树

1 前置技能

树同构计数

2 算法

首先把问题转化成对输入的 n 个集合和点的合法对应关系进行计数（这里认为集合有编号），再除以每种集合出现次数的阶乘之积即可。

考虑建立一张左右各有 n 个点的二分图 G，其中左侧的每个点表示每个集合，右侧即所求的结点 $1, 2, \cdots, n$，左侧第 i 个点和右侧第 j 个点连边当且仅当第 i 个集合包含 j。

据观察，若将原树黑白染色，可以发现在 G 中，右侧的黑点通过左侧的点连接成了一棵树（虽然我们目前并不知道左侧点的标号）。更进一步地，如果存在合法的树，这张二分图由两个连通块构成，其中每个都是一棵树：第一棵树 T_1 包含所有黑点（右侧）和所有白点的对应集合（左侧），而第二棵树 T_2 包含所有白点（右侧）和所有黑点的对应集合（左侧）。

现在的问题变成需要找到两个一一映射 f 和 g，其中 f 将 T_2 中白点映射到 T_1 中白点的对应集合，g 将 T_1 中黑点映射到 T_2 中黑点的对应集合，且要满足 $(x, f(y))$ 在 T_1 中的是否有边和 $(g(x), y)$ 在 T_2 中是否有边完全一致（因为确定编号后 T_1 和 T_2 两棵树和原树都完全相同）。

此时可以发现，将 g 和 f^{-1} 合并即是 T_1 到 T_2 的同构，即黑白染色后原树的自同构。

令重心为根，则可将问题转化为有标号有根树的同构计数，可以通过哈希或离散化在 $O(n)$ 或 $O(n * log_2 n)$ 时间解决。

3 实现细节

求得重心后需要检查 T_1 和 T_2 的重心是否处于 G 的不同侧。

可能存在两个重心，但此时不需要将答案 $\times 2$ 因为要求黑白染色的结果不能改变。

2.茫茫人海

设 $f(s_1, s_2)$ 为 0 或 1 表示这两个串是否是一个合法解。考虑建立一个 DFA，每个点表示一些两个长度相等的字符串对的集合 (s_1, s_2)。具体来说，如果对任意长度相等的字符串对 (t_1, t_2)，$f(s_1 t_1, s_2 t_2) = f(s_1' t_1, s_2' t_2)$，那么我们认为 $(s_1, s_2) \sim (s_1', s_2')$，即这两个字符串对在 DFA 的同一个点上。

构建 DFA 可以从初始两个空串出发 dfs，每次枚举两个字符 c_1, c_2，考虑 $(s_1 c_1, s_2 c_2)$ 所在的 DFA 点。

具体来说，对于一组 (s_1, s_2)，如果最前面的字符匹配方式固定，则可以删掉前面的部分，精简成删去之后的字符串对。需要一些比较复杂的讨论，最后的 DFA 会包含约 3000 多个点，转移边约 20000，直接在 DFA 上 dp 即可。

图

1 前置技能

最短路优化建图

2 算法

原问题即为对于每个点 u，令 1 为源点 u 为汇点跑流量为 2 的最小费用流。

考虑模拟以上过程。先模拟第一次流，以 1 为源点求出到每个点的最短路径并得到最短路树。那么对于每个点 u，只需将 1 到 u 的路径上的边反向，费用变为相反数，再跑一次最短路即可。

首先需要利用 $\mathrm{Johnson}$ 算法解决负边的问题，即记 $dist[u]$ 表示从 1 到 u 的最短路径，对于边 $u \to v$，若原权值为 w，令 $w \leftarrow w + dist[u] - dist[v]$，此时一定有 $w \geq 0$ (否则不符合最短路径的定义)。注意变换后到 1 到 u 的最短路为变换前最短路 $+dist[u]$。

在本题中，观察到变换后最短路树上的边权全部变为了 0。接下来建立张 n 个点的新图 G。初始 G 为空。对于原图中的每条变换后的非树边 $u \to v$ 权值为 w，考虑最短路树上 u 到 v 的路径上的点集 S (这里认为最短路树是无向的），对于任意 $x \in S$，在 G 中从 x 向 v 连一条权值为 w 的边。最终在 G 上跑以 1 为源点的最短路即可得到每个点第二次流的结果。

证明考虑固定一个点 u，设第二次流时 1 到 u 的路径中经过的非树边依次为 $(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k)$。考虑从 1 走到 u_1 以及从 v_i 走到 u_{i+1} 的过程，分类讨论可以发现最优情况下 v_i 一定在 u_{i+1} 和 v_{i+1} 的路径之间，故 G 中包含了所有有用的边。

只需要用倍增或树剖等方式优化建图即可。

3 实现细节

注意优化建图时虚点之间的边大部分边权是 0，故不用把虚点放进优先队列中更新，只需标记是否访问过即可，这样只会有 $O(n+m)$ 个点被扔进优先队列中，复杂度 $O((n+m)log_2(n+m))$。