T1 小 h 学步

原本想出成期望，用点期望的运算性质。但这个东西在 csp 模拟赛是违法的。

最终答案为：

$$ans = \sum_{\text{一种选法}} ((\sum x_i)^2 + (\sum y_i)^2 + (\sum z_i)^2)$$

其中对于同一个 i，x_i, y_i, z_i 互相关联。

将 x 部分的平方拆开得到 $\sum x_i x_j$，由于任意的 p 有 $\sum x_p = 0$，所以：

$\sum_{\text{一种选法}} \sum_{i \neq j} x_i x_j = 0$

y, z 部分同理。

所以 $ans = n^{n-1} \times (\sum x_i^2 + \sum y_i^2 + \sum z_i^2)$。

T2 小球进洞

对于一种挖洞情况，答案为每个点子树内洞的个数乘积。

直接 dp 是 $O(nm^2)$ 的，记录子树内挖了多少洞，经过一条边时枚举有多少洞挖在此边上即可。

发现这个东西关于第二维是不超过子树 $size$ 次的多项式。直接转成点值就能合并，复杂度 n^2。

但这个东西在 csp 模拟赛是违法的。尝试优化之前的 dp 状态。

因为洞多而球少，用不到的洞（子树内没有球进入的洞）是无所谓在哪条边上的，我们可以只记录进球了的洞。

设 $dp_{u,i}$ 表示 u 子树的球进入了 i 个不同的洞中。儿子合并完后，考虑 u 点进入的洞，可能是 $leaf_u + i$ 中的一个，也可能新开一个洞，洞在子树内概率 $\frac{size_u - 1}{n-1}$。

最后用组合数选出这 i 个洞即可。$ans = \sum \binom{m}{i} i! dp_{1,i}$。

T3 快速kmp

设 $f(S)$ 表示只保留字符集 S 下序列的最小正周期的周期数，则 $f(S) = \gcd(f(\{u, v\}))$ 其中 $u, v \in S$。

这个不难观察得到。证明则考虑每次加入一种字符，由于和每种之前字符的周期数都是 $f(S)$ 倍数，则以 $\frac{len}{f(S)}$ 为一段，加入前每段相同则加入后每段依旧相同的，所以 $f(S) \geq \gcd(f(\{u, v\}))$。又显然有 $f(s) \mid f(u, v)$，所以结论成立。

用 kmp 可以算出任意两个元素 x, y 的 $f(\{x, y\})$，当作一条边。时间 $O(nk)$。询问即求两端都在 S 集合中的边的 $\gcd$。

当 $k \leq 20$，可以直接暴力 qk^2。$k \leq 36$ 原本想给折半，但发现 qk^2 飞快。

k 更大时，延续折半思想，我们直接分块，B 个为一块，共 $C = \lceil k/B \rceil$ 块（实测 $B = 8$ 最快），考虑在块内直接对 2^B 种情况来预处理。块内的边直接 $O(2^B B^2)$ 暴力计算。两块块之间的边也可以做到 $O(4^B)$ 处理完，因为我们可以只用 1 次 $\gcd$ 从 $x - lowbit(x)$ 推到 x。注意，这道题中所用的求 $\gcd$ 都是多次求的多个数的 $\gcd$，理论时间应为 $O(\log V + cnt)$，均摊到每次 $\gcd$ 这里直接认为是 $O(1)$。询问时枚举两块直接使用预处理好的信息即可。

时间复杂度 $O(nk + C\cdot 2^B B^2 + C^2\cdot 4^B + C^2 Q)$。

T4 宿舍派对

直接枚举终点，上 lct 维护基环树，时间复杂度 $O(n\log n)$。大常又难写，勇者可以视常数获得 $50\sim 70$。

但这个东西在 csp 模拟赛是违法的。

如果固定终点则图也固定。但显然不能这么直接，可以将终点固定在某区间内，此时区间外的边就固定了。于是可以分块。

当区间外边固定，可以计算出每个点第一次走到区间内的位置与距离。在区间内枚举终点并重建区间内的边，当一条边指向区间外，则用之前维护好的信息走回来即可。时间 $O(n^{1.5})$。

进一步地，可以分治处理，每层跑一个 `DAG` 拓扑排序，对 $[l,mid]$ 中每个点维护出，如果终点在 $[mid+1,r]$，则第一次走到 $[mid+1,r]$ 时的位置与路径信息，然后进入 $[mid+1,r]$ 递归，另一侧同理。类似于分块，建图时一条边指向 l,r 以外了，则用前几层存的信息走回区间内即可，最多跳递归深度次。

通过维护的信息，可以把 $op=1/2$ 的询问直接预处理掉。

对于 $op=3$ 的询问，用预处理时存下来的信息，考虑 x,y 在当前分治区间 mid 的哪侧，如果 x,y 在不同侧，则让 x 走到 y 另一边。

时间复杂度 $O(N\log^2 N + Q\log N)$，可以拿到 70 分。

多出的 $\log$ 是利用前几层信息，让一条指向外的边走回来。我们直接把这个信息也记录下来。记录下 $l\le i\le r$ 的 i 的 L_i 与 R_i 边走回 l,r 区间时的信息，递归时，更新信息就是合并上这一层算出的从一侧到另一侧的信息。空间复杂度 $O(N\log N)$。时间 $O((N+Q)logN)$。