

动态规划一直是 ACM 竞赛中的重点，同时又是难点，因为该算法时间效率高，代码量少，多元性强，主要考察思维能力、建模抽象能力、灵活度。

本人动态规划博客地址：

http://blog.csdn.net/cc_again/article/category/1261899

动态规划（英语：Dynamic programming，DP）是一种在数学、计算机科学和经济学中使用的，通过把原问题分解为相对简单的子问题的方式求解复杂问题的方法。动态规划常常适用于有**重叠子问题**和**最优子结构**性质的问题，动态规划方法所耗时间往往远少于朴素解法。

动态规划背后的基本思想非常简单。大致上，若要解一个给定问题，我们需要解其不同部分（即子问题），再合并子问题的解以得出原问题的解。通常许多子问题非常相似，为此动态规划法试图仅仅解决每个子问题一次，从而减少计算量：一旦某个给定子问题的解已经算出，则将其**记忆化**存储，以便下次需要同一个子问题解之时直接查表。这种做法在重复子问题的数目关于输入的规模呈**指数增长**时特别有用。

动态规划问题满足三大重要性质

最优子结构性质：如果问题的最优解所包含的子问题的解也是最优的，我们就称该问题具有最优子结构性质（即满足最优化原理）。最优子结构性质为动态规划算法解决问题提供了重要线索。

子问题重叠性质：子问题重叠性质是指在用递归算法自顶向下对问题进行求解时，每次产生的子问题并不总是新问题，有些子问题会被重复计算多次。动态规划算法正是利用了这种子问题的重叠性质，对每一个子问题只计算一次，然后将其计算结果保存在一个表格中，当再次需要计算已经计算过的子问题时，只是在表格中简单地查看一下结果，从而获得较高的效率。

无后效性：将各阶段按照一定的次序排列好之后，对于某个给定的阶段状态，它以前各阶段的状态无法直接影响它未来的决策，而只能通过当前的这个状态。换句话说，每个状态都是过去历史的一个完整总结。这就是无后向性，又称为无后效性。

动态规划分类有很多划分方法，网上有很多是按照状态来分，分为一维、二维、区间、树形等等。我觉得还是按功能即解决问题的类型以及难易程度来分比较好，下面按照我自己的理解和归纳，把动态规划的分类如下：

一、简单基础 dp

这类 dp 主要是一些状态比较容易表示，转移方程比较好想，问题比较基本常见的。主要包括递推、背包、LIS（最长递增序列），LCS（最长公共子序列），下面针对这几种类型，推荐一下比较好的学习资料和题目。

1、递推：

递推一般形式比较单一，从前往后，分类枚举就行。

简单：

[hdu 2084 数塔](#) 简单从上往下递推

[hdu 2018 母牛的故事](#) 简单递推计数

[hdu 2044 一只小蜜蜂...](#) 简单递推计数（Fibonacci）

[hdu 2041 超级楼梯](#) Fibonacci

[hdu 2050 折线分割平面](#) 找递推公式

推荐：

[CF 429B B. Working out](#) 四个角递推

[z oj 3747 Attack on Titans](#) 带限制条件的计数递推 dp

[uva 10328 Coin Toss](#) 同上题

[hdu 4747 Mex](#)

[hdu 4489 The King's Ups and Downs](#)

[hdu 4054 Number String](#)

2、背包

经典的背包九讲：<http://love-oriented.com/pack/>

推荐博客：<http://blog.csdn.net/woshi250hua/article/details/7636866>

主要有 0-1 背包、完全背包、分组背包、多重背包。

简单：

[hdu 2955 Robberies](#) 01 背包

[hdu 1864 最大报销额](#) 01 背包

[hdu 2602 Bone Collector](#) 01 背包

[hdu 2844 Coins](#) 多重背包

[hdu 2159 FATE](#) 完全背包

推荐：

[woj 1537 A Stone-I](#) 转化成背包

[woj 1538 B Stone-II](#) 转化成背包

[poj 1170 Shopping Offers](#) 状压+背包

[zoj 3769 Diablo III](#) 带限制条件的背包

[zoj 3638 Fruit Ninja](#) 背包的转化成组合数学

[hdu 3092 Least common multiple](#) 转化成完全背包问题

[poj 1015 Jury Compromise](#) 扩大区间+输出路径

[poj 1112 Team Them UP](#) 图论+背包

3、LIS

最长递增子序列，朴素的是 $O(n^2)$ 算法，二分下可以写成 $O(n \lg n)$ ：维护一个当前最优的递增序列——找到恰好大于它更新

简单：

[hdu 1003 Max Sum](#)

[hdu 1087 Super Jumping!](#)

推荐：

[uva 10635 Prince and Princess](#) LCS 转化成 LIS

[hdu 4352 XHXJ's LIS](#) 数位 dp+LIS 思想

[srm div2 1000](#) 状态压缩+LIS

[poj 1239 Increasing Sequence](#) 两次 dp

4、LCS

最长公共子序列，通常 $O(n^2)$ 的算法

[hdu 1503 Advanced Fruits](#)

[hdu 1159 Common Subsequence](#)

[uva 111 History Grading](#) 要先排个序

[poj 1080 Human Gene Functions](#)

二、区间 dp

推荐博客: <http://blog.csdn.net/woshi250hua/article/details/7969225>

区间 dp, 一般是枚举区间, 把区间分成左右两部分, 然后求出左右区间再合并。

[poj 1141 Brackets Sequence](#) 括号匹配并输出方案

[hdu 4745 Two Rabbits](#) 转化成求回文串

[zoj 3541 The Last Puzzle](#) 贪心+区间 dp

[poj 2955 Brackets](#)

[hdu 4283 You Are the One](#) 常见写法

[hdu 2476 String Printer](#)

[zoj 3537 Cake](#)

[CF 149D Coloring Brackets](#)

[zoj 3469 Food Delivery](#)

三、树形 dp

比较好的博客: <http://blog.csdn.net/woshi250hua/article/details/7644959>

一篇论文: <http://doc.baidu.com/view/f3b19d0b79563c1ec5da710e.html>

树形 dp 是建立在树这种数据结构上的 dp, 一般状态比较好想, 通过 dfs 维护从根到叶子或从叶子到根的状态转移。

[hdu 4123 Bob's Race](#) 二分+树形 dp+单调队列

[hdu 4514](#) 求树的直径

[poj 1655 Balancing Act](#)

[hdu 4714 Tree2Cycle](#) 思维

[hdu 4616 Game](#)

[hdu 4126 Genghis Khan the Conqueror](#) MST+树形 dp 比较经典

[hdu 4756 Install Air Conditioning](#) MST+树形 dp 同上

[hdu 3660 Alice and Bob's Trip](#) 有点像对抗搜索

[CF 337D Book of Evil](#) 树直径的思想 思维

[hdu 2196 Computer](#) 搜两遍

四、数位 dp

推荐一篇论文: <http://wenku.baidu.com/view/d2414ffe04a1b0717fd5dda8.html>

数位 dp, 主要用来解决统计满足某类特殊关系或有某些特点的区间内的数的个数, 它是按位来进行计数统计的, 可以保存子状态, 速度较快。数位 dp 做多了后, 套路基本上都差不多, 关键把要保存的状态给抽象出来, 保存下来。

[hdu 2089 不要 62](#) 简单数位 dp

[hdu 3709 Balanced Number](#) 比较简单

[CF 401D Roman and Numbers](#) 状压+数位 dp

[hdu 4398 X mod f\(x\)](#) 把模数加进状态里面

[hdu 4734 F\(x\)](#) 简单数位 dp

[hdu 3693 Math teacher's homework](#) 思维变换的数位 dp

[hdu 4352 XHXJ's LIS](#) 数位 dp+LIS 思想

[CF 55D Beautiful Numbers](#) 比较巧妙的数位 dp

[hdu 3565 Bi-peak Numbers](#) 比较难想

[CF 258B Little Elephant and Elections](#) 数位 dp+组合数学+逆元

五、概率(期望) dp

推荐博客: <http://www.cnblogs.com/kuangbin/archive/2012/10/02/2710606.html>

推荐博客: <http://blog.csdn.net/woshi250hua/article/details/7912049>

推荐论文:

[《走进概率的世界》](#)

[《浅析竞赛中一类数学期望问题的解决方法》](#)

[《有关概率和期望问题的研究》](#)

一般来说概率正着推, 期望逆着推。有环的一般要用到高斯消元解方程。期望可以分解成多个子期望的加权和, 权为子期望发生的概率, 即 $E(aA+bB+\dots) = aE(A) + bE(B) + \dots$

[ural 1776 Anniversiry Firework](#) 比较基础

[hdu 4418 Time travel](#) 比较经典 BFS+概率 dp+高斯消元

[hdu 4586 Play the Dice](#) 推公式比较水

[hdu 4487 Maximum Random Walk](#)

[jobdu 1546 迷宫问题](#) 高斯消元+概率 dp+BFS 预处理

[hdu 3853 L00PS](#) 简单概率 dp

[hdu 4405 Aeroplane chess](#) 简单概率 dp, 比较直接

[hdu 4089 Activation](#) 比较经典

[poj 2096 Collecting Bugs](#) 题目比较难读懂

[zoj 3640 Help me Escape](#) 从后往前, 比较简单

[hdu 4034 Maze](#) 经典好题, 借助树的概率 dp

[hdu 4336 Card Collector](#) 状态压缩+概率 dp

[hdu 4326 Game](#) 这个题状态有点难抽象

六、状态压缩 dp

这类问题有 **TSP**、**插头 dp** 等。

推荐论文: <http://wenku.baidu.com/view/ce445e4f767f5acfalc7cd51.html>

推荐博客: http://blog.csdn.net/sf_____/article/details/15026397

推荐博客: http://www.notonlysuccess.com/index.php/plugin_dp/

[hdu 1693 Eat the Trees](#) 插头 dp

[hdu 4568 Hunter](#) 最短路+TSP

[hdu 4539](#) 插头 dp

[hdu 4529 状压 dp](#)

[poj 1185 炮兵阵地](#)

[poj 2411 Mandriann's Dream](#) 轮廓线 dp

[hdu 3811 Permutation](#)

[poj 1038](#)

[poj 2441](#)

[hdu 2167](#)

[hdu 4026](#)

[hdu 4281](#)

七、数据结构优化的 dp

有时尽管状态找好了，转移方程的想好了，但时间复杂度比较大，需要用数据结构进行优化。常见的优化有二进制优化、单调队列优化、斜率优化、四边形不等式优化等。

1、二进制优化

主要是优化背包问题，背包九讲里面有介绍，比较简单，这里只附上几道题目。

[hdu 1059 Diving](#)

[hdu 1171 Big Event in Hdu](#)

[poj 1048 Follow My Magic](#)

2、单调队列优化

推荐论文：<http://wenku.baidu.com/view/4d23b4d128ea81c758f578ae.html>

推荐博客：<http://www.cnblogs.com/neverforget/archive/2011/10/13/11.html>

[hdu 3401 Trade](#)

[poj 3245 Sequece Partitioning](#) 二分+单调队列优化

3、斜率优化

推荐论文：[用单调性优化动态规划](#)

推荐博客：<http://www.cnblogs.com/ronaflx/archive/2011/02/05/1949278.html>

[hdu 3507 Print Article](#)

[poj 1260 Pearls](#)

[hdu 2829 Lawrence](#)

[hdu 2993 Max Average Problem](#)

4、四边形不等式优化

推荐博客: <http://www.cnblogs.com/ronaflx/archive/2011/03/30/1999764.html>

推荐博客: <http://www.cnblogs.com/zxndgv/archive/2011/08/02/2125242.html>

[hdu 2952 Counting Sheep](#)

[poj 1160 Post Office](#)

[hdu 3480 Division](#)

[hdu 3516 Tree Construction](#)

[hdu 2829 Lawrence](#)