

动态规划总结

by Amber

1. 按状态类型分

写在前面：

从状态类型分，并不表示一题只从属于一类。其实一类只是一种状态的表示方法。可以好几种方法组合成一个状态，来解决问题。

1.1. 编号（长度）动态规划

共性总结

本类的状态是基础的基础，大部分的动态规划都要用到它，成为一个维。

一般来说，有两种编号的状态：

状态(i)表示前 i 个元素决策组成的一个状态。

状态(i)表示用到了第 i 个元素，和其他在 1 到 i-1 间的元素，决策组成有的一个状态。

题库

a) 最长不下降子序列

以元组(i)作为状态，表示第 i 个作为序列的最后一个点的时候的最长序列。于是很容易想到 $O(n^2)$ 得算法。但本题可 **合理组织状态**，引入一个单调的辅助数组，利用单调性二分查找，优化到 $O(n\log n)$ 。关于优化详见优化章。

一些问题可将数据 **有序化**，转化成本题。

应用：

拦截导弹(NOIP99 Advance 1) 就是原题。

Beautiful People (sgu199)，要将数据有序化：其中一个权作为第一关键字不下降排列，另一个权作为第二关键字不上升。

Segment (ural 1078)，将线段的左端点有序化就可以了。

b) LCS

状态(i,j)，表示第 1 个字符串的第 i 位，与第 2 个字符串的第 j 位匹配，得到的最长的串。若有多个串要 LCS，则加维，即几个串就几个维。我也将此题归入 **路径问题**。

c) 花店橱窗布置(IOI99)

见 **路径问题**。

1.2. 区间动态规划

共性总结

本类问题与下一章的 **划分问题** 的 **决策的分割点无序** 交集比较大（占本类问题的 30%）。

题库

a) 石子合并

见 **划分问题**

b) 模版匹配(CEOI01,Patten)

这题特殊的地方是状态的值是一个集合而不是一个数。

c) 不可分解的编码(ACM World Final 2002)

d) Electric Path(ural1143)

e) 邮局(IOI2000 Day2 1)

若状态表示的思路从第 i 个村庄可以从属于哪个邮局，无最优子结构。转变一个方向：第 k 个邮局可以“控制”一个区间的村庄 $[i,j]$ 。于是方程就显然了：

$$f(k,i,j)=\min\{f(k-1,p,i-1)+w(i,j)\}(k-1\leq p\leq i-1)$$

$S(i)$ 为村庄 i 到原点的距离。

$$w(i,j)=\min\{k|\sum\{|S(k)-S(p)|\}(i\leq p\leq j)\}(i\leq k\leq j)$$
 找到 $[i,j]$ 间最好的一个邮局点。

不过可以发现 $\sum\{|S(k)-S(p)|\}$ 是单调的，所以取中位数就可以了。即上式中 k 的取值范围只有 $\text{floor}((i+j)/2)$, $\text{ceil}((i+j)/2)$ 两个。Floor 是下取整。Ceil 是上取整。这样每次转移时间降到 $O(1)$ 。

注意到是区间连续的，即 $(p,i-1)$ 和 (i,j) 中的 $i-1,i$ 是连续的，所以空间可以降维： $f(i,j)$ 表示放前 i 个邮局到前 j 个村庄的最优值。

$$f(i,j)=\min\{f(i-1,p-1)+w(p,j)\}(i-1\leq p\leq j-1)$$

$e(i,j)$ 为当 $f(i,j)$ 到达最优值时的 p 。

通过证明四边形不等式，得到 $e(i,j)\leq e(i,j+1)\leq e(i+1,j+1)$

决策数量又少了一个数量级。

1.3. 坐标动态规划

共性总结

之后的一些问题，状态是由坐标维与其他的维组成。本类与划分问题(是 2 维或多维的坐标系的划分)与路径问题的交集占本类问题中大多数。

题库

a) 棋盘分割(NOI99 4)

主要是将公式变形，变形后的公式很容易看出方程。

状态是由 2 个坐标组成的 4 元组 $(x1,y1)(x2,y2)$ ，表示一个子棋盘。这有点像之前的区间动态规划，只不过是 1 维转 2 维。

后见路径问题。

1.4. 数轴动态规划

共性总结

题库

a) 01 背包

应用：

装箱问题 (NOIP01 Trade 4)

就是原题。

值币分割

可利用方程的性质，空间降 1 维。

币值可重复的币值分割(pku1742, Problem F LouTianCheng's Contest in POJ)

使用左右法在定位上加速。

另给状态加一个属性 last,记录上一次剩下的可用的同币值硬币数(利用了当前转移是唯一前驱的特点)。

b) 取火柴问题(sgu153 Playing with matches)

c) Stone Pile(ural1005 Stone Pile)

d) 公路巡逻(CTSC2000)

1.5.5.树型动态规划

共性总结

1) 动态规划的顺序

一般按照后序遍历的顺序,即处理完儿子再处理当前节点,才符合树的子结构的性质。

2) 多叉树转换为二叉树

由于要分配附加维到各个节点,而分配附加维是个划分问题,若还是按当前节点到各个儿子节点分配,则成了一个**整数划分**问题, $O(n^2)$ 。所以要把多叉树转换为二叉树,这样才能按动态规划的方式只决策当前点的分配问题, $O(n)$ 。

3) 加当前点的选或不选的常数维

加此维解决的是后效性问题。

.....

4) 在将边信息转成树时的技巧

将读入的边分裂成 2 条边,将这 2 条边关联起来(就是找到一条边,另一条边的编号就知道)。用**前向星表示法**表示边(按起点有序),以后用边的时候,用了一条边打不可用标志,也将关联边打不可用标志。这样可以保证 $O(n)$ 的时间完成信息处理,而且在父节点找儿子的过程中带来很大的方便。

5) 复杂度

树型动态规划复杂度基本上是 $O(n)$;若有附加维 m ,则是 $O(nm)$ 。

题库

a) 选课(CTSC97-3)

由于要分配课程数,所以要多叉树转换为二叉树。

b) 贪吃的九头龙(NOI02-3)

若小头数大于 1 的话,则让不同的小头吃一段树枝的 2 个端点。

这样就把问题转化成:附加维是大头吃的个数,当前点由不由大头吃的常数维的动态规划。由于涉及划分问题,所以要多叉树转换为二叉树。

c) 求树的质心(sgu134 Centroid)

给出一棵边不带权的树,求点,使得去掉此点后,剩下的最大的连通子图的顶点数最小。

d) 求树中的点最远距离最近。

给出一棵边带权的树,求树中的点,使得此点到树中的其他结点的最远距离最近。

Computer Network (sgu149)

Computer Net (ural1056)

1.6. 集合动态规划（状态压缩）

共性总结

1) 数据特殊性

给出的数据在某一个或几个维度上一般具有比较小的范围(可以枚举一类状态)。一个枚举的状态是一个集合。

2) 编码

由于集合中元素个数的不定性或范围大，直接开数组存，不好索引数组(编程复杂度太高)，所以要将集合编码。

利用数据的可枚举性，将枚举的状态(集合)编码。一般来说码值的范围要很小(尽量排除无用的码值，如炮兵：当前格和上格存在炮兵的情况是非法的，可以排除)。

规定编码的码值代表的意思，要尽量规定好维护的码值。(如炮兵：当前格存在炮兵的用 2，上格存在炮兵用 1。这样下一层的规划时，只要码值-1 即可)。

有时候可以直接利用编码的顺序动态规划，因为这时编码已经是**拓补有序**。如 TSP 问题当前已选点集合的状态的前驱的编码的值一定比当前的编码的值小。

3) 状态压缩

对有限阶段的放置情况，行走情况编码(其实质也是放置的集合或行走路线的集合)，这样的编码，也有人谓之：“状态压缩”。此类题以“炮兵阵地”为典型，进行扩展。

题库

a) 购物 (IOI95-2)

可将每种物品按 5 进制编码。(5 为每种物品数的上限)

由于物品数的上限为 5，比较小，也可直接开数组存。

b) Roger 游戏任务一 (CTSC98 Day2 4)

一个正方体在一个方格内的状态只有 24 种，而且可以通过顶面和前面来表示，这样用 3 维的状态(x,y,p)就可以解决，p 为 1 到 24 种状态中的一种。

c) TSP 问题

观察一下 TSP 的搜索过程：for (x in 未选点) TSP(x)

即当前路的最后一个节点为 x,现在要选择下一个节点 y,而 y 要在未选点的集合中。若未选点或已选点的集合已确定，则后效性消除。可以 DP。状态为令 X 为当前路的已选点的集合(含 i)，当前路的最后一个节点为 i。2 元组(X,i)为经过已选点的集合 X 到节点 i 的最短长度。将 X 编码即可。

注意：**并没有因为动态规划将问题从 NP 类带到 P 类。**

应用: [DNA Laboratory](#)(Problem B,TU-Darmstadt Programming Contest 2004)

将每个串的交迭部分求出，就可以将问题专成 TSP

但要输出字典序最小的，则需要注意 DP 顺序。

有具体的报告。

d) 炮兵阵地

十分经典，详见**报告**。

应用:

[Another Chocolate Maniac](#)(sgu132) 类似炮兵的做法的最值，只不过是求最小值，麻烦点。

[Hardwood floor](#)(sgu131) 类似炮兵的做法的统计

[Little Knights](#)(sgu225) 类似炮兵的做法的统计,数据量太大要 const

[Little Kings](#)(sgu223) 类似炮兵的做法的统计

[Bugs 公司](#)(CEOI 2002) 类似炮兵的做法的最值

1.7. 利用动态规划思想求最值，编号（循环变量）的迭代

共性总结

要利用上次的一些运算“剩下”的循环变量作当前循环的边界，主要在于找出一种决策顺序，使之成立。

题库

a) [奶牛浴场](#)

b) [Communication System](#)

将数据有序化，从大到小枚举带宽，每次可利用上次处理的结果 Min，来决策当前状态。称作迭代，或就是一种动态规划。

(zju1409, Problem C Tehran 2002 Iran Nationwide Internet Programming Contest)

1.8. 记忆化搜索

题库

a) [Magic Trick](#) (Problem G, TU-Darmstadt Programming Contest 2004)

2. 按转移方式分

2.1. 存在性

递推

1) [01 统计](#)(CTSC99 1)

2) 卡特兰数

[circle](#)(sgu130)

3) 鹰蛋

2.2. 求一系列的分割（合并）点（划分问题）

2.2.1. 决策的分割点有序

共性总结

a) [有序性](#)

每次决策的点的编号是有序的，即要按决策的顺序输出分割点的编号的话，编号是

有序的，满足分割点的编号按升序排列。

b) 方程一般形式

$$f(n,m)=\text{optimize}\{f(k,m-1)+w(k+1,n)\}$$

(n,m) 表示从 1 到 n 个点中划分为 m 个部分的最优值； k 为决策的分割点，即第 m 个部分为 $k+1$ 到 n ；这里 optimize 可以为 \max, \min 。

题库

a) 整数划分

常应用在将一个权分配给一定的小分割块，如：将大堆的石子分成一定的小堆，小堆可为空，大堆要分完。有时应用在树型动态规划（二叉转多叉）中。

b) 乘积最大(NOIP00 Advance 2)

就是按上面的一般式的方程做。

2.2.2. 决策的分割点无序

共性总结

a) 无序性

每次决策的点的编号是无序的，即要按决策的递归顺序输出分割点的编号的话，编号是无序的。

b) 方程一般形式

$$f(i,j)=\text{optimize}\{f(i,k-1)+f(k+1,j)\}+w(i,j)$$

(i,j) 表示从 i 到 j 的范围内选取一个分割点 k 的最优值，子问题是分割点左边 $(i,k-1)$ 和右边 $(k+1,j)$ 的点的范围的最优值；这里 optimize 可以为 \max, \min 。

方程很类似 2 叉树的性质。

c) 四边形不等式

此类的问题，有些可用四边形不等式优化。见优化章。

题库

a) 石子合并(NOI95 2)

经典，详见报告。

可用四边形不等式优化成 $O(n^2)$

其实还可以用类似堆的数据结构在 $O(n\log n)$ 的时间内完成，但这就不是动态规划了。

应用：

构造最优二叉排序树(CTSC96 2)

b) 多边形(IOI98)

这题值的正负号处理要注意，乘法运算，由于符号的加入，使原本的的最优解，一下变成负的。

c) 加分二叉树(NOIP03 Advance 3)

方程就是一般式，转移的函数： $w(i,j)=\text{sum}(i,k-1)*\text{sum}(k+1,j)+d(k)$ 。由于 $w(i,j)$ 不满足凸单调性，所以不能用四边形不等式优化。

d) 括号序列(Problem B, NEERC 2001)

这题的分割点不是一个元素，而是元素间的一条线。

主要的思维方式是从递归定义。

2.3. 路径问题

共性总结

a) 行走方向决定阶段性

有规定源点与终点。每次行走方向都有一定的规定，使原点 to 终点的所有路径形成无环有向图。

b) 多源或多汇

当多源或多汇时，应该加维，使得每个源，都有一个路径的状态与之对应。如有 n 个源的网格类问题，常常转态是 $(x_1, y_1)(x_2, y_2) \dots (x_n, y_n)$ 。但是源太多的话，空间上不允许，可以降低问题转成网络流问题。

c) 双向动态规划

由于有规定源点与终点，可以双向动态规划，但要考虑效果好不好，理论上是比原来少 $1/2$ ，但有时由于可用于决策的状态较少，效果就不错了。

d) 决策稀疏性

就是所谓走法，若对于一个状态，它的前驱或者后继数很少(从无环有向图角度，就是入度或出度少)，称决策稀疏。

e) 状态稀疏性

就是很多状态是没有用的，如排列的 LCS，状态为 2 维的 (x, y) ，但对于一个 x 只有一个 y 是有效个。所以实质上状态数还是线形的。

本类一些技巧性的东西较多，在题库中具体说明。

题库

a) 方格取数(NOIP00 advance 4)

$(x_1, y_1)(x_2, y_2)$

对角线空间优化

b) 花店橱窗布置(IOI99)

我对本题有个小改造：若花瓶无序，如何做，有序指：对于花束 $i <$ 花束 j ，花束 i 对应的花瓶编号 $<$ 花束 j 对应的花瓶编号。那么这样就是一个 NP 问题了，可用后面的基于状态压缩的动态规划解决。

3. 动态规划的优化

3.1. 迭代

3.2. 四边形

3.3. 凸性的优化

最主要的未总结，给出相关的题与已有的报告（自己或他人的）