# 题解

## T1. 扫雷

按题意模拟即可。

## T2. 五彩树

如果以 $u$ 为根，不难证明存在一种最优情况满足 $S$ 中的节点是以某个节点为根的一棵完整的子树。

如果以 $1$ 为根，$S$ 中的节点就有三种情况：

1. 整棵树。
2. 以某个点（不为节点 $1$）为根的一棵完整的子树。
3. 整棵树除去以某个点（不为节点 $1$）为根的一棵完整的子树后剩下的部分。

第一种情况不难判断，对于后两种情况，不妨假设上文中的那个节点为 $x$，则有：

在第二种情况中，$u$ 的最佳位置是整棵树除去以 $x$ 为根的子树后，距离 $\mathrm{fa}[x]$ 最远的节点。

在第三种情况中，$u$ 的最佳位置是只考虑以 $x$ 为根的子树时，距离 $x$ 最远的节点。

这两个值，可以直接 DFS 求出后者，然后再换根 DP 求出前者。

接下来考虑如何判断以某个点为根的子树，或除去以某个点为根的子树后剩下的部分中是否存在至少 $k$ 种颜色。

考虑以 $1$ 为根时的 DFS 序，显然这两者对应的都是其中一个区间（这里把前缀 + 后缀也称作一个区间）。

所以把 DFS 序复制一份接在后面，然后使用双指针即可维护固定右端点时，最右的满足区间中至少有 $k$ 种颜色的左端点的位置，将这个位置与区间的左端点比较即可判断某个点的子树中是否存在至少 $k$ 种颜色，然后再将答案与之前求出的子树中 / 子树外的最大距离取较大值即可。

时间复杂度为 $\mathcal{O}(m+n)$。

## T3. 大洗牌

意即求一置换 $p$ 满足 $p^k=a^{-1}$。如果可行，求出不同的 $p$ 的个数。

考虑将 $a^{-1}$ 拆成循环的不相交乘积，令长度为 $i$ 的循环的个数为 $b[i]$。

考虑一个长度为 $x$ 的循环的 $k$ 次方，令 $g=\gcd(x,k)$，则会变为 $g$ 个长度为 $\dfrac{x}{g}$ 的不相交循环。

此时已知 $i=\dfrac{x}{g}$ 和 $k$ 的值，但不知道 $x$ 和 $g$ 的值。

因为 $x=g\cdot i$，所以考虑求出 $g$ 的值。

考虑关系 $\dfrac{k}{g}\perp\dfrac{x}{g}$，即 $\dfrac{k}{g}\perp i$。

比较同一质数的次幂不难发现可能的 $g$ 满足：$\gcd(i^{\infty},k)\mid g\mid k$。

即 $g$ 为 $k$ 的因数，但为 $\gcd(i^{\infty},k)$ 的倍数。这里 $\gcd(i^{\infty},k)$ 可以理解为 $\lim\limits_{n\to\infty}\gcd\left(i^{\lfloor n\rfloor},k\right)$。

换句话说，若 $d \,\Big|\, \dfrac{k}{\gcd(i^\infty, k)}$，则 $g$ 可以为 $d \cdot \gcd(i^\infty, k)$。

为了方便，令 $h = \gcd(i^\infty, k)$。

若只要求出一组方案，显然最优策略是让 $g = h$，因为可能的 $g$ 都是 $h$ 的倍数，这样不会产生任何浪费。

则只要对于任意的 $i$ 均有 $h \mid b[i]$ 就不难构造出一组方案，否则无解。

时间复杂度为 $\mathcal{O}(n + \sqrt{n} \log k)$，其中的 $\sqrt{n}$ 是因为 $b[i]$ 只有 $\mathcal{O}(\sqrt{n})$ 个位置非零。可以获得 $70$ 分。

若要求出方案数，因为每个 $i$ 都是独立的，只要求出每个 $i$ 的方案数，再将它们相乘即可。

对于某个 $i$，即是将 $b[i]$ 个循环分成若干个集合的不交并，其中每个集合的大小都是 $h$ 的倍数但是 $k$ 的因数。

对于一个大小为 $x$ 的集合，又有 $\dfrac{i^x x!}{ix} = i^{x-1} \cdot (x-1)!$ 种还原成一个循环的方法。

考虑它的组合意义，可以发现就是求无序组合方案数。

考虑对于每个合法的 $g$ 的 EGF，即 $\displaystyle\sum_{\substack{1 \le g \le b[i] \\ h|g|k}} \frac{i^{g-1} x^g}{g}$，做多项式 Exp，求得的即是答案的生成函数。

即 $\displaystyle \mathrm{Ans} = b[i]! \cdot [x^{b[i]}] \exp\left(\sum_{\substack{1 \le g \le b[i] \\ h|g|k}} \frac{i^{g-1} x^g}{g}\right)$。

因为 $\sum b[i] = n$，所以时间复杂度为 $\mathcal{O}(n \log n + \sqrt{n} \log k)$。