NPIO Round 6 Sol

前言

“概要”部分概括了该题目的分档，以及每档分的主要做法。可以根据自身情况，选择不懂部分阅读。

数的拆分(number)

原题：[P8778 [蓝桥杯 2022 省 A] 数的拆分]

概要

本题主要分为三个部分：

- 子任务 1~3：暴力枚举。直接枚举可能的 x_1, x_2, y_1, y_2 并检查是否合法。
- 子任务 4,5,8：预处理所有答案。一档使用调和级数的枚举；一档根据性质，使用线性筛 / 埃氏筛；一档搜索出所有可能的答案，做到 $\mathcal{O}(T + V \ln V)$ 或 $\mathcal{O}(T + V)$ 或 $\mathcal{O}((T + \sqrt{V}) \log V)$。
- 子任务 6,7,9：正解。不断使用若干数乘积 $\le a_i$ 的性质减少枚举质因子的开销，同时可以预处理质数使复杂度再除 $\ln V$。

Part 1

子任务 1

可以手玩得到，$1 \sim 5$ 中只有 4 能分解成 $1^2 \times 2^2$ 的形式。

子任务 2

先枚举 x_1, x_2，再枚举 y_1, y_2。由于 $x_1^{y_1}$ 要 $\le a_i$，所以 y 是 $\log V$ 级别的，复杂度为 $\mathcal{O}(TV^2 \log^2 V)$，实际上远远不满。

子任务 3

先枚举 y_1, y_2，再枚举 x_1, x_2，枚举时保证 $x_1^{y_1}$ 和 $x_2^{y_2}$ 都小于等于 a_i，这样枚举次数为 $(\sqrt{V} + \sqrt[3]{V} + \sqrt[4]{V} + \ldots)^2 = \mathcal{O}(V)$，总复杂度 $\mathcal{O}(TV)$。

Part 2

子任务 4

考虑 $x_1^{y_1} \times x_2^{y_2} \le a_i$，所以可以先对 $1, \ldots, V$ 的每个数预处理出其是否为 $x^y, (x, y \in \mathbb{N}_+, y \ge 2)$ 的形式。然后对每个满足上述条件的数 i，枚举同样满足条件的数 j，标记 $i \times j$ 合法，由于 $i \times j \le V$，所以枚举次数 $\sum\limits_{i=1}^{V} \frac{V}{i} = \mathcal{O}(V \ln V)$。总复杂度 $\mathcal{O}(T + V \ln V)$。（由测试点 $17 \sim 19$ 所证明的结论可知，合法的数只有 $\mathcal{O}(\sqrt{V})$ 个，所以实际上枚举次数较少，可以直接通过测试点 $10 \sim 11$，运用 `bitset` 优化存储空间后甚至能在 ≈ 2 秒的时间内通过测试点 $12 \sim 14$）

子任务 5

考察一个数可以分解的充要条件：可以发现，如果 $v=\prod p_i^{k_i}$，存在一个质因子 p，其指数为 1，那么这个质因子不能分到 $x_1^{y_1}$ 或 $x_2^{y_2}$ 的任意一部分，这个数无法被分解。实际上，如果 $\forall p_i$, $k_i=0\lor k_i\ge 2$，数 v 就是一定可以分解的。钦定 $y_1=2,y_2=3$，那么对于任意 $k\ge 2$，k 都可以被拆成若干个 2 和 3 相加的和。所以**数 v 可以被分解的充要条件**就是 $v=a^2b^3,(a,b\in\mathbb{N}_+)$ 。

那么可以通过线性筛，对每个数预处理其质因子 p_i 中最小的 k_i，如果 $=1$ 那么这个数就无法被分解。时间复杂度 $\mathcal{O}(T+V)$。

子任务 8

通过打表，可以猜测可以被分解的数较少。实际上，这种数的数量级是 $\mathcal{O}(\sqrt{V})$，以下是证明：

$$
\begin{aligned}
&\sum_{a=1}^{\sqrt{V}}\sqrt[3]{\frac{V}{a^2}}\\
\le&\int_1^{\sqrt{V}}\sqrt[3]{\frac{V}{x^2}}\mathrm{d}x\\
=&\sqrt[3]{V}\int_1^{\sqrt{V}}x^{-\frac{2}{3}}\mathrm{d}x\\
=&\sqrt[3]{V}\int_1^{\sqrt{V}}x^{-\frac{2}{3}}\mathrm{d}x\\
=&\sqrt[3]{V}\left(3x^{\frac{1}{3}}\right)\Big|_1^{\sqrt{V}}\\
=&\sqrt[3]{V}\times 3(V^{\frac{1}{2}\times\frac{1}{3}}-1)\\
=&\mathcal{O}(\sqrt{V})
\end{aligned}
$$

> 附注：这种形如 a^2b^3 的数又被称为 Powerful Number，可用于筛积性函数前缀和。

实际上，在 10^{13} 范围内，这种数只有大约 6.8×10^6 个，可以先预处理出 $\sqrt{V}$ 内的质因子，然后搜索每个质因子的指数，得到所有满足条件的数。排序后每次二分回答询问即可。时间复杂度 $\mathcal{O}((T+\sqrt{V})\log V)$。这也同时可以通过测试点 $12\sim 14$。

Part 3

子任务 6

由于任意满足条件的数都可以写成 a^2b^3 的形式，所以可以枚举对询问的每个数**分解质因数**，寻找有没有质因子次幂 $=1$，如果没有即可以分解。分解质因数的复杂度是 $\mathcal{O}(\sqrt{V})$，如果预先处理 $\sqrt{V}$ 中的质数，可以降到 $\mathcal{O}(\sqrt{\frac{V}{\ln V}})$。

时间复杂度 $\mathcal{O}(T\sqrt{V})$。

子任务 7

可以发现，如果一个数合法，那么其所有质因子幂次都 ≥ 2。不妨将幂次 ≥ 3 的因子除去，再检查这个数是否是完全平方数。由于 $p^3\le V$，所以只要枚举 $\sqrt[3]{V}$ 以内的质因子即可。

时间复杂度 $\mathcal{O}(T\sqrt[3]{\frac{V}{\ln V}})$。

子任务 9

更进一步，对于 a^2b^3，$\min(a,b)^5 \le a^2b^3 \le V$，所以 $\min(a,b) \le \sqrt[5]{V} < 4000$，所以如果枚举 4000 内的质因数，并将其除去，那么 a,b 中**至少有一个**会被除干净，即剩下的数要么是完全平方数，要么是完全立方数（也可以反证：如果同时剩下下两个大于 4000 的质因子 p,q，那么 $p^2q^3 > 4000^5 > V$）。

那么枚举 4000 内的质因子并除去，并检验剩下的数是否为完全平方数或完全立方数即可。

时间复杂度 $\mathcal{O}(T\sqrt[5]{\frac{V}{\ln V}})$。

括号序列(bracket)

原题：[LOJ #6043. 「雅礼集训 2017 Day7」蛐蛐国的修墙方案]

概要

本题主要分为三个部分：

- 子任务 1~5：暴力搜索。利用卡特兰数分析复杂度；并运用搜索剪枝，在过程中判断合法性，来省去最后的扫描计算。
- 子任务 6~7：特殊性质。通过手玩得到解，并发现题目的关键性质（与排列相关）。
- 子任务 8~10：正解。运用关键性质，减少枚举量，最终在可接受的复杂度内解决本题。

Part 1

子任务 1

满足条件：$n \le 4$。

手玩，当 $n = 2$ 时，合法括号序列只有 `()`，对应排列 $[2,\ 1]$。

当 $n = 4$ 时，`()()` 对应 $[2,?,4,?]$ 或 $[4,?,2,?]$，`(())` 对应 $[3,4,?,?]$ 或 $[4,3,?,?]$。

子任务 2

满足条件：$n \le 20$。

可以搜索每个位置填 `(` 还是 `)`，再 $\mathcal{O}(n)$ 检查括号序列的合法性，以及构造出的图是否满足条件。

复杂度 $\mathcal{O}(n2^n)$。

子任务 3

可以在搜索的过程中，顺便检查合法性和图是否满足条件，不用最后再检查。复杂度 $\mathcal{O}(2^n)$。

子任务 4

根据[卡特兰数]相关知识（或者打表可知），长度为 $2k$ 的括号序列数量为 $H_k = \dfrac{\binom{2k}{k}}{k+1}$，当 $n = 28$，即 $k = 14$ 时，仅有大约 2.7×10^6 种，所以搜索出所有合法的括号序列（搜索过程中记录两种括号已填的个数，以及前缀 `(` 的数量减 `)` 的数量，用于判断当前该种括号能不能填），再判断图是否满足条件，即可做到复杂度 $\mathcal{O}(nH_{n/2})$。

子任务 5

同理，在搜索过程中顺便判断图是否满足条件，复杂度 $\mathcal{O}(H_{n/2})$，$H_{16} \approx 3.5 \times 10^7$，由于剪枝掉了比较多不合法状态，实际上能通过 $n \le 40$。

Part 2

子任务 6

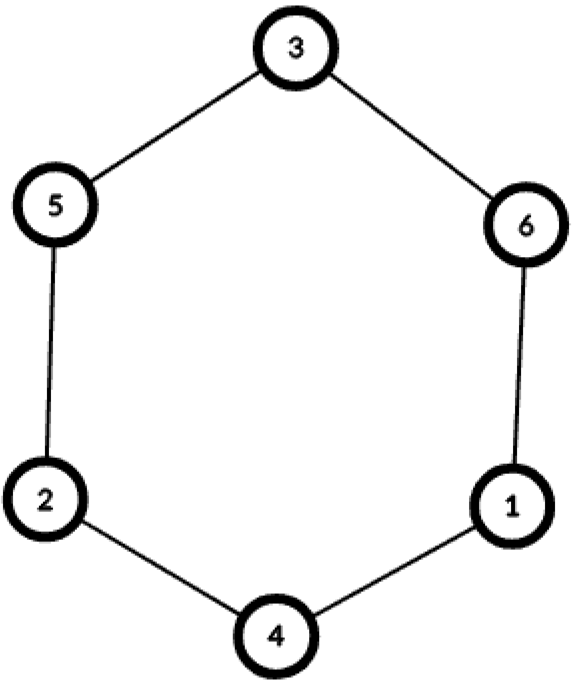
即 $P = [2, 1, 4, 3, 6, 5, \dots, n, n-1]$，由于图每个点度数均为 1，所以开头填 `(` 后，下一个只能填 `)`。这样接着做下去，得到唯一的合法括号序列为 `()()()...()`。

子任务 7

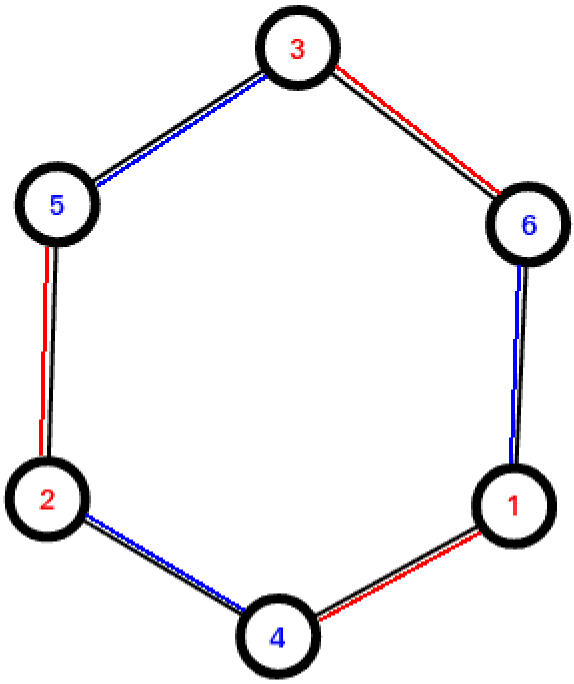
例如 $n = 6$ 时，$P = [4, 5, 6, 2, 3, 1]$。手玩一下可以发现 `((()))` 就是一种合法的括号序列。

考虑其他情况，不妨设前两个括号为 `()`，那么由于 1 与 4 连了边，所以最后一个括号只能是 `)`；由于 $P_2 = 5$，但 2 填了 `)`，为使 5 度数为 1，所以第 5 个括号需要是 `(`，而中间两个无论怎么填，`(` 总会连接到一个度数已经为 1 的点，导致图不合法。

分析上述过程，可以得到，点 i 与点 P_i 的度数，是影响位置 i 是否能放 `(` 的因素，那么不妨对于所有 i，在点 i 和点 P_i 间连一条无向边。由排列的经典结论，这会形成若干个环，环内下标 $i_1, i_2, \dots$ 构成的集合与 $P_{i_1}, P_{i_2}, \dots$ 构成的集合相等。其中填 `(` 相当于选择边 (i, P_i)，否则相当于不选。



关键性质：由于每个点度数必须恰好为 1，所以一个环只有两种满足度数条件的选法，即选第 $1, 3, 5 \dots, n-1$ 条，或者选第 $2, 4, 6, \dots, n$ 条。



（例如上文的排列 P，要么选择点 $1,2,3$ 填左括号，选择边 $(1,4),(2,5),(3,6)$，即图中红色部分；要么选择点 $4,5,6$ 填左括号，选择边 $(4,2),(5,3),(6,1)$，即蓝色部分）

对于该特殊性质，由于只有一个环，而蓝色方案括号序列不合法，所以只有唯一的合法括号序列：`(((...(()))...))`。

Part 3

子任务 8

由于 $i \neq P_i$，所以连接 (i, P_i) 形成的图中最多有 $\frac{n}{2}$ 个环，对每个环枚举 2 种填括号的方式，最后检查括号序列是否合法。复杂度 $\mathcal{O}(n2^{\frac{n}{2}})$。

子任务 9

考虑**特殊处理 2 个点构成的环**，贪心地考虑，在编号较小的那个点填左括号一定更优（环的任意填法都保证左右括号数量相等，那么先填左括号，更不容易出现前缀右括号更多的情况）。那么只需要对于环长 ≥ 3 的环进行枚举。复杂度 $\mathcal{O}(n2^{\frac{n}{3}})$。

子任务 10

对于长度为奇数的环，可以发现，无论如何填括号都不能满足每个点度数为 1 的条件。由于题目中保证给出的排列一定合法，所以**只存在长度为偶数**的环，对于环长 ≥ 4 的环进行枚举，特殊处理环长为 2 的环，即可做到 $\mathcal{O}(n2^{\frac{n}{4}})$，可以通过此题。

数据结构(struct)

原题：[UOJ637【美团杯2021】A. 数据结构]

概要

本题主要分为四个部分：

- 测试点 1~5：暴力枚举及优化。由于 n 较小，可以预处理 n^2 种询问的答案并直接回答。
- 测试点 6~9：复杂度较劣的做法。例如 $\mathcal{O}(n\sqrt{m})$ 或者 $\mathcal{O}((n+m)\log^2 n)$ 的做法。
- 测试点 10~16：特殊性质。可以得到本题需要二维数点，以及正难则反的思想。
- 测试点 17~20：正解。

Part 1

测试点 1~2

对于每次询问，扫描整个序列（如果下标在区间内，则元素 $+1$）并记录哪些颜色出现过。时间复杂度 $\mathcal{O}(nm)$。

测试点 3~5

扫描时，对每种颜色维护出现次数，在此基础上维护当前出现的颜色种类数。对于询问区间左端点相同的所有询问，可以在一次右端点从小到大的扫描中全部解决。所以预处理所有询问的答案。总时间复杂度 $\mathcal{O}(n^2+m)$。

Part 2

测试点 6~9

关于经典问题，区间数颜色数量，有一个莫队做法，显然它在这道题也适用。类似测试点 3~5，维护每种颜色出现次数，和当前出现的颜色种类数。将 i 加入到区间内，相当于 a_i 出现次数 -1，a_i+1 次数 $+1$。时间复杂度 $\mathcal{O}(m+n\sqrt{m})$。

如果是正解，但使用扫描线 + 线段树或者主席树来解决二维数点问题，如果常数较大，可能也只能过这个包。

Part 3

测试点 10~13

原序列中颜色只有奇数。那么不会出现操作后的序列 $[l,r]$ 中的颜色与 $[1,l)\cup(r,n]$ 中颜色相同的情况。答案可以用“$[l,r]$ 中颜色数 $+$ 总颜色数 $-$ 被完全包含在 $[l,r]$ 中的颜色数”计算。

$[l,r]$ 中颜色数是经典问题（区间颜色数），对每个位置 i 预处理上一个颜色相同的位置 pre_i，对于一个区间，只对该颜色第一次出现计数，即 $\sum\limits_i[pre_i<l\wedge l\le i\le r]$，这是一个二维数点问题，可以使用树状数组 + 扫描线解决。

被完全包含在 $[l,r]$ 中的颜色数，对每种颜色预处理第一次出现的位置 mn_c，最后一次 mx_c，即统计 $\sum\limits_i[l\le mn_i\wedge mx_i\le r]$，也是二维数点问题，解决方法同上。

时间复杂度 $\mathcal{O}((n+m)\log n)$。

测试点 14~16

每种颜色只有一个连续段。

发现确定出现过的颜色种数比较困难，正难则反，考虑哪些颜色在最终答案中**不出现**。

不妨设颜色 c 第一次出现的位置为 mn_c，最后一次为 mx_c。

假如颜色 c 和颜色 $c-1$ 都在原序列里出现，且颜色 c 在最终答案中不出现，需要 c 都在区间内，且 $c-1$ 都在区间外，写成条件就是：

$$(l \le mn_c \wedge mx_c \le r) \wedge (mx_{c-1} < l \vee mn_{c-1} > r)$$

由于右侧的两个条件最多成立一个，所以又可以写成：

$$\begin{cases}(mx_{c-1} < l \le mn_c \wedge mx_c \le r) & ,mx_{c-1} < mn_c \\ (l \le mn_c \wedge mx_c \le r < mn_{c-1}) & ,mx_c < mn_{c-1}\end{cases}$$

分讨颜色 c 和颜色 $c-1$ 的另外几种出现情况，同理也是二维数点。时间复杂度 $\mathcal{O}((n+m)\log n)$。

测试点 17~20

考虑在一般情况中颜色 c 何时在答案中不出现。如果颜色 c 在原序列出现过，显然 $[mn_c, mx_c]$ 中不能出现 $c-1$，不妨设 mn_c 前，最靠近 mn_c 的 $c-1$ 的位置为 pre；mx_c 后最靠近的 $c-1$ 位置为 nxt。那么条件就可以写成：

$$(pre < l \le mn_c) \wedge (mx_c \le r < nxt)$$

而如果一个颜色 c 没有出现过，那么只要 $[l, r]$ 区间内不包含 $c-1$ 即可。对于所有颜色为的 $c-1$ 两个相邻位置 pre, nxt，都有：

$$(pre < l < nxt) \wedge (pre < r < nxt)$$

即对于 $c-1$ 的任意一对 (pre, nxt)，如果有 $l, r \in (pre, nxt)$，就说明 c 没有出现过。

这样，也转换成了二维数点问题。对于第二种情况，对于颜色 c，(pre, nxt) 的对数是 $\mathcal{O}(cnt_c)$ 的，总共对数是 $\mathcal{O}(\sum_c cnt_c) = \mathcal{O}(n)$ 的。总复杂度为 $\mathcal{O}((n+m)\log n)$。

构造数组 (array)

原题：[P8863 「KDOI-03」构造数组]

概要

本题主要分为三个部分：

- 测试点 1~8：手玩和暴力枚举。
- 测试点 9~14：特殊性质。$\forall b_i = 1$，转化为更简单问题。
- 测试点 15~25：正解。不断优化"操作有标号，将 a_i 分配到操作中"的计数过程。

Part 1

测试点 1

由于一次操作两个数，所以 $\sum b_i \equiv 1 \pmod 2$ 一定无解。

测试点 2~3

当 $b_1 = 0$ 时方案数为 1，但 $b_1 > 0$，所以输出 0。

测试点 4~5

当 $b_1 = b_2$ 时方案数为 1；否则为 0。

测试点 6~8

按照题意进行搜索，复杂度为 $\mathcal{O}\left(\binom{n}{2}^{\frac{\sum b_i}{2}}\right)$。

Part 2

由于 $\forall i, b_i \ge 1$ 且 $\sum b_i = n$，所以 $\forall i, b_i = 1$。

测试点 9~11

可以使用状压 DP，例如 dp_S 表示前 $\frac{|S|}{2}$ 次操作选了集合 S 中的下标。

测试点 12~14

稍微转化一下，转化为:

有 n 个人要完成 $\frac{n}{2}$ 份大作业，两个人一组完成大作业，需要按顺序提交作业，求方案数。

考虑对于每个人选搭档，第 1 个人有 $n-1$ 种选择，第 2 个人 $n-3$ 种，以此类推，最后乘 $\frac{n}{2}!$ 即可。

Part 3

首先考虑将题目转化一下，也就是令 $m = \frac{\sum_{i=1}^{n} b_i}{2}$（若无法整除则显然无解），即对于 $i = 1, 2, \dots, n$，将 b_i 个 i 填入 m 个**有标号二元组**中，且每个二元组内部不考虑顺序，也不能填入相同的数。要求计数填入所有数的方案，一个方案被形式化地定义为如下集合:

$$A = \{(i, P) \mid P = \{p_1, p_2, \dots, p_{b_i}\}\}$$

对于某个 i，其对应的 $\{p_1, p_2, \dots, p_{b_i}\}$ 表示 i 被填入哪些二元组中。

考虑如何将转化后的一个方案转化为题目所定义的原方案，显然，对于原方案的第 x 次操作，也即标号为 x 的二元组，只需找到 2 个（也仅有 2 个）i_1, i_2 使得 $x \in (i_1, P) \land x \in (i_2, P)$，将 i_1, i_2 按大小填入第 x 个二元组即可。容易证明两类方案一一对应。

例如：$n = 3, b = [1, 2, 3]$:

$$\begin{matrix} [(1,3),(2,3),(2,3)] & & \{(1,\{1\}),(2,\{2,3\}),(3,\{1,2,3\})\} \\ [(2,3),(1,3),(2,3)] & \Leftrightarrow & \{(1,\{2\}),(2,\{1,3\}),(3,\{1,2,3\})\} \\ [(2,3),(2,3),(1,3)] & & \{(1,\{3\}),(2,\{1,2\}),(3,\{1,2,3\})\} \end{matrix}$$

考虑 DP。通过对 A 中 i 从小到大排序，我们可以定义以标号 i 为阶段。而在 b_i 个 i 填入哪些二元组的决策中，依照新方案的形式化定义，决策顺序实际上并不重要。因此，只需将 m 个二元组分类为已经填入 2 个数的、仅填入 1 个数的和未填的，即可完整定义一个状态。

考虑设 $f(i,m_1,m_2)$ 表示仅填完 $1\sim i$，且满足当前 m 个二元组有 m_2 个填入 2 个数的、m_1 个填入 1 个数的和 $m_0=m-m_1-m_2$ 个未填的所有填入方案的数量。容易得到状态转移:

$$f(i,m_1,m_2)=\sum_{0\le k\le \min(b_i,m_2)} f(i-1,m_1+k,m_2-k)\binom{m_1}{k}\binom{m_0}{b_i-k}(m_0\ge 0)$$

但是，这样的转移是 $\mathcal{O}((n+m)m^2)$ 的。

注意到，仅考虑填完 $1\sim i$ 时，$m_1=\sum_{j=1}^{i} b_j-2m_2$，因此我们直接删除 m_1 这一维，即

$$f(i,m_2)=\sum_{0\le k\le \min(b_i,m_2)} f(i-1,m_2-k)\binom{m_1}{k}\binom{m_0}{b_i-k}(m_1,m_0\ge 0)$$

注意到内层的 k 受到 $\min b_i$ 限制，所以 k 的总枚举量为 $\sum b_i=\mathcal{O}(m)$，且 DP 的第二维也是 $\mathcal{O}(m)$ 的。

因此，使用滚动数组，即可在空间允许的情况下做到 $\mathcal{O}(m^2)$ ，实际上由于大量不合法状态，常数进一步缩小，可以通过本题。

另有一种容斥原理的[做法]，可能需要用多项式推式子。