以下计数题答案全部模 998244353 。

CF715E

https://codeforces.com/contest/715/problem/E

给定两个排列 p,q，但是其中有些位置未知，用 0 表示。

现在让你补全两个排列，定义两个排列 p,q 之间的距离为每次选择 p 中两个元素交换，使其变成 q 的最小次数。

现在你需求出对于 $i\in[0,n-1]$ 求出补全后相似度为 i 的方案数。

$n\le 250$。

sol

考虑相似度是什么。对于确定的 p,q ，可以一通操作使得 q 变成 $1,2,\cdots,n$ ，然后就很好做。最后发现相似度是 $n-\text{cycle}(p,q)$ ，其中 $\text{cycle}(p,q)$ 表示连边 $p_i\to q_i$ 得到的环数。

先把已经存在的边连出来。因为只关心环数，所以可以把链先缩掉。最后等价于 $p=[1,2,\cdots,a,0,0,\cdots,0],q=[0,\cdots,0,n-b+1,n-b+2,\cdots,n]$ ，其中 $a+b\le n$ 。设 $c=n-a-b$ 。设 a,b,c 分别为一二三类点。

容易发现对最终的图的唯一限制就是不能存在一类点连向二类点的边。并且图的个数乘上 $c!$ 就是排列个数。

那么图中的一个环要么只有一类点或二类点，要么可以被拆成若干条链，每条链都形如 $a_1\to a_2\to\cdots\to c_0\to b_1\to\cdots\to b_k$ 。

所以就是选出若干个一类点造环，选出若干个二类点造环，再把剩下的点与一些三类点连成链，再用 c 个点造环。

造环非常简单，用斯特林数乱搞即可。连成链的方案数略微复杂一点，注意到一类点和二类点独立，然后容斥一下。

CF1508F

https://codeforces.com/contest/1508/problem/F

给定 $1 \sim n$ 的排列 $a_1, a_2, \cdots, a_n$ 和 q 个区间 $[l_1, r_1], [l_2, r_2], \cdots, [l_q, r_q]$。定义：

- 排列 $b_1, b_2, \cdots, b_n$ 对区间 $[l', r']$ **保序**当且仅当 $\forall l' \le x < y \le r'$ 满足 $[a_x < a_y] = [b_x < b_y]$。
- 排列 $b_1, b_2, \cdots, b_n$ 是 k-**相似的**当且仅当 $\{b_n\}$ 对所有 $[l_i, r_i]$ $(1 \le i \le k)$ 保序。
- 一个 DAG 是对 k-相似排列的**编码**，当且仅当其所有拓扑序与所有 k-相似排列一一对应。

对于 $k = 1, 2, \cdots, q$，求对 k-相似排列的编码的最少边数。

$n \le 2.5 \times 10^4, q \le 10^5$。

sol

根据经验知道每个点只需要保留往左往右连向最小点的边。

容易在 $O(\log n)$ 的时间内求出一条边会保留多久，但是可能存在的边似乎很多，但是好像又应该不是太多。

能够存在的边都一定在某个区间内相邻，所以对给出的这些区间做一个莫队即可。

CF1237G

https://codeforces.com/problemset/problem/1237/G

有n个朋友住在一条环形的街道上，他们和他们的房子按顺时针标号为0到$n-1$。

一开始第i个人有a_i块石头。他们想让他们之间石头分配得完美均衡：每个人都应拥有相同数量的石头。

改变石头分布的唯一途径是举行会议。在一次会议中，连续k个房子的人（记住这条街道是环形的）聚集在同一个地方并带上他们的石头。所有带来的石头可能会在参与会议的人中任意重新分配。会议结束后，每个人回到自己的房子。

找到一种方案使得把石头分配得完美均衡且举行尽量少的会议。

输出举行会议的次数及每次会议的描述。

$2 \le k < n \le 10^5, 0 \le a_i \le 10^4$ 。

sol

对于一个环，一个容易想到的构造方法是选择一个点开始，每次把下面 $k-1$ 个位置都摆好，然后以最后那个位置为开头接着做。

怎样分析它的优秀之处呢？这种做法可能会使得某些位置被覆盖两次，但是脑洞一下可以发现，如果在相邻两个人之间插板，那么每块板只会被插一次。

所以现在考虑所有没有被插过的板，从这里切开。对于每个线段，如果长度为 l ，那么一个显然的下界是 $(l-1)/(k-1)$ 上取整。手玩一下发现可以取到。

因为是上取整，所以只要 $(l-1) \ne 0 \pmod{(k-1)}$ 那么就可以把这段和相邻的某一段合并，答案不会更劣。

另外一个限制是每段之和都等于 $l \cdot avg$ ，相当于每个位置都减去 avg 之后和都为 0 。

现在已经可以随便倍增了。

CF1264F

https://codeforces.com/problemset/problem/1264/F

给你一个长度为 n 的递增的等差正整数数列 $(a, a+d, a+2d \ldots a+(n-1)d)$。

需要你构造一个长度为 n 的递增的等差正整数数列 $(b, b+e, b+2e \ldots b+(n-1)e)$ 满足以下条件:

- $0 < b, e < 2^{64}$
- 对于所有的 $0 \leq i < n$，$a+id$ 的十进制表示是 F_{b+ie} 的十进制表示的后 18 位的子串。（如果 F_{b+ie} 没有 18 位，那么考虑它的所有位）

其中 F_i 是指斐波那契数列的第 i 项（$F_0=0, F_1=1$）。

$a+(n-1)d < 10^6$ 。

sol

因为样例没有无解的情况，所以猜想一定有解。然后因为任意一个等差数列都是 $1, 2, \cdots, 10^6$ 的子列，所以显然只需要把这个做出来就可以做出任意的等差数列。

考虑斐波那契数列有什么性质。因为拿出来的位置也是等差数列，所以考虑从这个位置加上 e 会变成什么。用到公式 $F_{m+n}=F_mF_{n+1}+F_{m-1}F_n$ 。

为了让操作可控，我们希望 F_n 形式好看。根据循环节的性质，令 $n=12*10^9$ ，那么 F_{kn} 就都是 10^9 的倍数。

直接算，算出 $F_{n+1}=8t\cdot 10^9+1$ ，其中 $\gcd(t,10)=1$ 。

首项也应该取特殊值。根据上面那个公式递推，得到 $F_{un+1}=8ut\cdot 10^k+1 \pmod{10^{2k}}$ 。

那么可以用 u 把 $8t$ 消掉，那就做完了。

CF1477F

https://codeforces.com/problemset/problem/1477/F

给定 n 个长度为 a_i 的巧克力，每次以正比于 a_i 的概率取得一个巧克力，然后在 $(0, a_i)$ 中随机选择一个实数 r 并将其分成 $r, a_i - r$ 两个部分放回。

计算使得所有巧克力的长度均小于 k 的期望操作次数。

$n \le 50, \sum a_i, k \le 2000$ 。

sol

题目的随机方法显然等价于初始已经割了若干刀，然后每次随机一个位置再切。设 L 为总长度。

先考虑 $n = 1$ 的情况。枚举切了 j 刀后仍**未完成**，然后容斥还剩多少段没完成，有式子

$$\begin{aligned}&\sum_{j\ge 0}\sum_{i=1}^{L/k}(-1)^{i-1}(1-\frac{ik}{L})^j\binom{j+1}{i}\\&=\sum_{i=1}^{L/k}(-1)^{i-1}(1-\frac{ik}{L})^{i-1}(\frac{L}{ik})^{i+1}\end{aligned}$$

然后考虑怎么处理 $n \ne 1$ 。自然的想法是把这些刀分配给每一段，那么就要求出每一段切了若干刀之后**完成**的 EGF 。需要注意的是， EGF 里面把 $\frac{a_i}{L}$ 这个概率也算进去了。

设 $m = a_i$ ，那么列出式子

$$\sum_i \frac{1}{i!}(\frac{m}{L}x)^i\sum_{j=0}^{m/k}(-1)^j\binom{i+1}{j}(1-\frac{jk}{m})^i$$

然后是繁琐的计算，最终算出

$$\sum_{j=0}^{m/k}(-1)^j\frac{1}{j!}\left(j((\frac{m-jk}{L})x)^{j-1}\exp{(\frac{m-jk}{L}x)}+(\frac{m-jk}{L})^jx^j\exp{(\frac{m-jk}{L}x)}\right)$$

然后要把这些多项式卷在一起。x 的次数不超过 L/k ，而 $\exp$ 里面的东西只需要记录 $\sum j$ ，也是 L/k 级别。并且注意到 $\exp$ 中的 j 和 x 的次数在单个 EGF 中不会超过 1 ，所以整体不会超过 n 。所以一共只有 $O(nL)$ 项，暴力卷积 $O(nL^2)$ ，用 NTT 优化做到 $O(nL\log L)$ 。

答案相信大家都会算。

AGC040F

https://atcoder.jp/contests/agc040/tasks/agc040_f

有两个**不可区分**的棋子放在数轴上。初始都在 0 点。

可以进行以下两种操作：

- 选择一个棋子，向右移动一格。
- 把坐标较小的那个棋子移动到坐标较大的棋子的位置。如果两个棋子在相同位置那么仍然可以这样操作。

做 n 次操作，使得一个棋子的位置在 A ，另一个在 B 。求出移动棋子的方案数。

由于棋子是不可区分的，所以两种方案不同当且仅当存在一个时刻 i ，使得棋子的位置集合不同。

$n \le 10^7$ 。

sol

想象有初始为空的两行。把操作 1 看做给某一行加一个球，把操作二看做加一根棍子把之后的操作和之前的操作割隔开。

因为题目的判重条件，可以认为每一段中第一行的球数都大于等于第二行的球数。那么第一行会有 B 个球，第二行不确定。但是最后一段中第二行会比第一行少 $B-A$ 个球，而我们并不关心前面的段中球数具体怎么分布，因为不管怎么分布，最后一根棍子都会消除所有的差异。

然后转换到坐标系上，用 (x,y) 表示两个棋子的坐标分别是 x,y 。那么操作有三种：

- $x := x+1$ ；
- $y := y+1$ ，只能在 $y < x-1$ 的时候使用；
- $y := x$ 。

操作一用了 B 次，不妨枚举操作二用了 k 次。特判掉 $k+B=n$ 的情况。

假装我们确定了操作一和操作二之间的顺序，现在要加入 $n-k-B$ 个操作三进去。

“只能在 $y < x-1$ 的时候使用”，相当于除了刚进行完操作三的时候和刚开始的时候，不能碰到 $y=x$ 这条直线。

根据前面所说，最后一次操作三需要保证 $x-y=A-k$ 。而因为以后不会再碰到 $y=x$ 这条直线，所以在只考虑操作一二的时候，只有最后一次碰到 $x-y=A-k$ 时才能使用这最后一次操作三。

同理，只有在最后一次碰到 $x-y=A-k-1, A-k-2, \cdots, 0$ 的时候才能使用操作三。并且这些位置是肯定会被碰到过的。而且每次使用的次数是任意的。

那么就把操作三的插入和操作一二给分离了，只需要用卡特兰数算出 (A,k) 的方案数即可。

gym102978C

https://codeforces.com/gym/102978/problem/C

有 R 个相同的红球和 B 个相同的蓝球，以及一个绿球。把它们任意排列，定义一种排列的得分是

- 设 l_R, l_B, r_R, r_B 表示绿球左边/右边的红球/蓝球个数，那么得分就是 $\lfloor \min(l_R/l_B, r_R/r_B) \rfloor$ （分母为 0 视为无穷大）。

求出所有排列方法的得分之和。

$1 \le R \le 10^{18}, 1 \le B \le 10^6$ 。

sol

由于场上大家都没做出来，所以直接枚举左右各有几个红球蓝球的方法是肯定行不通的。

考虑先确定红蓝球的顺序，再给绿球放一个地方。把蓝球看做往右走，红球看做往上走，那么得到一条从 $(0,0)$ 走到 (B,R) 的路径。可以在路径上任选一个点放绿球，那么权值就是这个点到 $(0,0)$ 和 (B,R) 的斜率的较小值下取整。

枚举一个正整数 x ，求有多少个点使得两边的斜率都大于等于 x 。显然就是从 $(0,0),(B,R)$ 各引一条斜率为 x 的直线，求夹在中间的点数。

再变为枚举一条直线的截距，求和这条直线的交点（整点）个数。

> 现在的问题：给定 H, W, A, B ，求 $f(H,W,A,B)$ 表示从 $(0,0)$ 走到 (W,H) ，与 $Ax+B$ 的交点的个数之和。保证 $0 \le B \le AW+B \le H$ 。
>
> 由于某些原因，交点个数和不交的路径条数有一定联系。
>
> > 给定非负整数 W, A, B ，求从 $(0,0)$ 走到 $(W, AW+B)$ ，且不曾超过 $y = Ax+B$ ，的路径条数。
> >
> > 因为 A, B 均为整数，所以可以把直线向上平移一位，求从 $(0,0)$ 走到 $(W, AW+B+1)$ ，且只在终点处碰到 $y = Ax+B+1$ ，的路径条数。
> >
> > 设 f_W 表示这个东西，那么枚举第一次碰到在哪里，有
> >
> > $$\binom{AW+W+B+1}{W} = \sum_i f_i \binom{A(W-i)+(W-i)}{W-i}$$
> >
> > 显然这是卷积，那么 $\sum_n \binom{An+n}{n} x^n$ 是啥呢？根据广义二项级数（参考具体数学），这东西就是 $\frac{1}{-A+(A+1)\mathcal{B}_{A+1}^{-1}}$ 。并且 $\sum_n \binom{An+n+B+1}{n} x^n$ 是 $\frac{\mathcal{B}_{A+1}^{B+1}}{-A+(A+1)\mathcal{B}_{A+1}^{-1}}$ 。
> >
> > 于是 $F(x) = \mathcal{B}_{A+1}^{B+1}(x)$ ，也就是 $f_W = \binom{AW+W+B+1}{W} \frac{B+1}{AW+W+B+1}$ 。
> >
> > 这东西有另一个形式： $\binom{W+AW+B}{W} - A \cdot \binom{W+AW+B}{W-1}$ 。
> >
> > （也就是 $\binom{tk+r}{k} \frac{r}{tk+r} = \binom{tk+r-1}{k} - (k-1)\binom{tk+r-1}{k-1}$ ）
>
> 现在来干一件神奇的事情：考虑 $f(H,W,A,B) - Af(H+1,W-1,A,B)$ 。一起枚举两边的交点位置，就得到
>
> $$\sum_x \binom{Ax+B+x}{x} \left(\binom{H-(Ax+B)+W-x}{W-x} - A \binom{\cdots}{W-x-1} \right)$$
>
> 后面那个大括号和上面有点神似。通过配凑，得到 $W' = W-x, B' = H-B-AW$ 。也就是说，它恰好等于从 $(x, Ax+B)$ 走到 (H,W) ，且从未走到 $y = Ax+B$ 下方的方案数。

> 这个“从未走到下方”比较难受。我们把后面的路径向上平移一格，然后要求不再碰到 $y = Ax + B$（显然，为了达成这个条件，这一步只能往上走，所以后面没有数多）。那么就其实是在枚举最后一次碰到 $y = Ax + B$ 在哪里。所以整体求和就是从 $(0, 0)$ 走到 $(W, H + 1)$ 的方案数，即 $\binom{W+H+1}{W}$。
>
> ！！！
>
> $f(H, W, A, B) - Af(H + 1, W - 1, A, B) = \binom{W+H+1}{W}$ ！
>
> 我们还有边界条件 $f(W + H, 0, A, B) = 1$，所以可以推出
>
> $$f(H, W, A, B) = \sum_{i=0}^{W} \binom{W + H + 1}{i} A^{W-i}$$
>
> 这东西甚至还和 B 无关！

于是现在可以直接列出式子

$$\sum_{x=1}^{R/B} (R - xB + 1) \sum_{i=0}^{B} \binom{R + B + 1}{i} x^{B-i}$$

这只需要求 B 个幂和即可。

gym102978J

https://codeforces.com/gym/102978/problem/J

给定长度为 n 的**不降正整数**序列 A ，对于每个 $0 \le k \le n$ ，求出满足下面条件的长为 n 的**不降非负整数**序列 x 的个数。

- $x_i \le A_i$
- $\sum[x_i = A_i] = k$

$n, A_i \le 250000$ 。

sol

先对 A 差分，表示 i 增加 1 之后多出现了几个可选项。

然后可以转化为：有一个 01 序列 X 和一个初始为空的栈 S ，从小到大扫 X ，执行以下操作：

- 如果 $X_i = 0$ ，那么向栈中塞入一个元素。
- 如果 $X_i = 1$ ，那么从栈中弹出任意个元素（可以弹 0 个）。如果弹之后栈为空那么这个位置是好的。

求有多少种操作的方法，使得恰好有 k 个好位置。

这也可以理解为括号序列， $X_i = 0$ 表示加入一个左括号， $X_i = 1$ 表示加入若干个右括号，求有多少个位置的右括号能恰好把左括号匹配完。

我们反过来做，维护一个初始为空的栈 T ，从大到小扫 X ，执行以下操作：

- 如果 $X_i = 1$ ，那么向栈中加入一个元素。
- 如果 $X_i = 0$ ，那么弹出任意个元素（可以弹 0 个），表示这个左括号到哪里才被匹配掉。

求有多少种操作的方法，使得操作结束时栈中恰好剩下 k 个元素。

然后考虑把过程对应到坐标系上。本来是每个横坐标有一个上界，从 $(1,0)$ 开始走到 $(n+1, A_n)$ 。转化之后变为每个纵坐标有一个下界，从 $(n+1, A_n)$ 走到 $(k+1, 1)$ （最后一步只能往下走到 $(k+1, 0)$ ），所以剩下 k 个元素在栈中）。

这恰好对应 $(A_{k+1} - 1, \cdots, A_n - 1)$ 的合法方案，也就是说现在不需要考虑有多少个地方顶到上界了。

然后算方案。一个无脑 DP 是把最后一列 $(n, ?)$ 的 DP 值赋为 1 ，然后每次把这一列的 DP 值推到前一列并做后缀和（把上面的往下推）。

然后大力分治，

- Lowerl left part: $(A_1, \ldots, A_{N/2})$
- Middle part: $(A_{N/2}, A_{N/2}, \ldots, A_{N/2})$
- Upper right part $(A_{N/2+1} - A_{N/2}, A_{N/2+2} - A_{N/2}, \ldots, A_N - A_{N/2})$

先做右上角，得到 $(i, A_{N/2} + 1)$ 位置的 DP 值，然后用这些 DP 值推出 $(N/2, j)$ 的 DP 值，然后做左下角。

UOJ597

https://uoj.ac/problem/597

给定整数 n, y 和一个长度为 n 的整数序列 $A = \{a_1, a_2, \cdots, a_n\}$，**保证序列 A 单调不减或单调不增。**

构建有向图 $G(V, E)$，其中 $V = \{1, 2, \cdots, n\}$，$E = \{(i, j) \mid 1 \le i, j \le n, a_i \ge j\}$。注意图 G 中可能包含自环。

定义边子集 $T \subseteq E$ **合法**当且仅当图 $G'(V, T)$ 中每个点的入度和出度不超过 1，自环对对应点的入度和出度均贡献 1。定义一个合法边子集 T 的**权值**为 $y^{\text{cycle}(T)}$，其中 $\text{cycle}(T)$ 表示图 $G'(V, T)$ 的环数，自环是一个环。

特别地，本题认为 $0^0 = 1$。

对于所有整数 $0 \le k \le n$，求所有大小为 k 的合法边集的权值和。

$n \le 10^5$。

sol

相信出题人是良心的，先看 A 单调不减的情况。

又有链又有环显得很乱，先考虑每个点都连边的情况。

随机游走之后决定每个环在编号最大的点封口。注意到对于 $x < y$，如果有边 (x, y)，那么一定有边 (y, x)，所以可以从左往右看每个点，如果不是所在环的最大点那么随便连一条边，否则就直接连向所在链的起点。因为上面所说，这条边一定存在。

这有一个小问题：如果 $a_i < i$，那么前面就没有点能连到它，所以无法封口。还有一个小问题是无法保证每个环都被考虑到了，但是只需要假装没被考虑到的环也是一种颜色即可。

所以答案就是 $\prod_i (a_i - (i-1) + [a_i \ge i](y-1))$。

那么再考虑上有些点不连边的情况呢？设 $dp_{i,j}$ 表示前 i 个点有 j 个点没有连边，有转移式

$$dp_{i,j} = ((a_i - (i-1) + j) + [a_i \ge i](y-1))dp_{i-1,j} + dp_{i-1,j-1}$$

然后用多项式技巧搞一搞。

然后看 A 单调不增的情况。

由对称性，感受到应该从大到小考虑。设 b_i 为能连到 i 的编号最大的点，即数据范围中的那个。

想要用上面一样的做法，但是有可能 $a_i < b_i$，那么就不保证一定能封口了。

先考虑 a, b 的性质。容易发现 $a_i \ge i \Leftrightarrow b_i \ge i$，且满足这个的 i 是一个前缀。另外 b 也是单调不增的。

现在的问题是考虑到一个 i 的时候，可能后面有一个 j 连向了 i，但 i 无法连向 j。为了避免这种情况，我们把主动权交给 i：对于边 (x, y)，总是在 $\min(x, y)$ 处考虑它。

现在在 i 处，如果 $a_i \ge b_i$，那么可以先自由选择一个 j 连向自己，然后再看是否要用 a 封住。否则，同样可以选择一个 j 连过去，然后用 b 封口。

发现这个简单的 $O(n^2)$ DP 非常正确，但是 DP 式子略显冗长。能否弄成和上面一样简洁的式子呢？

仔细观察我们干了什么。发现对于 $a_i \ge i$ 的位置，在刚才的 DP 中 a_i, b_i 完全是等价的，相当于把 $i \le a_i < b_i$ 的位置的出入边反向了。我们观察一下反向之后的图长什么样子。

发现新图仍然可以被一组 a'_i 描述！

那么就可以写出类似上面的 DP 式子，同样可以用多项式技巧优化。