题解

T1 特

首先考虑给定一个序列 a ，如何计算该序列的最长上升子序列。设计 $dp\{i\}$ 表示最长上升子序列末尾为元素 a_i 的最长长度，则有 $\forall 1 \le j < i$ ，若 $a_j < a_i$ ，则有 $dp\{i\} \leftarrow dp\{j\} + 1$ 。可以发现这个转移的过程可以使用数据结构对值域维护以进行优化。

接下来考虑两个序列 a 和 b ，实际上类似的考虑设计 $f\{i\}$ 表示最长上升子序列末尾为 a_i 的最长长度，$g\{i\}$ 表示最长上升子序列末尾为 b_i 的最长长度，则转移是类似的。时间复杂度 $O(n \log v)$ ，可以通过离散化做到 $O(n \log n)$ 。

T2 立

考虑将从点 s 到点 t 再回到点 s 的路径进行拆解，并分析其形态，经过分析，可以发现将点 s 到点 t 的路径作为主体路径之后，从点 t 到点 s 的路径可以被简化为，再主体路径上反复横跳，且之后跳跃到的点必然在之前停留住的点之前。

基于此进行分析，不妨考虑设 $dp\{x,y\}$ 表示此时从路径 $x \to t$ 和 $y \leftarrow t$ ，则首先可以将路径扩展一个点得到 $dp\{z,y\}$ 或 $dp\{x,z\}$ ，随后注意到路径可能会有重合部分，对于重合部分，考虑使用类似点 x 和点 y 交换技巧，即预处理出点 y 到点 x 的最短路径，随后便可以得到路径 $y \to t$ 和 $x \leftarrow t$ ，因此预处理维护转移系数，直接使用堆维护 $dp\{x,y\}$ 的转移可以做到 $O(\sum n^3 \log n)$ ，但注意到 $dp\{x,y\}$ 的值域仅有 $O(n)$ ，因此将堆换为维护值域的队列即可做到时间复杂度 $O(\sum n^3)$ 。

T3 独

首先注意到树上一个连通块存在唯一的深度最小的节点，否则无法构成一个连通块，因此对于一次查询点 x 所在连通块的大小，不妨从点 x 开始一路跳与点 x 颜色相同的父亲节点，直到跳到根节点或深度最浅的节点 y 满足节点 y 与其父节点颜色不同，则点 x 与点 y 在同一个连通块。

定义一个点 y 满足点 y 与其父节点颜色不同时的点 y 为标志点。接下来，注意到一次修改操作将可能会新生成一个标志点，并将子树内的所有标志点全部删除。因此，查询点 x 所在连通块大小时我们直接跳转到点 x 祖先中首个标志点并查询。对于每次修改点 x 子树，对于子树部分，使用 平衡树/set 维护 dfs 序以快速找到子树内的所有标志点并删除；对于祖先部分，注意到我们会影响到连通块大小的标志点仅有点 x 本身（如果点 x 是标志点）和点 x 父节点祖先的首个标志点，分别计算影响前后的贡献变化量即可。

注意到时间复杂度瓶颈在于寻找一个点祖先处的首个标志点，使用线段树区间覆盖即可，时间复杂度 $O(n \log n)$ 。

T4 行

首先考虑 $m=1$ 时如何处理，当 $m=1$ 时，我们会将这 n 个元素进行任意分组，贡献为每组权值和的乘积，考虑使用组合意义拆解乘积，则这等价于对于每个 k ，我们在这 n 个元素中选出 k 个元素乘积起来，然后考虑这 k 个元素在互不相同的组内的贡献，可以发现这相当于把其他的 $n-k$ 个元素加入到这 k 个元素代表的组内，贡献系数即为 k^{n-k} 。

接下来考虑 $m>1$ 时的情况，可以根据类似的组合意义发现，这等价于对于每个 k ，我们在这 nm 个元素中选出 k 个元素乘积起来，随后将这 k 个元素强制互不相同的组，并将其他的 $n-k$ 个元素加入到这 k 个元素代表的组中，且要求加入之后每个组的元素数量均为 m 的倍数。注意到钦定这 k 个元素后，对于每个组固定组内元素数量之后，方案数等价于可重集组合，因此对该部分设计状态 $dp\{i,j\}$ 表示此时新建了 i 个组且已经使用了 jm 个元素的方案数，转移即新增加一个组且枚举组内的元素为 km ，则有 $dp\{i,j\}\to dp\{i+1,j+k\}$ ，预处理该信息后即可做到 $O(n^3)$ 。

考虑优化，将组内元素数量 $\le\sqrt{n}$ 的组和数量 $>\sqrt{n}$ 的组分开处理，前者元素数量 $\le\sqrt{n}$ ，后者总体组数 $\le\sqrt{n}$ ，然后将两边的动态规划的结果合并即可，时间复杂度 $O(n^{2.5})$ 。