「NOIP 2022 模拟赛」解题报告

xcyle

树上划分（plan）

算法一（20pts）

我会暴力！

时间复杂度 $O(2^n \times n \times a)$。

算法二（50pts）

我会树形dp！

设 $f_{x,cnt,now}$ 表示在 x 子树内断了 cnt 条边，最上面的连通块的 $\gcd$ 是 now。特别的，$now = 0$ 表示 x 的父边被割。时间复杂度 $O(n^2a^2)$。

算法三（80pts）

我会优化状态！

注意到有效状态 $now = 0$ 或 $now \mid a_x$。时间复杂度 $O(n^2d(a)^2)$。当树退化为链时为 $O(n^2d(a))$。

但如果你再加上只搜有用状态和快速 gcd ，出题人就根本卡不掉了。

算法四（100pts）

我会设计更好的状态！

把算法二中的状态 now 从“在子树中连通的 $\gcd$ ”改成“最终所在的连通块的 $\gcd$”。

也就是说，这个 $\gcd$ 会包含子树外的部分，并且在还未考虑子树外的断边方案时被提前钦定。

这样我们转移的时候就可以少枚举一个值了。

时间复杂度 $O(n^2d(a))$。

路径查询（glyph）

算法一（25pts）

我会暴力！我会预处理！

算法二（50pts）

我会做特殊性质！

这个性质可以看成：询问的起点 $s(x_1, y_1)$ 都在主对角线的下面，询问的终点 $t(x_2, y_2)$ 都在主对角线的上面。

对于最优路径与主对角线的交集中的任意一个点 x，必有 $dis(s,t) = dis(s,x) + dis(x,t)$。

又因为最优路径会经过主对角线上的至少一个点，$dis(s,t) = \min\limits_{x\text{在主对角线上}} dis(s,x) + dis(x,t)$。

因此我们可以预处理主对角线上的每个点到其它所有点的 dis。每次询问用 $O(n)$ 的时间复杂度求答案。

时间复杂度 $O(n^3 + qn)$。

算法三（80pts）

我会数据结构！

很多数据结构都可以硬套，但复杂度不一定非常优秀。

此处举一例，分块。设 B 为块长，每隔 B 个位置取一条对角线，做算法三。

时间复杂度 $O(\frac{n^4}{B} + qB^2)$。平衡后 $O(q^{\frac{1}{3}} n^{\frac{8}{3}})$。

算法四（100pts）

我会分治！

特殊性质的提示性其实很强。我们考虑一个分治的过程，每次将 $n \times n$ 的网格分成四个 $\frac{n}{2} \times \frac{n}{2}$ 的子网格。

如果一个询问的起点和终点属于同一个子网格，递归下去做子问题。否则，我们可以套用特殊性质的做法：具体来说，对于大小为 $t \times t$ 的网格，这部分的处理是 $O(t^3 + q_{here} t)$ 的，其中 q_{here} 表示这样的询问数量。

对于一个询问，最多递归 $\log n$ 层，在某一层被解决时耗费 $O(n)$ 的复杂度。剩下的就是预处理 dis 的复杂度，简单算一下 $T(n) = 4T(\frac{n}{2}) + n^3 = O(n^3)$。

时间复杂度 $O(n^3 + qn)$。

构造排列（constr）

算法一（33pts）

我会简单构造！

选出所有前 5 个数是奇数，后 5 个数是偶数的排列。

显然满足题意。

算法二

我会证明上界！

对于排列 A，若它可以通过若干次如下操作变成 B，称它们等价：

- 选择 $1 \le i \le \frac{n}{2}$，交换 A 中 $2i-1$ 和 $2i$ 的位置。

所有排列被划分为 $\frac{n!}{2^{\frac{n}{2}}} = 113400$ 个等价类。容易发现每个等价类中最多选一个。

算法三（39pts）

我会写checker！

出题人没想到好的实现方式，就直接把所有东西插进字典树暴力判定，貌似挺优秀的。

考虑暴力枚举所有排列，贪心插入并检查合法性。按字典序跑出来 113214 个。

加入随机调整或者优化贪心策略可能可以得到更高的分数。

另：如果有人有更好的 checker 实现方式或者因 checker 问题失分请联系出题人。

算法四（100pts）

我会构造！

设 S_i 表示一个集合，其元素是长度为 i 的排列。我们期望让 S_i 作为 $n=i$ 的答案。

初态 $S_2 = \{[1,2]\}$。接下来我们考虑通过 S_{2i-2} 构造出 S_{2i}。

拿出 S_{2i-2} 中的每一个排列，选定两个位置插入 $2i-1, 2i$ 。若只考虑 $2i-2, 2i-1, 2i$ 的相对位置，容易发现两种插入方式中恰好有一种满足它们和升序是循环同构的，即 $2i-2, 2i-1, 2i$ 或 $2i-1, 2i, 2i-2$ 或 $2i, 2i-2, 2i-1$。

具体来说，若选择的两个位置都在 $2i-2$ 前面，顺序为 $2i-1, 2i$；若选择的两个位置在 $2i-2$ 一前一后，顺序为 $2i, 2i-1$；若选择的两个位置都在 $2i-2$ 后面，顺序为 $2i-1, 2i$。

因此，用 S_{2i-2} 中的一个排列，可以生成 $\binom{2i}{2}$ 个排列，将这些东西放进 S_{2i} 中，是可以取到上界的。

容易发现这样构造出来的集合满足题意。可以使用反证法，如果集合中的两个排列 A, B 几乎相同，找到最小的元素 $i \le \frac{n}{2}$ 使得两个排列中 $2i-1$ 和 $2i$ 的位置被交换了，可以推出两排列中 $2i-2$ 的位置不同，得证。

黑箱博弈（oracle）

算法一（20pts）

我会暴力！

暴力枚举询问的区间集合，判句可以考虑枚举 $x=1\dots n$，每个询问的作用就是把一些不合法的位置删掉，最后除了 x 的其它位置都需要被删完。

时间复杂度 $O(nm2^m)$。

算法二（40pts）

我会观察！

我们维护一个将 $1\dots n$ 这些位置分段的过程，一开始所有位置都属于同一段。如果一个询问集合 $S=\{I_1\dots I_m\}$ 是合法的，当且仅当它可以通过重复下述过程使得任何两个位置都不属于同一段：

- 选择区间 $I\in S$ 不满足所有点属于同一段或者满足所有点是某一段的前缀或后缀：
  - 若不满足所有点属于同一段，则可以在两个区间端点处继续分段。
  - 如果是某一段的前缀或后缀，则可以把这一段分成两段。

容易证明这是充要的。

注意到每次最多多分两段，因此答案不为 0 的情况下 m 的下限是 $\frac{n}{2}$。

算法三（80pts）

我会容斥和朴素dp！

考虑将算法二的过程改成容斥，我们dp的时候直接钦定最后的那些段是什么，这样我们如果把一段看成一个整体的话仍然是子问题，而段内部的区间可以任取。

设 f_i 为 $n=i$ 时的生成函数，我们有：

$$(1+x)^{\frac{n(n+1)}{2}}=\sum_{i=1}^{n}f_i[y^n](\sum_{j=1}^{\infty}(1+x)^{\frac{(j-1)(j-2)}{2}}y^j)^i$$

可以暴力做多项式操作，时间复杂度 $O(n^7)$。实际上 n 都是开满的，所以自信打表理论上也能过。

算法四（100pts）

我会插值！

求 f_n 这个 n^2 次多项式不如先求出来 n^2 个点值最后在插回去。注意到这样的好处是可以避免多项式乘法。

求单个点值是 $O(n^3)$ 的，因此求值总共是 $O(n^5)$ 的。

而插值的部分因为我们只要求 m 次的系数，因此可以预处理插值那个式子的前后缀快速合并。这部分是 $O(n^4)$ 的。

总时间复杂度 $O(n^5)$，空间大概是 $(\frac{n}{2})^4$。常数比较小，std 只跑了 1.2s。