Day2 讲题 + 斯特林数

Scape

June 10, 2018

得分情况

100 分:1 人

题解

为了方便少点符号我们不妨考虑 $H = W = 4$ 的情形
注意到我们要干的事情大概是求一个方程的解数
我们可以尝试变换这些方程, 但是注意还是要保证方程和原方程等价性
于是就变成了

$$a[x][y] + a[x+4][y+4] = a[x][y+4] + a[x+4][y]$$

$$\sum_{i=1}^{4} a[x][i] = \sum_{i=1}^{4} a[x+4][i]$$

$$\sum_{i=1}^{4} a[i][x] = \sum_{i=1}^{4} a[i][x+4]$$

来考虑这些方程的一些性质

题解

注意到我们可以把 x, y 膜 4 同余的放一起处理
然后我们注意到一个性质, 对于 $x, y \leq 4$, 必然有
$a[x][y] = a[x+4][y] = a[x+8][y] \cdots$ 或
$a[x][y] = a[x][y+4] = a[x][y+8] \cdots$
于是我们 2^{16} 枚举一下每个数是向下相等还是向右相等, 然后向下向右是独立的, 就可以直接用一个组合数来算了
注意到为了防止重复算, 向下相等的不能向右相等, 所以要多容斥一下
时间复杂度 $O(2^{HW}(H+W)^2)$

得分情况

100 分:3 人
60 分:1 人

题解

我们先来证明这样一个事情。

令 $LCM = lcm(a_1, \cdots, a_n)$

对于相同的 k,$f(i * LCM + k)$ 是一个关于 i 的 N 次多项式

我们令 $g(i)$ 表示每个 a_i 的体积和都不超过 LCM 并且他们体积和是 i 的方案数, 我们可得

$f(i * LCM + k) = \sum_{j=0}^{n-1} g(k + LCM * j) \times C_{i-j+n}^{n-1}$

把 $g(i)$ 看成常数这就是一个 N 次多项式

我们接下来要求这个的前缀和，也就是一个 $N + 1$ 次多项式, 暴力之后拉格朗日插值就可以了

时间复杂度 $O(N^2 \prod a_i)$

得分情况

60 分:1 人。

题解

对于部分分，可以简单地 $O(ans)$ 地构造出答案
考虑递归构造

题解

我们现在列出了两排点，并且以两排点的最后一个点为结尾的拓扑序个数分别为 (x, y)
考虑在第一列后面新加一个点，并且把第一列原来的最后一个点向新点连边，然后我们将第二列的倒数第二个点连向这个新点。
仔细观察后不难发现，现在以两排点最后一个点为结尾的拓扑序个数为 $(x + y, y)$
于是可以辗转相减构造，感性理解这样点数大概是 $O(logans)$ 级别的（即初始的 (x,y) 取在 fib 附近，这样只会辗转相减 log 次
注意到要求构造的拓扑序个数很小，所以直接枚举最开始的状态 $(x, ans - x)$ 直接枚举也可以

第一类斯特林数

只考虑无符号的第一类斯特林数
$s(n,k)$ 表示把 n 个数划分成 k 个圆排列的方案数。
也可以说是长度为 n 的排列有 k 个循环的方案数。
$s(n,k)=\sum_{i=1}^{n} s(n-i,k-1)\times(i-1)!\times\binom{n-1}{i-1}$
$s(n,k)=s(n-1,k-1)+s(n-1,k)\times(n-1)$

随便来点例题

循环个数在区间 $[L, R]$ 内的长度为 n 的置换个数。
对 998244353 取模。
$n \leq 10^5$。

Sol

就是算 $\sum_{i=L}^{R} s(n,i)$
$s(n,k) = s(n-1,k-1) + s(n-1,k-1) \times (n-1)$
$x^{\overline{n}} = \prod_{i=1}^{n}(x+i-1) = \sum_{i=0}^{n} s(n,i)x^i$
分治 FFT，$O(n\log^2 n)$

Sol

考虑倍增

$f_n(x) = \prod_{i=1}^{n}(x + i - 1)$

$f_{2n}(x) = f_n(x) \times f_n(x + n)$

令 $f_n(x) = \sum_{i=0}^{n} A^i x^i$

$f_n(x + n) = \sum_{i=0}^{n} \sum_{j=i}^{n} A_j \times \binom{i}{j} \times n^{j-i}$

$T(n) = T(n/2) + O(n \log n)$

时间复杂度 $O(n \log n)$。

第二类斯特林数

$S(n,k)$ 表示把 n 个数划分成 k 个集合的方案数。
$\sum_{i=0}^{n} S(n,i) = B_n$
$s(n,k) = \sum_{i=1}^{n} s(n-i,k-1) \times \binom{n-1}{i-1}$
$s(n,k) = s(n-1,k-1) + s(n-1,k) \times k$

怎么快速算某一项

注意到 $m^n = \sum_{i=0}^{m} S(n,i) \times i! \times \binom{m}{i}$
二项式反演一下就得到 $S(n,m) = \frac{1}{m!}\sum_{k=0}^{m}(-1)^k\binom{m}{k}(m-k)^n$
FFT，$O(n\log n)$。

数的幂

$x^n = \sum_{k=0}^{n} x^{\underline{k}} S(n,k)$

JZPTREE

给一棵 n 个节点边权为 1 的树，对于每一个 i 求出 $\sum_{j=1}^{n} dist(i,j)^m$

$n \le 5 \times 10^4, m \le 500$

JZPTREE

维护 i 次方和。
$(x+1)^i = \sum \binom{i}{j} x^j$。
$O(nm^2)$

JZPTREE

维护 $x^{\underline{i}}$
$(x+1)^{\underline{i}} = (x+1)x^{\underline{i-1}} = x^{\underline{i}} + i \times x^{\underline{i-1}}$。
$x^n = \sum_{k=0}^{n} x^{\underline{k}} S(n,k)$
$O(nm)$

Costly Graph

对于一张无向图，对答案的贡献为所有节点的权值和。
一个节点的权值为它的度数的 m 次方。
求 N 个点带标号简单无向图的权值和。
对一个 NTT 模数取模。
$N \le 10^9, m \le 10^5$

Costly Graph

每个节点都是等价的
对答案的贡献为 $2^{\binom{N-1}{2}}\sum_{i=0}^{n-1}\binom{n-1}{i}i^m$
考虑算右边那个部分。

Costly Graph

$\sum_{i=0}^{n} \binom{n}{i} i^{m}$
$\sum_{i=0}^{n} \binom{n}{i} \sum_{j=0}^{m} S(m,j) i^{\underline{j}}$
$\sum_{j=0}^{m} S(m,j) \sum_{i=j}^{n} \binom{n}{i} i^{\underline{j}}$
$\sum_{j=0}^{m} S(m,j) \sum_{i=j}^{n} \binom{n-j}{i-j} n^{\underline{j}}$
$\sum_{j=0}^{m} S(m,j) n^{\underline{j}} \times 2^{n-j}$
FFT $O(m \log m)$

火车题

一张无向图的权值为联通块个数的 m 次方。
求所有的 n 个点的带标号无向图权值和。
对 998244353 取模。
$T \leq 1000, n \leq 30000, m \leq 15$。

火车题

又是第一喜欢的吊打 std 环节。

火车题

令 x_i 表示联通块 i 是否存在，那么图的权值为 $(\sum x_i)^m$。
把这个玩意用斯特林数展开之后看下组合意义。
对于某 k 个联通块，如果他们在图中同时出现，那么对答案的贡献为 $S(m,k) \times k!$。

火车题

f_i 表示 i 个点的无向连通图个数。
$g_{i,j}$ 表示 i 个点 j 个联通块的无向图个数。
f_i 大家都会算
$g_{i,j} = \sum_{k=1}^{i} \binom{i-1}{k-1} f_k \times g_{i-k,j-1}$
答案为 $\sum_{i=1}^{n} \sum_{j=1}^{m} \binom{n}{i} g_{i,j} \times 2^{\binom{n-i}{2}} \times j! \times S(m,j)$

CountTables

$w \times h$ 的网格，用 C 种颜色染色，要求满足任意两行或两列都互不相同。
$w, h, C \leq 4000$。

CountTables

有 C^m 本质不同的行，任意两行不相同的方案数是 $\binom{C^m}{n}$
如果列有 k 个等价类，那么方案是 $\binom{C^k}{n}$

CountTables

f_i 为至少 i 个等价类,g_i 为恰好 i 个等价类。
$f_i = S(m, i) \times \binom{C^i}{n}$
$g_i = f_i - \sum_{j=1}^{i-1} g_j \times S(i, j)$
$O(m^2)$

unknown

n 个点 m 条边的连通图方案数?
$n, m \leq 50$。

unknown

暴力 dp？
$O(n^6)$

unknown

有等式

$$\sum_{k=1}^{n} S(n,k) \times (k-1)! \times (-1)^{k-1} = [n=1]$$

unknown

算出 f_k 表示 n 个点划分成 k 个联通块，联通块之间不能有边，一个联通块内不需要联通，总共有 m 条边的方案数。
注意到一个有 n 个联通块的图，会在 f_k 里算 $S(n,k)$ 次，只要在 f_k 前乘上 $(-1)^{k-1}(k-1)!$ 就可以正确地算贡献。
也就是要算 $\sum_G(-1)^{c(G)}(c(G)-1)!\binom{e(G)}{m}$
其中 G 是每个联通块都是完全图的图。

unknown

设 $f_{n,k,e(G)}$ 表示 n 个点,k 个联通块，$e(G)$ 大小的方案数，转移枚举 1 号点所在联通块大小。
可以做到 $O(n^5)$。

unknown

再看一眼这个式子

$$\sum_G (-1)^{c(G)}(c(G)-1)!\binom{e(G)}{m}$$

设 $f_{n,m}$ 表示 $\sum_{G,e(G)=m}(-1)^{c(G)-1}(c(G)-1)!$
那么答案就是

$$\sum_i f_{n,i}\binom{i}{m}$$

unknown

怎么算 f 呢。
考虑不要枚举 1 号点所在的连通块，而是任意枚举一个连通块，这样一张图会被计算到 $c(G)!$ 次。那么如果我们枚举 1 号点以外的任意连通块，一张图就会被算到恰好 $(c(G)-1)!$ 次了。这一步的复杂度变成了 $O(n^4)$。

Count the Buildings

n 幢楼高度分别为 1 到 n，你需要排列这些楼的相对位置使得从左看去恰好有 x 幢楼，从右看去恰好有 y 幢楼，问方案数。
$T \leq 10^5, n \leq 2000$。

Count the Buildings

假设最高的楼的位置固定，最高楼的编号为 n，那么我们为了满足条件，可以在楼 n 的左边分 $x-1$ 组，右边分 $y-1$ 组，且用每组最高的那个元素代表这一组，那么楼 n 的左边，从左到右，组与组之间最高的元素一定是单调递增的，且每组中的最高元素一定排在该组的最左边，每组中的其它元素可以任意排列（相当于这个组中所有元素的环排列）。右边反之亦然。
然后，可以这样考虑这个问题，最高的那个楼左边一定有 $x-1$ 个组，右边一定有 $y-1$ 个组，且每组是一个环排列。我们可以先把 $n-1$ 个元素分成 $x-1+y-1$ 组，然后每组内部做环排列。再在所有组中选取 $x-1$ 组放到楼 n 的左边。所以答案是 $S(n-1, x+y-2) \times \binom{x+y-2}{x-1}$。

TJOI2016 求和

$f_n = \sum_{i=0}^{n} \sum_{j=0}^{i} S(i,j) \times 2^j \times j!$
$n \le 10^5$

TJOI2016 求和

$g_n = \sum_{i=0}^{n} S(n, i) \times 2^i \times i!$
意义是把 n 个不同数放入若干个集合，且每个集合有两种状态的方案数
$g_n = \sum_{i=1}^{n} 2\binom{n}{i} g_{n-i}$

SRM686 CyclesNumber

求所有长度为 n 的置换的循环个数的 m 次方和，对一个大质数取模。
$n \le 10^5, m, t \le 500$

SRM686 CyclesNumber

令 x_i 为循环 i 是否出现，那么答案为 $(\sum x_i)^m$
对于某 k 个循环，如果他们同时出现，那么对答案的贡献为 $S(m,k) \times k!$
我们考虑怎么计算每一个 k 的出现次数。

SRM686 CyclesNumber

$s(n+1,k+1)$
我们试图构造一个长度为 n 的置换中选出 k 个循环到有 $k+1$ 个循环的$\boxed{\text{F}}$度为 $n+1$ 的置换的双射。对于一个长度为 n 的排列，我们新增加一个元素 $n+1$ 把没有被选中的循环拼接起来。
如果没有被选取的元素从小到大为 $a_1,\cdots,a_n$，它们对应的元素是 $x_1,\cdots,x_n$，增加循环 $n+1- > x1- > x2- > \cdots - > n+1$。

SRM686 CyclesNumber

$\sum_{k=1}^{m} s(n+1, k+1) \times k! \times S(m, k)$

Description

Description

- 求出满足

$$\sum_{i=1}^{m} x_i \le s$$

$$\forall i \le m, x_i > 0$$

$$\forall i \le n, x_i \le t$$

的解数

Description

- 求出满足

$$\sum_{i=1}^{m} x_i \le s$$

$$\forall i \le m, x_i > 0$$

$$\forall i \le n, x_i \le t$$

的解数

- $m - n \le 1000, m \le 10^9, t \le 10^5, nt \le s \le 10^{18}$。

Description

- 求出满足

$$\sum_{i=1}^{m} x_i \leq s$$

$$\forall i \leq m, x_i > 0$$

$$\forall i \leq n, x_i \leq t$$

  的解数
- $m - n \leq 1000, m \leq 10^9, t \leq 10^5, nt \leq s \leq 10^{18}$。
- 我的集训队自选题，TCO2013 Round 3A Hard 增强版

Solution 1

Solution 1

- 首先我们知道答案就是

$$\sum_{(\sum_{i=1}^{n} x_i)=k(\forall 1\le i\le n, 1\le x_i\le t)} \binom{s-k}{m-n}$$

Solution 1

- 首先我们知道答案就是

$$\sum_{(\sum_{i=1}^{n} x_i)=k(\forall 1\le i\le n,1\le x_i\le t)} \binom{s-k}{m-n}$$

- 显然答案是关于 k 的 $m-n$ 次多项式

Solution 1

- 首先我们知道答案就是

$$\sum_{(\sum_{i=1}^{n} x_i)=k(\forall 1\le i\le n, 1\le x_i\le t)} \binom{s-k}{m-n}$$

- 显然答案是关于 k 的 $m-n$ 次多项式
- 那么对于每一个特定的 k, 答案就可以写成形如 $\sum_{i=0}^{m-n} a_i k^i$ 的形式

Solution 1

- 首先我们知道答案就是

$$\sum_{(\sum_{i=1}^{n} x_i)=k(\forall 1\le i\le n,1\le x_i\le t)} \binom{s-k}{m-n}$$

- 显然答案是关于 k 的 $m-n$ 次多项式
- 那么对于每一个特定的 k, 答案就可以写成形如 $\sum_{i=0}^{m-n} a_i k^i$ 的形式
- 注意到无论 k 取何值,a_i 都是不变的

Solution 1

- 首先我们知道答案就是

$$\sum_{(\sum_{i=1}^{n} x_i)=k(\forall 1\le i\le n, 1\le x_i\le t)} \binom{s-k}{m-n}$$

- 显然答案是关于 k 的 $m-n$ 次多项式
- 那么对于每一个特定的 k, 答案就可以写成形如 $\sum_{i=0}^{m-n} a_i k^i$ 的形式
- 注意到无论 k 取何值,a_i 都是不变的
- 于是答案就变为

$$\sum_{k=0}^{m-n} a_k \sum_{x_1=1}^{t} \sum_{x_2=1}^{t} \cdots \sum_{x_n=1}^{t} (\sum_{i=1}^{n} x_i)^k$$

Solution 1

Solution 1

- a_k 是容易通过把组合数暴力展开求出的

Solution 1

- a_k 是容易通过把组合数暴力展开求出的
- 令

$$f(n,k)=\sum_{x_1=1}^{t}\sum_{x_2=1}^{t}\cdots\sum_{x_n=1}^{t}(\sum_{i=1}^{n}x_i)^k$$

$$=\sum_{x_1=1}^{t}\sum_{x_2=1}^{t}\cdots\sum_{x_n=1}^{t}(\sum_{i=1}^{n-1}x_i+x_n)^k$$

$$=\sum_{x_1=1}^{t}\sum_{x_2=1}^{t}\cdots\sum_{x_n=1}^{t}\sum_{i=0}^{k}(\sum_{i=1}^{n-1}x_i)^i x_n^{k-i}\binom{k}{i}$$

$$=\sum_{i=0}^{k}f(n-1,i)\binom{k}{i}\sum_{x=1}^{t}x^{k-i}$$

Solution 1

Solution 1

- 对上面的递推式矩阵优化一下即可

Solution 1

- 对上面的递推式矩阵优化一下即可
- 时间复杂度 $O((m-n)^3 log(m-n) + t(m-n))$

Solution 2

Solution 2

- 为了方便在这里贴出用到的公式:

$$x^{\underline{n}} = \sum_{k=0}^{n} (-1)^{n-k} \left[{n \atop k} \right] x^k \tag{1}$$

$$x^n = \sum_{k} \left\{ {n \atop k} \right\} x^{\underline{k}} \tag{2}$$

$$\Delta^n f(x) = \sum_{i=0}^{d} c_i \binom{x}{i-n} \text{其中} f(x) = \sum_{i=0}^{d} c_i \binom{x}{i} \tag{3}$$

其中 $\left[{n \atop k} \right]$ 表示第一类斯特林数, $\left\{ {n \atop k} \right\}$ 表示第二类斯特林数

Solution 2

Solution 2

- 我们容斥一下，可知答案为 $\sum_{k=0}^{n}(-1)^k\binom{n}{k}\binom{s-kt}{m}$

Solution 2

- 我们容斥一下，可知答案为 $\sum_{k=0}^{n}(-1)^k\binom{n}{k}\binom{s-kt}{m}$
- $(-1)^k\binom{n}{k}$ 不好化简，我们考虑 n 阶差分，显然有

Solution 2

- 我们容斥一下，可知答案为 $\sum_{k=0}^{n}(-1)^k\binom{n}{k}\binom{s-kt}{m}$
- $(-1)^k\binom{n}{k}$ 不好化简，我们考虑 n 阶差分，显然有
- $$\Delta^n f(x)=\sum_{k=0}^{n}(-1)^{n-k}\binom{n}{k}f(x+k)$$

Solution 2

- 我们容斥一下，可知答案为 $\sum_{k=0}^{n}(-1)^k\binom{n}{k}\binom{s-kt}{m}$
- $(-1)^k\binom{n}{k}$ 不好化简，我们考虑 n 阶差分，显然有
- $$\Delta^n f(x)=\sum_{k=0}^{n}(-1)^{n-k}\binom{n}{k}f(x+k)$$
- 稍加处理就得到我们需要的形式，令 $f(x)=\binom{s-xt}{m}$，所求即 $\Delta^n f(0)$:

Solution 2

- 我们容斥一下，可知答案为 $\sum_{k=0}^{n}(-1)^k\binom{n}{k}\binom{s-kt}{m}$
- $(-1)^k\binom{n}{k}$ 不好化简，我们考虑 n 阶差分，显然有
- $$\Delta^n f(x)=\sum_{k=0}^{n}(-1)^{n-k}\binom{n}{k}f(x+k)$$
- 稍加处理就得到我们需要的形式，令 $f(x)=\binom{s-xt}{m}$，所求即 $\Delta^n f(0)$:
- 对差分的常用技巧是将其转化为牛顿级数，我们先试图将 $f(x)$ 转化成一个多项式，然后再转化成牛顿级数。

Solution 2

Solution 2

- 下降幂可以用斯特林数很容易地转化成多项式（公式 (1)）:

Solution 2

- 下降幂可以用斯特林数很容易地转化成多项式（公式 (1)）:
- $$\binom{s-xt}{m} = \frac{1}{m!}\sum_{j=0}^{m} x^j \sum_{i=j}^{m} (-1)^{m-i+j} \binom{i}{j} \begin{bmatrix} m \\ i \end{bmatrix} t^j s^{i-j}$$

  继续转化多项式（利用公式 (2)），令多项式中 x^i 的系数为 a_i。

  $$\sum_{i=0}^{m} a_i x^i = \sum_{j=0}^{m} \binom{x}{j} \sum_{i=j}^{m} a_i \begin{Bmatrix} i \\ j \end{Bmatrix} j!$$

  于是得到牛顿级数，令 $\binom{x}{i}$ 的系数为 b_i，则 b_n 为所求（公式 (3)）。

Solution 2

Solution 2

- 将 a_i 代入，最终得到：

$$\sum_{k=0}^{n}(-1)^k\binom{n}{k}\binom{s-kt}{m}$$

$$=\frac{n!(-1)^{n+m}t^n}{m!}\sum_{i=0}^{m-n}\sum_{j=0}^{m-n-i}(-1)^j\binom{n+i+j}{j}\left\{{n+i \atop n}\right\}$$

$$\left[{m \atop n+i+j}\right]t^is^j$$

Solution 2

- 将 a_i 代入，最终得到：

$$\sum_{k=0}^{n}(-1)^k\binom{n}{k}\binom{s-kt}{m}$$

$$=\frac{n!(-1)^{n+m}t^n}{m!}\sum_{i=0}^{m-n}\sum_{j=0}^{m-n-i}(-1)^j\binom{n+i+j}{j}\left\{ n+i \atop n \right\}$$

$$\left[m \atop n+i+j \right] t^i s^j$$

- 最后只剩下计算快速斯特林数的问题了，注意到有 $n+i-n\leq 1000, m-n-i-j\leq 1000$。

Solution 2

Solution 2

- 我们考虑斯特林数的组合意义。
  第二类斯特林数表示将 n 个元素分为 k 个非空子集的方案数。
  容易发现至多只有 $n-k$ 个子集的大小大于 1。
  我们暴力枚举大于 1 的子集个数，则有:

$$\left\{ n \atop k \right\} = \sum_{i=0}^{n-k} \binom{n}{n-k+i} g(n-k+i, i)$$

其中 $g(n,k)$ 为将 n 个元素分为 k 个大小大于 1 的子集的方案数。
考虑第 n 个元素要么与此前一个元素一起成为一个新集合，要么加入之前的集合，则有
$g(n,k) = kg(n-1,k) + (n-1)g(n-2,k-1)$

Solution 2

Solution 2

- 第一类斯特林数表示将 n 个元素分为 k 个非空环排列的方案数。
  类似第二类斯特林数，有

$$\begin{bmatrix} n \\ k \end{bmatrix} = \sum_{i=0}^{n-k} \binom{n}{n-k+i} f(n-k+i, i)$$

  其中 $f(n,k)$ 为将 n 个元素分为 k 个大小大于 1 的环排列的方案数。
  我们用一个序列表示这些环排列，每个环排列我们令其中最小的元素排在第一个，排列之间用一个特殊元素分割，考虑第 n 个元素要么与此前一个元素一起成为一个新环排列，要么插入之前的环排列构成的序列，则有：
  $f(n,k) = (n-1)f(n-1,k) + (n-1)f(n-2,k-1)$
  因此最终复杂度为 $O((m-n)^2)$。