计数类 DP 与数位 DP

李淳风

长郡中学

2024 年 9 月 26 日

前言

对于计数类问题和数位统计问题这两类问题，都特别强盗“不重不漏”，统计对象的可能情况一般也比较多，通常需要精确的划分和整体性的计算。因此，使用动态规划抽象出问题中的“子结构”和推导的“阶段”，将有助于我们准确而高效地进行求解。

例题

Gerald and Giant Chess

给定一个 $H * W$ 的棋盘，棋盘上只有 N 个格子是黑色的，其它格子都是白色的。
在棋盘左上角有一个卒，每一步可以向右或向下移动一格，并且不能移动到黑色格子中。求这个卒从左上角移动到右下角，一共有多少种可能的路线。
$1 \leq H, W \leq 10^5, 1 \leq N \leq 2000$。输出对 $10^9 + 7$ 取模后的结果即可。

例题

Gerald and Giant Chess

给定一个 $H*W$ 的棋盘，棋盘上只有 N 个格子是黑色的，其它格子都是白色的。
在棋盘左上角有一个卒，每一步可以向右或向下移动一格，并且不能移动到黑色格子中。求这个卒从左上角移动到右下角，一共有多少种可能的路线。
$1 \le H, W \le 10^5, 1 \le N \le 2000$。输出对 10^9+7 取模后的结果即可。

棋盘上的白色格子数量巨大，而黑色格子最多只有 2000 个。我们应该考虑如何依靠黑色格子进行计数。如果不考虑黑色格子的限制，那么从左上角走到右下角的方案数就是 C_{H+W-2}^{H-1}。如果我们再求出经过了至少一个黑色格子的路线数量，二者相减就得到了只经过白色格子的方案数。

Gerald and Giant Chess

把所有黑色格子按照行、列坐标递增的顺序排序。特别地，我们假设左上角是第 0 个黑色格子，右下角是第 $N+1$ 个黑色格子。设第 i 个黑色格子在第 x_i 行 y_i 列，那么我们可以设 $f[i]$ 表示从左上角走到第 i 个黑色格子，并且途中不经过其他黑色格子的路线数：

$$f[i] = C_{x_i+y_i-2}^{x_i-1} - \sum_{j=1}^{i-1} f[j] * C_{x_i-x_j+y_i-y_j}^{x_i-x_j} \quad (x_j \le x_i, y_j \le y_i)$$

这个式子的含义是，枚举 j 作为从左上角到 (x_i, y_i) 经过的第一个黑色格子，也就是从左上角到 (x_j, y_j) 不能经过其它黑色格子，路线数为 $f[j]$。从 (x_j, y_j) 到 x_i, y_i 再随便走，方案数是一个组合数。求和符号的部分求出了从左上角到 (x_i, y_i)，途中经过至少一个黑色格子的路线数，用总数剪掉，就得到了 $f[i]$。

我们为什么这样设计状态转移方程？因为要想避免重复计数，最好的方法就是找到一个“基准点”，围绕这个“基准点”构造一个不可划分的整体，以避免子问题之间的重叠。在本题中，如果第一个经过的黑色格子不同，那么对应的路线肯定不会重复。而 j 枚举了 i 之前的所有黑色格子，所以也不会遗漏。最终 $f[N+1]$ 就是本题的答案。

例题

Connected Graph

求 N 个节点的无向连通图有多少个，节点有标号，编号为 $1 \sim N$，$1 \le N \le 50$。注意，节点有标号意味着节点之间是不同的。

例题

Connected Graph

求 N 个节点的无向连通图有多少个，节点有标号，编号为 $1 \sim N$, $1 \leq N \leq 50$。注意，节点有标号意味着节点之间是不同的。

在计数类动态规划中，我们通常考虑把一个问题划分成若干个子问题，以便进行递推。一个连通图不容易划分，而一个不连通的无向图则很容易划分成节点更少的两部分。所以我们可以先求出 N 个节点的无向图总数，再减去 N 个点的不连通无向图的数量。

Connected Graph

N 个点的无向图总数显然是 $2^{N*(N-1)/2}$，含义是每条边都可以有或者没有。接下来考虑不连通的情况。一张不连通的无向图必定由若干个连通块构成。根据之前提到过的“基准点”思想，我们可以枚举标号为 1 的节点所在的连通块包含的节点个数 k，从 $2 \sim N$ 这 $N-1$ 个节点中选出 $k-1$ 个与 1 号节点一起构成大小为 k 的块。显然，我们有 C_{N-1}^{k-1} 种选法。剩余 $N-k$ 个节点构成任意无向图。
综上所述，设 $f[i]$ 表示 i 个节点的无向连通图个数，状态转移方程为：

$$f[i] = 2^{i*(i-1)/2} - \sum_{j=1}^{i-1} f[j] * C_{i-1}^{j-1} * 2^{(i-j)*(i-j-1)/2}$$

初值为 $f[1] = 1$，目标为 $f[N]$。

例题

How Many of Them?

在无向连通图中，若一条边被删除后，图会分成不连通的两部分，则称该边为割边。求满足下列条件的无向连通图的数量：

- 由 N 个节点构成，节点有标号，编号为 $1 \sim N$;
- 割边恰好有 M 条；
- 没有自环和重边。

$2 \leq N \leq 50, 0 \leq M \leq N*(N-1)/2$。答案对 10^9+7 取模。

例题

How Many of Them?

在无向连通图中，若一条边被删除后，图会分成不连通的两部分，则称该边为割边。求满足下列条件的无向连通图的数量：

- 由 N 个节点构成，节点有标号，编号为 $1 \sim N$;
- 割边恰好有 M 条；
- 没有自环和重边。

$2 \le N \le 50, 0 \le M \le N * (N-1)/2$。答案对 $10^9 + 7$ 取模。

在无向图中，不含割边的极大连通子图称为双连通分量。若把每个双连通分量看作一个节点，则所有双连通分量和割边构成一个树形结构。

How Many of Them?

设 $f[i][j]$ 表示 i 个点构成的，包含 j 条割边的无向连通图数量。先考虑 $j>0$ 的情况。

为了避免重复，我们以编号为 1 的点为“基准点”，枚举 1 号点所在的双连通分量的大小 k。从 $2\sim i$ 这 $i-1$ 个点中选出 $k-1$ 个，与 1 号点共同构成双连通分量，方案数为 $f[k][0]*C_{i-1}^{k-1}$。

接下来考虑图中其他部分。如果去掉 1 号点所在的双连通分量，那么无向图会被分成若干个连通块，每个连通块都有一条连接到 1 号节点所在双连通分量的边。

我们再设 $g[i][j][k]$ 表示由 i 个点组成的、包含 j 条割边的、有 k 个连通块的无向图数量。回顾 $f[i][j]$ 的计算，去掉 1 号点所在的双连通分量后，剩余部分的方案数就是 $g[i-k][j-x][x]$，其中 x 是一个正整数，表示剩余部分为 x 个连通块。因为我们还要把这 x 个连通块连接到 1 号点所在的双连通分量上，所以连接方案还有 k^x 种：

$$f[i][j]=\sum_{k=1}^{i-1}(f[k][0]*C_{i-1}^{k-1}*\sum_{x=1}^{\min(i-k,j)}g[i-k][j-x][x]*k^x)$$

How Many of Them?

特别地，设 $h[i]$ 表示 i 个节点的无向连通图数量，在上一题中我们已经掌握了求法。于是当 $j=0$ 时，有：

$$f[i][0] = h[i] - \sum_{j=1}^{i-1} f[i][j]$$

最后，我们来考虑 g 的计算方法。同样以编号最小的节点所在的连通块为基准，枚举这个连通块的大小 p，和割边数量 q。当然，需要从 $i-1$ 个节点中再选出 $p-1$ 个节点构成这个连通块，所以该连通块的方案就有 $f[p][q] * C_{i-1}^{p-1}$ 种。另外，我们需要从该连通块中选出一个节点，用于与 1 号点所在的双连通块相连，显然有 p 种选法。除此之外，剩余部分的方案数就是 $g[i-p][j-q][k-1]$。于是我们得到方程：

$$g[i][j][k] = \sum_{p=1}^{i} \sum_{q=0}^{k} f[p][q] * C_{i-1}^{p-1} * p * g[i-p][j-q][k-1]$$

注意这里的 $g[i][j][k]$ 与之前的定义稍有差别，若是单纯计算“i 个点构成的、包含 j 条割边的、有 k 个连通块的无向图数量”，则不需要乘 p。整个算法的时间复杂度为 $O(n^5)$。

例题

A Decorative Fence

有 N 块长方形的木板，长度分别为 $1, 2, \cdots, N$，宽度都是 1. 现在要用这 N 块木板组成一个宽度为 N 的栅栏，满足在栅栏中，每块木板两侧的木板要么都比它高，要么都比低。

显然，有很多种构建栅栏的方案，每个方案可以写成一个长度为 N 的序列，序列中的各元素就是木板的长度。把这些序列按照字典序从小到大排序，给定整数 C，求排名为 C 的栅栏中，各木板的长度从左到右依次是多少。

$1 \le N \le 20, 0 < C < 2^{63}$。

例题

A Decorative Fence

有 N 块长方形的木板，长度分别为 $1, 2, \cdots, N$，宽度都是 1. 现在要用这 N 块木板组成一个宽度为 N 的栅栏，满足在栅栏中，每块木板两侧的木板要么都比它高，要么都比低。
显然，有很多种构建栅栏的方案，每个方案可以写成一个长度为 N 的序列，序列中的各元素就是木板的长度。把这些序列按照字典序从小到大排序，给定整数 C，求排名为 C 的栅栏中，各木板的长度从左到右依次是多少。
$1 \le N \le 20, 0 < C < 2^{63}$。

类似于倍增优化 DP 中的“拼凑”思想，我们可以用试填法来确定排名为 C 的栅栏。具体来说，我们从小到大枚举第一块木板的长度，当第一块木板长度为 h 时，设后面 $N-1$ 块木板构成栅栏的方案数为 T，若 $T \ge C$，则说明第一块木板的长度应该为 h，可以继续确定第二块木板的长度。否则，令 C 减去 T，h 增加 1，从夫上述判断。为了让这一方法更加高效，我们需要预处理 T 的值。

A Decorative Fence

设 $f[i][j][0/1]$ 表示用 i 块长度不同的木板构成栅栏，其中最左边的木板的长度从小到大排在第 j 位，最左边的木板处于低位/高位的方案总数：

$$f[i][j][0] = \sum_{p=j}^{i-1} f[i-1][p][1]$$

$$f[i][j][1] = \sum_{p=1}^{j-1} f[i-1][p][0]$$

注意到 “用长度为 $1 \sim i$ 的木板构成栅栏，最左边的长度为 i” 与 “用 i 块长度不同的木板构成栅栏，最左边的长度排名为 i” 两个问题是等价的，所以我们后者来设计动态规划会更好进行状态转移、数量统计。
计算完 f 之后就可以进行试填法了。记录已经确定的木板长度，以及已经确定的最靠右第 $i-1$ 块木板处于低位还是高位即可。注意第一块木板可能处于低位，也可能处于高位。
动态规划的时间复杂度为 $O(N^3)$，试填法的时间复杂度为 $O(N^2)$。

数位统计 DP

数位统计 DP 是与数字相关的一类计数问题。在这类问题中，一般给定一些限制条件，求满足限制条件的第 K 小的数是多少，或者求区间 $[L, R]$ 内有多少个满足限制条件的数。解决方法一般都可以用到“试填法”，先通过动态规划进行预处理，再基于“拼凑思想”按位求出最终的答案。

例题

启示录

古代人认为 666 是属于魔鬼的数。不但如此，只要某数字的十进制表示中有三个连续的 6，古代人也认为这个是魔鬼的数。现在，给定一个 X，请你求出第 X 小的魔鬼数。
$x \leq 5 * 10^7$，每个测试点包含不超过 1000 组数据。

例题

启示录

古代人认为 666 是属于魔鬼的数。不但如此，只要某数字的十进制表示中有三个连续的 6，古代人也认为这个是魔鬼的数。现在，给定一个 X，请你求出第 X 小的魔鬼数。
$x \le 5 * 10^7$，每个测试点包含不超过 1000 组数据。

设 $f[i][3]$ 表示由 i 位数字构成的魔鬼数有多少个，$f[i][j](0 \le j \le 2)$ 表示由 i 位数字组成的、开头已经有连续 j 个 6 的非魔鬼数有多少个。
注意在计算 f 时，我们允许前导 0 的存在。
考虑第 i 位（最高位）是什么数字，容易得到转移方程：

$$f[i][0] = 9 * (f[i-1][0] + f[i-1][1] + f[i-1][2])$$

$$f[i][1] = f[i-1][0], f[i][2] = f[i-1][1]$$

$$f[i][3] = f[i-1][2] + 10 * f[i-1][3]$$

动态规划完成后，我们先通过 $f[i][3]$ 来确定第 X 小的魔鬼数的位数。
然后按照“试填法”的思想，从左到右依次考虑每个数位，同时记录当前末尾已经有连续的几个 6 即可。

例题

月之迷

如果一个十进制数能够被它的各位数字之和整除，则称这个数位为“月之数”。给定整数 L 和 R，求闭区间 $[L, R]$ 内有多少个月之数。
$1 \leq L, R < 2^{31}$，每个测试点不超过 3000 组数据。

例题

月之迷

如果一个十进制数能够被它的各位数字之和整除，则称这个数位为“月之数”。给定整数 L 和 R，求闭区间 $[L, R]$ 内有多少个月之数。
$1 \leq L, R < 2^{31}$，每个测试点不超过 3000 组数据。

首先，区间 $[L, R]$ 内的个数可以拆成两个区间 $[1, R], [1, L-1]$ 中的个数相减，所以我们只需要考虑 $[1, R]$ 这样的区间形式。
我们还是采取试填法的思想，从高位到低位给每一位填数。由于我们只需要在某一位上填了一个比上限小的数，那么后边的每一位上的数就都可以任意填了。只有在每一位上都与 R 相同，才需要继续向后扫描。

月之迷

因此我们可以预处理出 $f[i][j][k][l]$ 表示由 i 位数字构成、各位数字之和为 j、对 k 取模余数是 l 的数由多少个，当我们可以任意填数的时候就可以用 f 数组来快速计算方案数了。f 的转移也不复杂：

$$f[i][j][k][l] = \sum_{p=0}^{9} f[i-1][j-p][k][(l - p * 10^{i-1}) \bmod k]$$

当然，由于各位数字之和是不确定的，并且取值范围不大，我们可以先枚举各位数字之和 sum，然后从左到右扫描每个数位。设当前正在处理第 i 位（最高位为第 N 位，最低位为第 1 位），当前已经填写的数字之和是 t，当前数值对 sum 取模的余数是 q。我们从小到大枚举第 i 位的数字 p。

- 若 p 小于上限 R 在第 i 位上的数字，则后边 $i-1$ 位可以随便填。因为最终的数值能被 sum 整除，所以第 $1 \sim i$ 位构成的数值对 sum 取模的余数为 $sum - q$。因此答案直接累加 $f[i-1][sum-t-p][sum][(sum-q-p*10^{i-1}) \bmod sum]$。
- 否则，令 $t = t + p, q = (q + p * 10^{i-1}) \bmod sum$，处理第 $i-1$ 位。