number

对于一个值域和定义域均为 U 的函数 $f(x)$，定义其将所有 $i=1,2,\cdots,n$ 由 i 向 $f(i)$ 连边形成的内向基环森林为 $T(f)$。

考虑对于某个 $g(x)$，$T(g^{n-1})$ 具有怎样的特征，将 $T(g^{n-1})$ 中 i 的出边视作在 $T(g)$ 中从 i 点沿出边方向走 $n-1$ 步的终点：对于 $T(g)$ 中在环上的点，在 $T(g^{n-1})$ 中其依然在环上；对于 $T(g)$ 中不在环上的点，在 $T(g^{n-1})$ 中其依然不在环上，但由于 $n-1$ 充分大，其在 $T(g^{n-1})$ 中与所在基环树上的环的距离必然为 1。

首先忽略 $T(g^{n-1})$ 中不在环上的点，容易发现如果有 i 个点在环上，那么环外的点的方案数即为 i^{n-i} （这是因为对于每一个环外点的连接方案，都容易反向构造出 $T(g)$），则我们只需要对于所有 $i=1,2,\cdots,n$ 计算出有 i 个点在环上的方案数 h_i。

接下来只需要考虑环长的限制，若 $T(g)$ 中有一个长度为 d 的环，那么在 $T(g^{n-1})$ 中其会被分裂为 $\gcd(d,n-1)$ 个长度为 $\dfrac{d}{\gcd(d,n-1)}$ 的环，记 $v_d=\dfrac{d}{\gcd(d,n-1)}$，对于每个 i，记所有 $v_d=i$ 的 d 形成的集合为 S_i，那么长度为 i 的出现次数 c 的限制即为：S_i 内元素除 i 后进行完全背包，c 需要能被若干可重物品的和表出，这里做完全背包的复杂度是总体 $O(n^2)$ 的。但通过对质因子的分析，S_i中必然有一个元素是其它元素的因子，因此可以省去完全背包。

那么，我们已经刻画了对于所有环长及其出现次数的限制，可以使用一个背包计算答案，每次枚举新加入的环长使用了多少个，使用组合数计算贡献。注意到对于长度为 i 的环，其至多使用 $O(\dfrac{n}{i})$ 个，那么由调和级数，总复杂度即为 $O(n\sum\limits_{i=1}^{n}\dfrac{n}{i})=O(n^2\log n)$。

yoimiya

考虑一个串 a，考虑刻画出所有的 i,k 使得 $a[i,i+k-1]=a[i+k,i+2k-1]$。

枚举 k，设 $b_i=[a_i==a_{i+k}]$，那么我们只需要找到 b 中所有长度 $\geq k$ 的连续段。

注意到这样的连续段的段数不超过 $\lfloor n/k\rfloor$，考虑一段一段找。发现长度 $\geq k$ 就至少需要经过 $k,2k,\cdots$ 这些位置中的一个，考虑枚举 $x=ik$，算出经过 x 的极长连续段长度。那么只需要查询 $[1,x],[1,x+k]$ 的最长公共后缀与 $[x,n],[x+k,n]$ 的最长公共前缀即可。SA 配合 ST 表容易做到 $O(n\log n)$。

现在我们找到了一个极长的连续段 $[L,R]$，那么相当于对每个 $i=L,L+1,\cdots,R-k$，连边 $(i,i+k)$。然后对每个连通块，把他的 L 限制取 max，R 限制取 min，每个连通块的 $(R-L+1)$ 的乘积就是答案。

注意到合并只会发生 $O(n)$ 次，考虑快速找到所有需要合并的位置。考虑建 $O(n\log n)$ 个虚点，(i,j) 代表 $[i,i+2^j-1]$ 这个区间整体的合并情况，然后用并查集维护每一层所有点的连通情况。

换言之，如果 $(x,j),(y,j)$ 已经连通，代表着我们已经确定 $b[x,x+2^j-1]=b[y,y+2^j-1]$。

那么每次区间连边我们首先拆成两个长为 2^k 的区间连边，然后自顶向下递归，即可在 $O(\alpha(n)\times\log n)$ 的时间内找到一个未被合并的位置。然后并查集在合并的时候处理一下对答案的影响即可。

由调和级数可知外层的连边次数不超过 $O(n\log n)$；另一方面在并查集上面每次连边必然会少一个连通块，而一共只有 $O(n\log n)$ 个点，因此复杂度为 $O(n\alpha(n)\times\log n)$。

tree

将没有边的点对之间视作有一条 0 权值的边，则答案不会改变，操作可以简化为随机一条边，将其权值加 1 并对 3 取模。

考虑把期望转化成对所有 $\binom{n}{2}^t$ 种方案的生成树权值和求和。对每棵生成树的贡献分开讨论，容易发现每棵生成树的贡献仅和其边中分别有多少权值为 $0,1,2$ 的边有关，可以先算出所有拥有恰好 i 条 0 边，恰好 j 条 1 边的生成树个数。

对于 p 的求解，考虑矩阵树定理，设 0 边的权值为 x，1 边的权值为 y，2 边的权值为 1，则 $p_{i,j}=[x^iy^j]P(x,y)$，其中 $P(x,y)$ 是矩阵树定理求出的 $\sum_T\prod_{(u,v)\in T}w_{u,v}$。

如果使用暴力二元多项式操作求解行列式，复杂度可能会到达 $O(n^7)$，一般的二元多项式操作也只能做到 $O(n^5\log^2 n)$，考虑对每个 $x=i,y=j$，$O(n^3)$ 地算出 $P(i,j)$ 的值，那么可以对两维依次做拉格朗日插值求出 $P(x,y)$ 的各项系数，这部分的复杂度是 $O(n^5)$。

接下来我们从初始生成树统计最终生成树的权值。对每个 $p_{x,y}$ 记 $f_{i,j}$ 表示有恰好 i 条边在操作过程中 $+1$，有恰好 j 条边在操作过程中 $+2$（均在 $\bmod 3$ 意义下）的所有最终生成树的权值总和，这是可以使用 $O(n^3)$ 的背包解决的。对于所有 (x,y) 对应的 f 乘上 $p_{x,y}$ 后按位累加，最终的 $f_{i,j}$ 即表示对于某 $n-1$ 条边有恰好 i 条边在操作过程中 $+1$，有恰好 j 条边在操作过程中 $+2$，剩余的 $\binom{n-1}{2}$ 条边无限制的每种操作方案的贡献系数，这部分的复杂度是 $O(n^5)$。

考虑计算每个 (i,j) 的操作方案个数，记其关于 t 的 egf 为 $F_{i,j}(t)=G_0^{n-1-i-j}(t)G_1^i(t)G_2^j(t)e^{\binom{n-1}{2}t}$，其中 $G_k(x)=\sum\limits_{i\ge 0}\dfrac{x^i}{i!}[i\bmod 3=k]$。

由单位根反演可求出各 G 的封闭形式：

- $G_0(x):\dfrac{1}{3}(e^{wx}+e^{(-1-w)x}+e^x)$
- $G_1(x):\dfrac{1}{3}((-1-w)e^{wx}+we^{(-1-w)x}+e^x)$
- $G_2(x):\dfrac{1}{3}(we^{wx}+(-1-w)e^{(-1-w)x}+e^x)$

其中 w 为三次单位根，在本题模数下其存在模意义下的值，注意 $-1-w=w^2$。

记最终答案的 egf 为 $F(t)=\sum\limits_{i,j}f_{i,j}F_{i,j}(t)=\sum e^{(aw+b)t}g_{a,b}$，所有 $g_{a,b}$ 可以通过对 $F_{i,j}$ 的封闭形式进行 $O(n^5)$ 背包求出，并且不同的 (a,b) 仅有 $O(n^2)$ 个，也就是单次询问只需要 $O(n^2)$ 次快速幂即可，这可以使用光速幂优化到每次询问 $O(n^2)$。

为了减小预处理常数，std 使用了多层光速幂，总复杂度 $O(n^5+qn^2+n^2mod^{\frac{1}{3}})$。