动态规划算法与数学

目录

1 引言

在算法竞赛中，动态规划（Dynamic Programming, DP）是一种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。计数等非最优化问题的递推解法也常被称作 DP。[1]

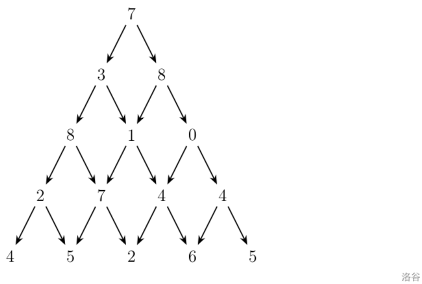
例 1. 【*IOI 1994*】*Number Triangles*[2]

观察下面的数字金字塔。

写一个程序来查找从最高点到底部任意处结束的路径，使路径经过数字的和最大。每一步可以走到左下方的点也可以到达右下方的点。

[1] https://oi-wiki.org/dp/

[2] https://www.luogu.com.cn/problem/P1216



在上面的样例中，从 $7 \to 3 \to 8 \to 7 \to 5$ 的路径产生了最大权值。

在算法竞赛中有计算次数的限制（一般数量级不超过 10^9），所以可以通过 DP 进行优化。如上题用 $a(i,j)$ 表示 i 行 j 列的数，设 $f(i,j)$ 表示走到 i 行 j 列是最大的和，那么有 $f(i,j) \leftarrow \max\{f(i-1,j-1), f(i-1,j)\} + a(i,j)$，其中不合法位置的 f 值定义为负无穷。则答案为 $\max\limits_{j=1}^{r}\{f(r,j)\}$。

我们可以使用类似思路进行数列递推相关的一些求解。

下面给出动态规划原理及其在数学方面的对应理解。

最优子结构　一个问题的最优解包含其子问题的最优解，称为具有最优子结构。一般不涉及最优化的数学题目中，可以理解为每个子问题的求解方式已知，且涉及的参数已知。

无后效性　已经求解的子问题，不会再受到后续决策的影响。

子问题重叠　如果有大量的重叠子问题，我们可以用空间将这些子问题的解存储下来，避免重复求解相同的子问题，从而提升效率。在数列递推中，可以理解为数列每一项的递推方式相同，以便于后续求解。

下面给出题目对上边的内容详细解释。

2　例题

2.1　经典数学题目

例 2.【*Fibonacci* 数列】有一个 15 级的楼梯，每一次可以上 1 或 2 级。求从底部走到最高一级的方案数。

解

设 f_i 表示走到第 i 级楼梯的方案数。

初始值：$f_0 = 1$，不合法位置 f 值为 0。

转移方程：$f_i = f_{i-1} + f_{i-2}$。

在计算中，得出 $f_0 = 1, f_1 = 1$ 之后，所有涉及的位置全部合法，然后模拟递推，或求数列通项的方式求出 f_{15} 即可。

答案 987

注 如果这道题问求从底部走到第 n 级楼梯的方案数，就必须算出通项公式。所以求通项还是很重要的。

例 3. 【经典题目】有甲，乙两人打球。如果前一局甲赢了，那么甲获胜概率为 p，否则为 q。求第 n 局甲获胜的概率。（第一局甲赢的概率为 p。）

解

设 f_i 表示第 i 局甲赢的概率，g_i 表示甲输的概率。

转移方程：$f_i = pf_{i-1} + qg_{i-1}$，其中 $g_n + f_n = 1$，则有：$f_i = (p-q)f_{i-1} + q$。

初始值：$f_0 = 1$。因为第一局甲赢的概率为 p，可以理解为第零局甲赢了。

求出通项即可。

答案 $f_n = \frac{(p-1)(p-q)^n - q}{p-q-1}$

这两道题目中，转移方程在计算 f_n 时，其他数据已知，且对于每个角标的方程相同，类似最优子结构和子问题重叠。这个方程也明显能看出具有无后效性。在这种情况下，使用数列相关知识求解即可。

例 4. 【经典题目】从网格左上角 $(1,1)$ 到右下角 (m,n)，每次只能向右或向下走，求路径总数。

解

设 $f(i,j)$ 表示走到 (i,j) 的方案数，显然有 $f(i,j) = f(i-1,j) + f(i,j-1)$，不合法位置方案数视为 0，答案为 $f(n,m)$。

引申

网格不为矩形，有部分缺失。

解：正常计算，缺失位置记作 0。

这道题可以看出，有时候可能原本一个题可以直接用组合数算（原题中 C_{n+m-2}^{n-1}），但是经过改编后不便使用组合数计算（实际上，如果 n,m 较大而缺失位置较少，使用组合数更好，但计算量仍会极大），就可以使用这样的递推暴力求解。

2.2 DP 问题选做

例 5. 【*Codeforces 148D*】*Bag of mice*[1]

龙和公主正在为除夕夜做什么争论不休。龙建议飞到山上去看仙女在月光下跳舞，而公主则认为他们应该早点睡觉。他们急于达成一个友好的协议，于是决定听天由命。

他们轮流从一个袋子里抽出一只老鼠，袋子里最初有 w 只白色老鼠和 b 只黑色老鼠。最先抽到白老鼠的人获胜。龙每抽出一只老鼠，袋子里的其他老鼠就会惊慌失措，其中一只老鼠就会从袋子里跳出来(公主抽老鼠时要小心，不要吓到其他老鼠)。公主先抽。公主获胜的概率是多少？

如果袋子里没有老鼠了，也没有人抽到白老鼠，龙就赢了。自己跳出袋子的小老鼠不被视为被抽中(不能确定获胜者)。老鼠一旦离开袋子，就永远不会再回来。每只老鼠从袋子中被抽出的概率与其他老鼠相同，每只老鼠跳出袋子的概率与其他老鼠相同。

解

显然这个问题需要有两个角标，因为有黑白两个参数。

设 $f(w,b)$ 表示还有 w 只白色老鼠和 b 只黑色老鼠时获胜的概率，转移如下：

1. 没有白色老鼠，即 $w=0$：一定失败。

$$f(w,b)=0$$

2. 有不超过一个黑色老鼠，即 $b\le 1$：必须马上抽到白色老鼠才行。

$$f(w,b)=\frac{w}{w+b}$$

[1] https://codeforces.com/problemset/problem/148/D

3. 那么此时可以抽到白色的概率：$f(w,b)=\dfrac{w}{w+b}$，接着考虑下一次：

（*1*）$b\geq 2$，可以两人一起抽两只黑色，跑一只白色：

$$f(w,b)\leftarrow f(w,b)+f(w-1,b-2)*\frac{b}{b+w}*\frac{b-1}{b-1+w}*\frac{w}{b-2+w}$$

（*2*）$b\geq 3$，可以两人一起抽两只黑色，跑一只黑色：

$$f(w,b)\leftarrow f(w,b)+f(w,b-3)*\frac{b}{b+w}*\frac{b-1}{b-1+w}*\frac{b-2}{b-2+w}$$

其中 *1*，*2*，*3* 独立，*3* 的（*1*）（*2*）可同时进行。

所以手动列个表（如果是数学题，w,b 不会太大），列个表算一下就有答案了。

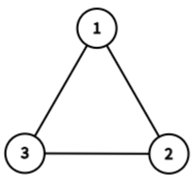
答案生成 *https://github.com/Heldivis/Code/blob/main/CF148D.cpp*

注 上述转移 *1*，*2*，*3* 存在重复部分，但是这样计算较为方便，易懂。注意，在计算中，分类讨论可以有重叠部分，但答案不能重复计算。

这道题分类讨论难度和计算量较难，对转移方程（基本式是得分式）和计算（不能乱七八糟恶心人）的要求较高。

例 6. 【*HNOI2013*】游走[1] 给定一个 n 个点 m 条边的无向连通图，顶点从 1 编号到 n，边从 1 编号到 m。

小 Z 在该图上进行随机游走，初始时小 Z 在 1 号顶点，每一步小 Z 以相等的概率随机选择当前顶点的某条边，沿着这条边走到下一个顶点，获得等于这条边的编号的分数。当小 Z 到达 n 号顶点时游走结束，总分为所有获得的分数之和。现在，请你对这 m 条边进行编号，使得小 Z 获得的总分的期望值最小。样例如下：（答案是 $\dfrac{10}{3}$）



[1]https://www.luogu.com.cn/problem/P3232

前置知识

1. 排序不等式（Rearrangement Inequality）

设 $x_1 \le x_2 \le \cdots \le x_n$ 和 $y_1 \le y_2 \le \cdots \le y_n$ 是两组实数，σ 是任意一个长度为 n 的排列，那么排序不等式表明：

$$\sum_{i=1}^{n} x_i y_{n-i+1} \le \sum_{i=1}^{n} x_{\sigma(i)} y_i \le \sum_{i=1}^{n} x_i y_i$$

即逆序和不大于乱序和，乱序和不大于顺序和。证明很简单，不再赘述。

2. 图论的一些概念

与一个顶点 v 关联的边的条数称作该顶点的度 *(degree)*，记作 $d(v)$。

3. 高斯消元（Gaussian Elimination）

是线性代数规划中的一个算法，可用来为线性方程组求解。本题需要手算解个 n 元一次方程组。

解

设 f_i 表示第 i 个点期望经过次数，则对 $1 \le i < n$ 有（因为到 n 后不能走出来，记 $f_n = 0$）：

$$f_i = [i = 1] + \sum_{j\ 和\ i\ 相连}^{j \ne n} \frac{f_j}{d(j)}$$

那么对于每个顶点，都能列出来这样的方程。n 个 n 元方程，可解。

那么第 i 条连接 u, v 的边的期望经过次数 $e_i = \dfrac{f_u}{d(u)} + \dfrac{f_v}{d(v)}$。

按照 e 从小到大排序，给期望小的边更大的编号，由排序不等式知这样一定有最小值。

答案生成 *https://github.com/Heldivis/Code/blob/main/LgP3232.cpp*

这类问题被称为“有后效性 DP”[1]。这类游走问题在高考数学试题中也常见。

[1] https://oi-wiki.org/dp/probability/

可见，在算法竞赛中的一类 DP 题目可以经过改编，成为高考数学的题目。其他类型 DP 和 DP 优化[1]存在计算量大，必须使用计算机（数据结构），前置知识多等问题，出成数学题目的可能较小。

但是，这里给出一种优化方式（加速递推），可以便于计算。**注意，一般数学题不会有这样的计算量，如果发现递推计算很难，建议换思路。**

例 7. 【*洛谷P1962*】斐波那契数列（改编）[2]

请你求出斐波那契数列 $F_n \bmod 10$ 的值。[34] 。

矩阵

一个 $m \times n$ 的矩阵 A 是一个由 m 行 n 列元素排列成的矩形阵列。

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

两个矩阵相乘的定义为一个 $n \times t$ 的矩阵 A 和一个 $t \times m$ 的矩阵 B 相乘，得到的结果为一个 $n \times m$ 的矩阵 C，$C_{i,j} = \sum_{k=1}^{t}(A_{i,k} \times B_{k,j})$。如

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 1\times7+2\times9+3\times11 & 1\times8+2\times10+3\times12 \\ 4\times7+5\times9+6\times11 & 4\times8+5\times10+6\times12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \\ 139 & 154 \end{bmatrix}$$

类似的，我们也可以定义出矩阵加法和矩阵减法（相应位置相加减）。

两个矩阵若行列数不满足上述条件，则无法进行运算。矩阵乘法遵循结合律和分配率，但不遵循交换律。

一个矩阵的乘方定义为多个该矩阵的乘积矩阵，根据定义，只有一个行列数相等的矩阵才能进行乘方运算。

在本题中，根据矩阵定义，有：$\begin{bmatrix} F_{n-1} & F_{n-2} \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} F_n & F_{n-1} \end{bmatrix}$

[1]https://oi-wiki.org/dp/

[2]https://www.luogu.com.cn/problem/P1962

[3]$a \div b = c \cdots d$，记作 $a \bmod b = d$

[4]原模数为 $10^9 + 7$, 一个大质数

于是 F_n 为矩阵 $\left(\begin{bmatrix}1 & 1\end{bmatrix}\times\left(\begin{bmatrix}1 & 1\\1 & 0\end{bmatrix}\right)^{n-2}\right)$ 的第一个元素。

也许有人会问欸你这个不是更复杂了吗，但是对于转移矩阵的次方，如 $A^{16}=(((A^2)^2)^2)^2$，加上每次只需要保留个位数，这样就简单一点了（结合律）。

答案生成 *https://github.com/Heldivis/Code/blob/main/LgP1962.cpp*

再次提醒：使用矩阵乘法加速递推大概率不会是正解的方法，并且只适用于要求只保留个位（或者对 5,7）等较小数取模，或者是选择题能取模看结果的提。这样大概率不会是正解的方法。

2.3 改编 DP 问题的一些数学问题

毕竟原本 DP 题目考察算法，有时候和数学题目的风格不一样。这里给出一些 DP 的题目，并改编为地道的数学题。

由于数学考试不能使用数据结构优化，没学过图论，状压没用，人有直觉等等原因，一些经典的 DP 套路不会出现，所以较为灵活。

例 8. 【*AtCoder Educational DP Contest A*】*Frog 1*[1]

N 个石头，编号为 $1,2,\dots,N$。对于每个 $i(1\le i\le N)$，石头 i 的高度为 h_i。最初有一只青蛙在石头 1 上。他将重复几次以下操作以到达石头 N：如果青蛙当前在石头 i 上，则跳到石头 $i+1$ 或石头 $i+2$。需要 $|h_i-h_j|$ 的费用，其中 j 是要落到上面的石头。求青蛙到达石头 N 之前需要的最小总费用。

解

$f_i=\min\{f_{i-1}+|h_i-h_{i-1}|,f_{i-2}+|h_i-h_{i-2}|\}$，不合法位置记作正无穷。

这道题很简单，可能有人说欸我画个图直接出来了，再看一个。

例 9. 【*AtCoder ACL Beginner Contest D*】*Flat Subsequence*[2]

[1] https://atcoder.jp/contests/dp/tasks/dp_a

[2] https://atcoder.jp/contests/abl/tasks/abl_d

数学表述 已知 $\{a_n\}=\{1,1,4,5,1,5,1,9,1,9,8,10,1,5,4,3,8,6,9,7,2,4\}$，且满足 $|a_{b_i}-a_{b_{i-1}}|\le 4$（$i\ge 2$），$\{b_n\}$ 单调递增。则 $\{b_n\}$ 最多有几个元素？

答案 16

解

显然这道题瞪眼瞪不出来（也可能是我太菜了 PwP）。

考虑使用 f_i 表示上一次选的数字是 i 的最大长度，**依次**令 $i=a_1,a_2,\dots,a_n$，那么有转移 $f_i\leftarrow \max\limits_{j=i-k}^{i+k} f_j$，答案为 $\max f_i$。

注 这道题要求 $\{b_n\}$ 单调递增，所以要**依次**令 $i=a_1,a_2,\dots,a_n$。

这道题在数据较少的时候就不是很好看出来（至少比最长(不)上升/下降子序列难），所以可以使用这样的方法进行计算。

例 10.【*AtCoder Educational DP Contest J*】*Sushi*[1]

数学表述 已知 $\{a_n\}=\{1,2,1,2,2,1,1,1,2,2\}$，每次随机选择 $i\in[1,10]\cap\mathbb{N}$，令 $a_i\leftarrow a_i-1$。求第一次 $\forall i\in[1,10]\cap\mathbb{N},a_i\le 0$ 时的期望操作次数（保留整数）。

答案 40

解

显然答案与数列顺序无关，只和每个数字出现的次数有关。设 $f(i,j)$ 表示还有 i 个 1，j 个 2 的操作期望次数。则转移如下：

$$f(i,j)=\underbrace{\frac{n}{i+j}}_{\text{选中一个正数}}+\underbrace{\frac{i}{i+j}\cdot f(i-1,j)}_{\text{选到 }1}+\underbrace{\frac{j}{i+j}\cdot f(i+1,j-1)}_{\text{选到 }2}$$

我感觉这样解决数学题最大的麻烦就是计算问题。建议列表，模拟计算。

3 思考题

这里给出几个简单的思考题，不一定有数学表达式通解，还可能涉及一些简单的优化，但可以看看，拓展思路。

[1] https://atcoder.jp/contests/dp/tasks/dp_j

随机数生成器（https://www.luogu.com.cn/problem/P3600）
【HAOI 2015】数组游戏 (https://www.luogu.com.cn/problem/P3179)
【ZJOI 2013】话旧（https://www.luogu.com.cn/problem/P3336）
【THUPC 2023 决赛】阴阳阵（https://www.luogu.com.cn/problem/P9385）

4 结语

众所周知，N 年 OI 一场空。但是这里面有一些思想是可以用到数学的，特别是简单的一些算法（数据结构：?）。

然后上面给的代码也许能喂给 AI 让 AI 帮忙跑？不知道没试过。

在 https://heldivis.github.io/数学/动态规划算法与数学 提供 PDF 下载。