

2024 钉耙编程中国大学生算法设计超级联赛 3 题解

2024 年 7 月 26 日

1 深度自同构

先考虑 n 个点的合法的树的个数，容易发现根据要求每个节点的所有子树的形态必定完全相同。因此可以递推，令 f_i 表示 i 个点的合法的树的个数，枚举根的儿子个数，有 $f_i = \sum_{d|(i-1)} f_d$ 。这一转移可以用枚举倍数的方法加速，复杂度为调和级数。

再考虑合法的森林个数，注意到森林中每棵树必定完全相同，因此 $ans_i = \sum_{d|i} f_d$ ，再用调和级数的复杂度算一遍即可，复杂度 $O(n \log n)$ 。

2 旅行

本题要求树上一些不相交的链使得权值和最大，每个链要求起始点和终止点颜色相同，权值为起点和终点的权值和。

考虑做树上 dp，可以在 LCA 处枚举经过这个点的链，如果这个点不被任何链经过，那么把所有孩子节点的 dp 值加起来，否则需要找一条经过 LCA 的链，假设为 (u,v) ，那么需要维护起来的权值有 uv 下面的子树值和这条链上其他点的其他孩子的 dp 值。

用线段树合并可以解决，线段树的下标为颜色，把链上一串 dp 值维护起来，这样在合并的时候可以顺带 dp 出最优方案。

3 游走

分类讨论，难点主要在检查是否冲突。

如果已知的信息 $p_i \neq 1$ 时我们就可以确定酒鬼移动了。此时首先要检查所有这样的信息 $p_i + q_i$ 的奇偶性一致，并且相邻时刻的信息对应的移动距离不能超过他们的时间差，这可以通过 set 维护 pair 支持插入。注意最早的信息关于起点的距离也有限制。

对于 $p_i = 1$ 的情况，不用保证 $p_i + q_i$ 奇偶性一样，但如果奇偶性不一样，则要保证时刻小于最晚移动时间。

总复杂度 $O(n \log n)$ 。

4 游戏

首先注意到最终这个答案只与任意两个人之间的差值有关，于是我们对于每个 k 可以先求出 $a_i - a_j = k (1 \leq i < j \leq n)$ 的数量。这个显然可以用 FFT 做到 $O(a_i \log a_i)$ 。

接下来考虑在两个值初始差值为 k 的情况下， t 轮后相同的概率。

构造多项式 $F(x) = (p_1x + p_2/x + p_3)$ ，其中 p_1 代表两个数差值 +1 的概率， p_2 代表两个数差值 -1 的概率， p_3 代表两个数差值不变的概率，那么 $F(x)^t$ 中 x^k 的系数就是 t 轮后两个值相同的概率。

p_1, p_2, p_3 易求得，难点在于 $F(x)^t$ 。

将 $F(x)$ 整体乘 x ， $F(x) = (p_1x^2 + p_2 + p_3x)$

令 $G(x) = F(x)^t$

$G'(x) = t * F'(x) * F(x)^{t-1}$

$G'(x) * F(x) = t * F'(x) * F(x)^t$

$G'(x) * (p_1x^2 + p_2 + p_3x) = t * (2 * p_1x + p_3) * G(x)$

于是发现 $G(x)$ 和 $G'(x)$ 之间存在递推关系，即我们可以根据 $G(x)$ 的前几项快速求出后面的项。

具体计算时要注意特殊处理 $G(x)$ 的前几项，求逆元的时候需要线性求出 t 以内所有数的逆元。

总时间复杂度 $O(T + a_i \log a_i)$

5 数论

考虑集合加入数字 gcd 如果变化至少减半，所以数列所有区间的 gcd 本质不同的个数只有 $O(n \log n)$ 个。

考虑对答案做简单容斥，计算有多少个合法方案不包含 i 。只需计算出前缀 $1 \cdots i-1$ 中选出若干区间符合条件的方案数、后缀 $i+1 \cdots n$ 中选出若干区间符合条件的方案数。

考虑对每种 gcd 分别求解，以前缀为例，考虑所有右端点在 $i-1$ 、 gcd 为当前考虑的 gcd 的区间，其左端点连续，构成了某个区间 $[l, r]$ 。这样计数的 dp 就可以用线段树来优化转移，复杂度和此时的 (l, r) 数对个数相关，由前述结论总复杂度均摊是 $O(n \log^2 a)$ 的。

6 字符串

首先本质不同是个没用的条件。我们考虑观察所有 occ 相同的子串，固定子串的右端点的位置 r ，左端点的位置是一段区间 $[l_x, r_x]$ ，那么随着 r 的增加， l_x 和 r_x 都单调不降。

那么考虑询问 1：我们只需要取出右端点 $r_p \geq r$ 的一些区间 $[l_x, r_x]$ ，将 $r_x \leq l$ 和 $l_x \leq l < r_x$ 分别统计贡献即可。具体的，由于根据右端点 r_p 排序之后， $r_x \leq l$ 和 $l_x \leq l < r_x$ 这两类都对应一段连续的区间，我们只需要维护 $r_x - l_x + 1, l_x$ 的前缀和即可。询问 2 类似。代码参考 std 中的 1006-2。

另外一个做法是使用基本子串结构，这里不再展开，可以参考 std 中的 1006-1。

7 单峰数列

维护差分数组，更容易判断题目要求的条件。

线段树维护区间是否是完全相同、严格递增、是否严格递减、是否单峰。

复杂度 $O((n + q) \log n)$ 。

8 比特跳跃

首先跳多次的代价肯定超过跳一次的代价。然后考虑跳跃到 i 如果不是从 i 子集过来，那一定不如直接从 1 跳过来。

因此原图基础上加上从 1 到其他点的跳跃边，跑一遍单源最短路。

然后每个点 i 求出子集 j 的 dis_j 的最小值加上 ki 尝试更新 dis_i 。

然后在此基础上再做一遍单源最短路即可。

总复杂度 $O((n + m) \log n)$ 。

9 圣芙蕾雅

首先求出每两位女武神是否可以分配同一套装甲，即判断是否存在一个事件（点）上，这俩女武神都已经分配了装甲，且后续都有可能参加战斗。

枚举每个女武神 x ，从每个对应战斗事件开始向前遍历，遇到对应分配事件停止。途中遇到的每个非对应分配事件，可得到一条冲突关系：女武神

x 与此分配事件对应的女武神 y 不能分配同一套装甲, 因为此时女武神 x 已分配装甲且后续有可能参加战斗, 而女武神 y 此时要被分配装甲。

令 D_x 表示女武神 x 对应的分配事件。

性质 1: 如果在事件 u 结束后, 女武神 x 依然可能参加战斗 (即从点 u 可以到达 x 对应的战斗事件节点, 下同), 则 D_x 支配 u , 否则不满足题意。

性质 2: 如果在事件 u 结束后, 女武神 x, y 都依然可能参加战斗, 则 D_x 支配 D_y 或 D_y 支配 D_x 。

证明: 根据性质 1, 有 D_x 支配 u 且 D_y 支配 u , 而支配关系为树形结构, 因此得到 D_x 支配 D_y 或 D_y 支配 D_x 。

性质 3: 如果女武神 x, y 不能分配同一套装甲, 且 D_x 支配 D_y , 则在 D_y 处, 女武神 x 依然有可能参加战斗。

证明: 如果 D_y 处女武神 x 已经一定不会参加战斗了, 那么可以和 y 分配同一套装甲。

性质 4: 如果女武神 x, y 不能分配同一套装甲, 女武神 y, z 不能分配同一套装甲, 但是女武神 x, z 可以分配同一套装甲, 且 D_x 支配 D_y , 则 D_y 支配 D_z 。

证明: 由于 y, z 不能分配同一套装甲, 因此 D_y 支配 D_z 或 D_z 支配 D_y 。假设 D_z 支配 D_y , 则 z 在 D_y 处依然可能参加战斗, 同理, 由于 x 在 D_y 处依然可能参加战斗, 则 x 与 z 不能分配同一套装甲, 与前提矛盾, 因此 D_y 支配 D_z 。

结论: 女武神间根据不能分配同一套装甲的关系建立无向图, 此图一定是弦图。

证明: 假设不是弦图, 则一定存在长度大于等于 4 的无弦环, 不妨设其为 $x_1, x_2, \dots, x_{n-1}, x_n$, 其中 $n \geq 4$ 。由于 x_1, x_2 不能分配同一套装甲, 不失一般性, 假设 D_{x_1} 支配 D_{x_2} , 反复应用性质 4, 可以得到 D_{x_1} 支配 D_{x_2} , D_{x_2} 支配 D_{x_3} , \dots , $D_{x_{n-1}}$ 支配 D_{x_n} 。由于 x_1, x_n 不能分配同一套装甲, 因此存在一个事件 u , 在 u 结束时这两女武神都依然有可能参加战斗。根据性质 1 得 D_{x_n} 支配 u , 则从 $D_i (1 \leq i \leq n)$ 都可以到达 u , 因此 x_1 在这些事件处都依然有可能参加战斗, 即 x_1 与 $x_i (2 \leq i \leq n)$ 都不能分配同一套装甲, 与前提矛盾。

因此后续问题即为在弦图上求最小染色数, 用完美消除序列求解即可。

事实上也可以不使用完美消除序列的构造算法, 对原图求支配树, 可以根据支配树的后序遍历来得到完美消除序列。时间复杂度 $O(k(n+m))$ 。

10 绘世之卷

10.1 做法一

首先考虑仅有增加颜色操作的情况。若新增颜色 x ，则需要考虑 x 作为被除数产生的答案和作为除数产生的答案。

当 x 作为被除数时，令 q 为商，则只要查询小于等于 $\frac{x}{q}$ 的最大值，就可以得到商为 q 时的最优情况，由于商只有 $O(\sqrt{x})$ 种，因此查询次数为 $O(\sqrt{n})$ 。

当 x 作为除数时，令 q 为商，则只要查询大于等于 xq 的最小值，就可以得到商为 q 时的最有情况，如果 $x \geq \sqrt{n}$ ，此时 $0 \leq q \leq \sqrt{n}$ ，查询次数为 $O(\sqrt{n})$ ；如果 $x < \sqrt{n}$ ，则余数必然小于 \sqrt{n} ，令 l 为最小存在的商，则当 $q \geq l + \sqrt{n}$ 时，一定不可能得到更有情况，因此查询次数依然是 $O(\sqrt{n})$ 。

查询总次数为 $O(q\sqrt{n})$ ，插入总次数为 $O(q)$ ，因此希望有一种单次查询时间复杂度 $O(1)$ ，单词修改时间复杂度 $O(\sqrt{n})$ 的方法。考虑分块，按照 \sqrt{n} 分块，为每个块维护一个懒标记即可。

再考虑有删除的情况，考虑线段树分治，预处理求出每种颜色存在的时间戳区间，放入以时间戳为下标的线段树中。遍历线段树，同时维护分块数据结构。可以用记录所有修改的方式来对数据结构进行回退操作。

总时间复杂度 $O(q \log n \sqrt{n})$ ，但是这种做法较为复杂，常数较大。

10.2 做法 2 (std 做法)

发现性质当前颜色数大于 \sqrt{n} 时，答案一定为 $O(\sqrt{n})$ 。证明：如果存在两个不超过 \sqrt{n} 的颜色，则这两个颜色就可以得到 $O(\sqrt{n})$ 的结果；否则，一定存在两个数的差不超过 \sqrt{n} ，此时这两个颜色一定可以得到 $O(\sqrt{n})$ 的结果。

因此可以根据当前颜色数来进行分类讨论，如果当前颜色数不超过 $O(\sqrt{n})$ ，则暴力即可；否则答案一定为 $O(\sqrt{n})$ ，枚举余数，余数小于当前最优解，寻找最小的商。此时仍然需要维护后继，考虑加入 x ，只要维护 $x - \sqrt{n}$ 到 x 的后继，其余处无意义。

因此可以省去数论分块与分块维护前驱后继的过程，大幅度减小常数。

时间复杂度仍然为 $O(q \log n \sqrt{n})$ 。

11 抓拍

首先周长 $= 2 * (\max x - \min x) + 2 * (\max y - \min y)$ 。

由于所有人速度都相同，考虑 $\max x - \min x$ 的变化速度只有可能是 $-2, -1, 0, 1, 2$ 而且一定是按照这个顺序（某一部分有可能不存在），因此 $\max x - \min x$ 关于时间的函数是凸的。对 $\max y - \min y$ 同理。

三分找极小值点即可。复杂度 $O(n \log n)$ 。

12 死亡之组

分类讨论，首先把 a_1 从集合中去掉：

如果 $a_1 \geq L$ ，那么选最小的三个。

如果 $a_1 < L$ ，那么选最大的，和最小的两个。

如果上述方案依然符合死亡之组的条件那么无解，否则有解。