，那么需求穷举出 $2^{(n-1)}$ 条路径（ n 为三角形高度），而使用动态规划的话，则时间复杂度降低到了 n^2 ，完成了质的飞跃。那么究竟为什么这么快呢？原因在于动态规划算法去掉了“无用和重复的运算”。在搜索算法中，假如从A->B有2条路径，一条代价为10，另外一条代价为100，B->终点有1024条路径。当我们选择了代价为10的那条路径走到B时，可以继续往下走完1024条路径到终点，但是在此之后，我们再从代价为100的路径从A走到B时，我们可以发现此时无论如何走，都不可能有刚才从10的路径走过来更好，所以这些计算是“无用”的计算，也可以说是“重复”的计算。这就是动态规划之所以“快”的重要原因。

问4 学习动态规划有什么捷径？

答：我们将动态规划的常见类型分为如下几种：

- * 矩阵型
- * 序列型
- * 双序列型
- * 划分型
- * 区间型
- * 背包型
- * 状态压缩型
- * 树型

其中，在技术面试中经常出现的是矩阵型，序列型和双序列型。划分型，区间型和背包型偶尔出现。状态压缩和树型基本不会出现（一般在算法竞赛中才会出现）。

每种类型都有着自己的题目特点和状态的表示方法。以矩阵型动态规划为例，一般题目会给你一个矩阵，告诉你有一个小人在上面走动，每次只能向右和向下走，然后问你比如有多少种方案从左上走到右下 (<http://www.lintcode.com/problem/unique-paths/>)。这种类型状态表示的特点一般是使用坐标作为状态，如 $f[i][j]$ 表示走到 (i,j) 这个位置的时候，一共有多少种方案。状态的转移则是考虑是从哪儿走到 (i,j) 这个坐标的。而序列型的动态规划，一般是告诉你一个序列；双序列的动态规划一般是告诉你两个字符串或者两个序列。将所做过的动态规划问题按照这些类别进行归类，分析状态的表示方法和状态转移方程的构造方法在每种类型中的近似之处，会让你更快的学会动态规划。

问5 什么样的问题适合使用动态规划？

答：可以使用动态规划的问题一般都有一些特点可以遵循。如题目的问法一般是三种方式：

1. 求最大值/最小值
2. 求可不可行
3. 求方案总数

如果你碰到一个问题，是问你这三个问题之一的，那么有90%的概率是使用动态规划来求解。要重点说明的是，如果一个问题让你求出“所有的”方案和结果，则肯定不是使用动态规划。

问6 解决一个动态规划问题的步骤是什么？

答：首先根据“问5”判断是否是动态规划的问题，如果是，则尝试将其按照“问4”进行分类，找到对应的类别和相似的问题。接着从下面的4个要素去逐步剖析解决这道题：

1. 状态是什么
2. 状态转移方程是什么
3. 状态的初始值是什么
4. 问题要求的最后答案是什么

每个步骤分析完成之后，就基本上解决了整道动态规划的问题。

问7 怎样优化动态规划的时间？

答：一般来说，使用动态规划求解的问题，时间上已经比暴力搜索要优化很多了。但是仍然存在着一些可以优化的空间。通常来说，动态规划的时间优化，有如下两种常见的方式：

1. 通过变换状态优化
2. 通过决策单调优化

对于通过变换状态来优化的问题比较难，需要一些经验和灵感。而对于决策单调的优化，则比较简单，但适用范围不广，一般只适用于划分型动态规划当中，通常这个方法可以将复杂度降低一个数量级。

问8 怎样优化动态规划的空间？

答：动态规划的空间优化只有一种方法，就是使用滚动数组进行优化。以一个二维的动态规划为例子。假如状态转移方程如下： $f[i][j] = f[i-1][j] + f[i][j-1]$ 。我们可以发现，第*i*层的状态，已经和第*i-2*层的状态没有关系了，那么这种情况下，用于存储第*i-2*层的空间就可以被重复利用。方法非常简单，把数组的第一维对2取模就可以了： $f[i \% 2][j] = f[(i-1) \% 2][j] + f[i \% 2][j-1]$ 。这种方法通常可以将空间复杂度降低一个数量级。

问9 有什么书籍和参考资料可以推荐么？

著名的背包九讲：

<http://vdisk.weibo.com/s/tanGyi7qM8TVE>

(也可以直接在网上搜索背包九讲)

问10 有哪些动态规划题目必须要练习的？

在LintCode上包含了30余道动态规划的练习题，都是从实际的面试问题中汇总的精选练习：

<http://www.lintcode.com/tag/dynamic-programming/>

九章算法班第22期正在火热报名中！

想要更加扎实的学习动态规划算法么？

本文作者主讲的九章算法班火热报名中！

北京时间7.31周日早9:30-11:30(美西时间7.18周六晚6:30-),

第一节课免费试听！

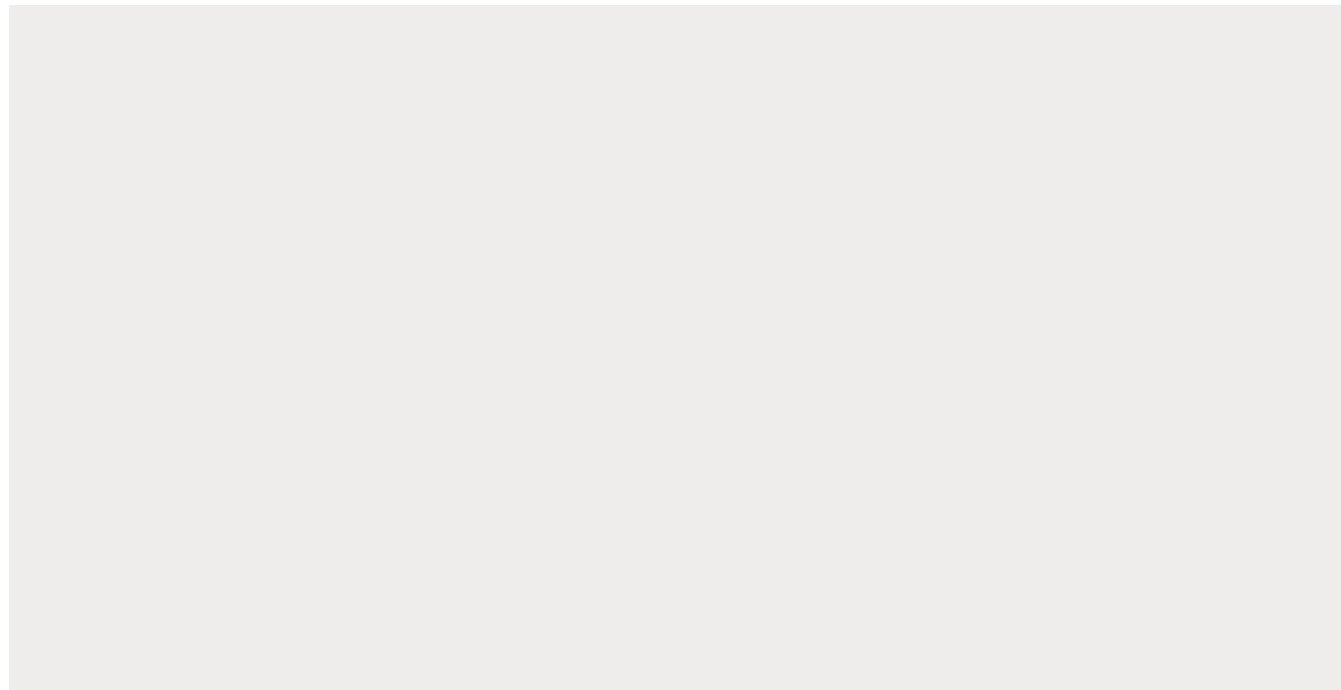
报名网址：<http://t.cn/RtLi3o6>，或猛戳“阅读原文”。

版权声明

本文作者：令狐老师，九章算法班主讲人。

九章算法，提供算法培训、IT求职咨询，帮助更多中国人找到好工作。

本文属九章算法版权所有，转载请注明以上声明。



[Read more](#)