动态规划基础入门

动态规划是ICPC中使用非常频繁的一类编程技巧。所谓动态规划（DP）也就是以状态为基础的多段决策，有对单个状态规划到最终求得全局最优解。动态规划有三个必备条件，那就是：状态、状态转移、状态的无后效性。本质上，动态规划是搜索的一种人工优化，能用动态规划解决的问题自然也能通过搜索进行暴力解决，不过当一些问题的状态转移的层次不是非常明显的时候，记忆化搜索也是非常好的一种取代方法。

## 基本理论

刚才说了要使用动态规划，那必须直到动态规划的三个特征：状态、状态转移、状态的无后效性。首先，确定状态非常重要的，一个好的状态是即完备又不冗余的，例如：在求TSP路径时，就不要记录以前路径的走法，只要记录下以前走过的节点和当前节点即可。另外状态需要有无后效性，也就是说当前状态值一旦确定不能再被更改，例如：再含环的图中，后面节点的状态就可能影响前面状态的值，所以不能直接套用动态规划。状态转移是计算状态的过程，通过逐步更新新状态的值，直到求出最终的解。

在确定状态和状态转移以后，针对不同的情况还有不同的优化方法，1）如果状态太多，可以考察能不能进行状态压缩；2）如果转移量太大，考察能不能用一些技巧（平行四边形），进行优化。

根据动态规划的结构，可将动态规划分为一维，二维，以及树形DP等。

## 经典问题

在DP技巧的使用中，最经典的问题就是多段图决策问题。给一张有向无环图，问从起点走到终点的最短距离。如下图所示，dp[i]记录起点到点i的最短路径值，状态转移方程如下：

dp[i] = min{dp[j] + distance(i, j)}, i, j之间有路径

初学者坑还会遇到很多经典的DP问题：多段图决策问题，最长公共子序列，最长回文字符串，最长递增子序列，0/1背包问题，找零问题，最大子段问题。

题目列举：POJ 1163 the Triangle，POJ 1579 Function Run Fun，POJ 2081 Recaman's Sequence，

POJ 1953 World Cup Noise， POJ 1458 Common Subsequence， POJ 2250 Compromise，POJ 1159 Palindrome，

POJ 1080 Humman Gene Function， POJ 2192 Zipper，POJ 3356 AGTC，POJ 1887 Testing the CATCHER，

POJ 2533 Longest Ordered Subsequence， POJ 1631 Bridging signals， POJ 1157 LITTLE SHOP OF FLOWERS，

POJ 1050 To the Max， POJ 1014 Dividing， POJ 1160 post office， POJ 1125 Stockbroker Grapevine，

POJ 1179 Polygon