

基于 LLMs 的组合数学解题能力优化研究

张清杰 张茂森

引言：LLMs 的组合数学解题能力难以充分激发

在项目开始，我们首先估计了应用 LLMs 解决组合数学问题的能力上限及优化空间。一方面，我们调研了开发者公布的 o1、gpt-4o 和 deepseek 的数学解题能力数据。如图 1 所示，以 MATH 数据集为例，o1 模型达到了 94.8% 的正确率，DeepSeek 模型为 74.7%，GPT-4o 模型为 60.3%。

图 1: 开发者发布的 GPT-o1、GPT-4o 和 deepseek 的数学解题能力 [1, 2]



另一方面，我们在组合数学问题的测试样例上评估了 o1、gpt-4o 和 deepseek 的正确率。如表 1 所示，我们发现 LLMs 的正确率却显著下降，例如 o1 的正确率由 MATH 数据集上的 94.8% 下降到 67%。由于 MATH 数据集也包括竞赛难度级别的数学问题 [3]，但 LLMs 的表现却优于组合数学测试样例，我们可以认为 LLMs 处理组合数学问题的能力也许没有充分激发，存在优化空间。

表 1: LLMs 在 MATH 数据集和组合数学测试样例上的正确率

数据集	o1	gpt-4o	deepseek
MATH 数据集	94.8	60.3	74.7
组合数学测试样例	67	49	45

那么，为什么 LLMs 的组合数学解题能力难以充分激发呢？我们总结了三个难点：

- 难点一：组合数学解题任务涉及高度专业化的知识
- 难点二：LLMs 的回答缺乏一致性

- 难点三：部分组合数学问题需要系统的推理能力

在后续章节中，我们将一一阐述这三个难点，并给出针对性的解决方案：

- RAG 方法：为 LLMs 提供专业知识
- Self-correction 方法：促进 LLMs 输出正确的回答
- CoT 方法：赋予 LLMs 系统的数学推理能力

对这三种方案以及他们的组合，我们用两类数据集、两种评测方案进行了详细的评测。为了兼顾评测的效率和准确性，我们应用了 LLM 评测和人工评测相结合的方案。

最终，我们依据最优评测结果选定基于 LLMs 解决组合数学问题的最佳方案：
o1+Self-correction+CoT。该方案在期末试题上的最高得分可达 58 分。

难点与解决方案

难点一：组合数学解题任务涉及专业性知识

组合数学解题涉及专业性知识。具体而言，我们通过 15 节课程、8 个章节的教材、8 次作业后才具备参与期末考试的能力，涉及多种专业的知识点：排列组合、格路问题、母函数、Burnside 引理.....

在解决组合数学问题时，LLMs 不知道该应用什么专业知识。例如，表 2 展示了一个 GPT-4o 不会应用组合数学相关专业知识解题的例子。在解决组合数学中典型的小球入盒问题时，GPT-4o 错误地应用了“星星与棒计数方法”，该方法不属于组合数学的范畴，从而导致回答出错。

表 2: GPT-4o 不知道应用什么专业知识来解决问题
(绿色表示正确答案，红色表示错误答案，黄色为关键信息)

问题	参考答案	GPT-4o 回答
有 m 个不同的盒子，将 r 个相同的小球放入其中的 n 个盒子中，要求这 n 个盒子互不相邻且每盒至少包含 k 个球 ($r \geq nk$, $m \geq 2n - 1$)。求满足要求的放球方案数。	先选择放球的盒子，不相邻组合方案数为 $\binom{m-n+1}{n}$ ；再决定盒子中小球数量，相当于 $x_1 + x_2 + \dots + x_n = r - nk$ 的非负整数解，方案数为 $\binom{r-nk+n-1}{n-1}$ 。因此总的方案数即为 $\binom{m-n+1}{n} \binom{r-nk+n-1}{n-1}$ 。	在这个组合数学问题中，我们需要将 r 个相同的小球放入 m 个不同的盒子中 然后，我们需要把 $r - nk$ 个相同的小球放入选定的 n 个盒子中。根据星星与棒的计数方法，这种分配数为： $\binom{r-nk+n-1}{n-1}$ 。总方案计数： $\binom{m-n+1}{n} \binom{r-nk+n-1}{n-1}$ 。将以上两个组合公式结果相乘得到总方案数： $\binom{m-n+1}{n} \binom{r-nk+n-1}{n-1}$

因此，我们需要为 LLMs 提供回答组合数学问题的专业知识。

RAG 方法：为 LLMs 提供专业知识

基于为 LLMs 提供回答组合数学问题的专业知识的目的，我们引入引入 RAG (Retrieval-Augmented Generation) 方法 [4]。RAG 方法为 LLMs 提供解题所需的参考资料，帮助其更准确地理解和应用组合数学知识。LLMs 会自动解析参考资料，在解题时检索相关专业知识。

我们提供的参考资料包括 2 份课程笔记和 8 份作业。参考资料经过预处理，在每份作业的开头总结了其涉及的知识点，确保模型能够高效检索和使用相关内容。图 2 为 1 份参考资料示例，附录 1 展现了 1 份完整的参考资料。

图 2: 参考资料示例。在开头总结了此作业涉及的知识点。
(附录 1 展现了 1 份完整的参考资料。)

该文件主要是burnside引理和polya定理在题目中的应用。当图案数数量较少，可以数清时，往往采用burnside引理。当面对多面体问题的时候，往往采用polya定理。需要注意，当多面体上每一面的涂色具有方向性时，可能部分转动群不再具备不动点，特别是存在一阶循环的转动群。

7.1

n 个人围绕圆桌就座,允许旋转圆桌,用 Burnside 引理求方案数

解:
不考虑等效旋转，方案数为 n 个人的全排列： $n!$ ，为引理的图像数。
分析转动情况：

- 恒等置换 (1个)：和原来保持一致， n 个点全为不动点。
- 顺时针转动 ($n-1$ 个)： $m \times 360/n, m \in [1, n-1]$ ，任何一个图像旋转后没有不动点
每个转动下相同的等价类是同一排列方案，所以根据Burnside引理，方案数为：

$$\frac{n!}{1 + (n-1)} = (n-1)!$$

我们发现应用 RAG 方法后，LLMs 能够避免使用不相关的知识，并且能够给出正确的分析过程和答案。如表 3 所示，使用 RAG 方法后，GPT-4o 避免了“星星与棒”的错误推理路径，基于第一章作业的参考资料给出详细的分析过程和正确答案。值得注意的是，第一章作业中并不包含被测问题，但是包含相似问题。LLMs 可以参照相似问题给出正确的分析过程和答案。

表 3: GPT-4o 使用 RAG 前后案例分析
(绿色表示正确答案，红色表示错误答案，黄色为关键信息)

问题	参考答案	不使用 RAG	使用 RAG
有\$m\$个不同的盒子，将\$r\$个相同的小球放入其中的\$n\$个盒子中，要求这\$n\$个盒子互不相邻且每盒至少包含\$k\$个球	先选择放球的盒子，不相邻组合方案数为 $\binom{m-n+1}{n}$ ；再决定盒子中小球数量，相当于 $x_1 + x_2 + \dots + x_n = r - nk$ 的非负整数解，方案数为	在这个组合数学问题中，我们需要将 r 个相同的小球放入 m 个不同的盒子中 然后，我们需要把 $r - nk$ 个相同的小球放入选定的 n 个盒子中。根据星星与棒的计数方法，这种分配数为： \dots	要解决这个问题，我们需要将符合这些限制条件的排球方案进行计数 然后，计算将 $(r - nk)$ 个球分配到这 (n) 个盒子中，不限制每个盒子可以为空。这与把 (r) 个相同的小球放到 (n) 个不同的盒子中的问题一样，其分配方式有 (C_{r+n-1}^{n-1}) 种选择。 综上所述，总的放球方案数为 $\binom{r-nk+n}{n}$

$(r \geq nk, \text{且 } m \geq 2n - 1)$ 。求满足要求的放球方案数。	$\binom{r - nk + n - 1}{n - 1}$ 。因此总的方案数即为 $\binom{m - n + 1}{n} \binom{r - nk + n - 1}{n - 1}$ 。	$\binom{r - nk + n - 1}{n - 1}$ 。总方案计数：将以上两个组合公式结果相乘得到总方案数： $\binom{m - n}{n - 1} \times \binom{r - nk + n - 1}{n - 1}$ 。	$\times C_{r'+n-1}^{n-1}$ 其中， C_{m-n+1}^n 确保了盒子的位置是互不相邻的， $C_{r'+n-1}^{n-1}$ 负责在这些确定位置的盒子中分配剩余的球[0]。
--	---	---	--

[0] 1-第一章作业.md

难点二：LLMs 的回答缺乏一致性

我们也调研了学术界运用 LLMs 解决数学问题的难点。[5]提出 LLMs 回答缺乏一致性是解决数学问题的难点。

对于同一道题的不同表现形式、不同推理过程，LLMs 可能会给出不同的回答。甚至对于相同问题的相同表现形式、相同推理过程，LLMs 也可能给出不同的回答。如表 4 所示，对于同一个问题，GPT-4o 的两次回答可能是不同的。

表 4: GPT-4o 的回答不具备一致性
(绿色表示正确答案，红色表示错误答案)

问题	参考答案	回答 1	回答 2
<p>现有 23 个不同的物体，将其放入 25 个不同的箱子，每个箱子中至多放入一个物体，且禁止出现相邻的空箱子，求方案数。</p>	<p>共有 $\binom{24}{2}$ 种方案选择两个不相邻的空箱子，然后排列所有物体，方案数为 $\binom{24}{2} \cdot 23! = \frac{23!}{2!} \cdot 24!$。</p>	<p>首先，我们来分析这个问题。我们有 23 个不同的物体和 25 个不同的箱子，</p> <p>.....</p> <p>综合以上步骤，符合条件的所有摆放方案的总数如下：$23! \times \binom{24}{2}$</p>	<p>要将 23 个不同的物体放入 25 个不同的箱子，要求每个箱子至多放入一个物体，并且禁止出现相邻的空箱子</p> <p>.....</p> <p>因此，所有方案总数可以写作：$\binom{25}{2} \times 23!$</p>

这些不同的回答中包括正确的回答和错误的回答。正确回答的存在说明 LLMs 具备解题能力，错误回答的存在说明 LLMs 回答缺乏一致性是正确解题的难点。因此，我们需要促进 LLMs 输出正确的回答。

Self-correction 方法：促进 LLMs 输出正确回答

为了促进具备解题能力的 LLMs 输出正确的回答，我们引入 Self-correction 方法 [6]。其核心思想在于让 LLMs 对其初始回答进行检查，并基于检查结果重新生成答案，具体分为如下三步（附录 2 展现了 Self-correction 提示语）：

1. 回答生成：模型生成初始回答。

2. **回答检查：**模型对初始回答进行自我检查。如果认为答案没问题，则结束；如果认为答案有问题，则进入回答修正。

3. **回答修正：**模型基于检查结果生成修正后的答案。

通过这一过程，模型能够进一步优化回答质量，促进生成正确的回答。如表 5 所示，self-correction 方法帮助模型将错误的回答更正为正确的回答。

表 5: Self-correction 使用前后案例分析

(绿色表示正确答案，红色表示错误答案)

问题	参考答案	Self-correction 前	Self-correction 后
从字典 $\{c_1, c_2, \dots, c_D\}$ 中选出 n 个字符组成字符串，要求字符 c_i 出现次数为 f_i ，保证 $\sum_{i=1}^D f_i = n$ 。设非负整数 k 满足 $k(D+1) > n$ ，求子串 $c_1 c_2 \dots c_D$ 的出现次数恰好为 k 的字符串的数目。	若存在 $f_i < k$ ，则方案数显然为 0。否则，先“打包” k 个子串 $c_1 c_2 \dots c_D$ ，然后在其中插入剩下的 $n - kD$ 个字符。安排剩下的字符位置的方案数为 $\binom{n - kD + k}{k}$ ，这些字符排列的方案数为 $\frac{(n - kD)!}{\prod_{i=1}^D (f_i - k)!}$ ，因此总的方案数即为 % $\begin{equation*} \binom{n - kD + k}{k} \frac{(n - kD)!}{\prod_{i=1}^D (f_i - k)!} = \frac{(n - kD + k)!}{\prod_{i=1}^D (f_i - k)!} \end{equation*}$	To solve this problem, we need to find the number of strings of length n formed from the alphabet $\{c_1, c_2, \dots, c_D\}$, **Number of Such Strings:** Given the constraints, the only way to create a string of length n where the substring $c_1 c_2 \dots c_D$ appears exactly k times is to construct the string as above, starting with c_1 and continuing the overlapping pattern. Since the characters are fixed in this pattern, there is only one such string possible. Answer: $n!$	To solve this problem, we need to calculate the number of strings of length n composed of characters from the set $\{c_1, c_2, \dots, c_D\}$, **Step 5: Calculate the Total Number of Strings** Multiply the number of ways to choose starting positions of substrings by the number of ways to arrange the remaining letters: $\text{Total number of strings} = C(n - k(D - 1), k) \times \frac{(n - kD)!}{\prod_{i=1}^D (f_i - k)!}$

难点三：部分试题需要系统的数学推理能力

我们发现，RAG 方法和 Self-correction 方法并不总是起效的。对于特定试题，例如特殊设定的多面体转动群分析问题，LLM 模型甚至无法通过题意的理解掌握多面体的基本特征，更不可能在此基础上利用 polya 定理（或 burnside 引理）解决问题，如表 6：

表 6-1: LLM 对多面体特征无法理解
(绿色表示正确答案)

题目	正确答案
奇怪多面体。现有这样一个多面体，每一面均为正三角形或正方形，每个顶点均与两个三角形和两个正方形相连，三角形和正方形是交错排列的。1) 请给出这个多面体的边数、面数和棱数。	<p>欠角为 60 度，所以有 12 个顶点。(2 分)</p> <p>每个顶点都关连两个三角形和两个正方形，所以有 8 个三角形和 6 个正方形，总共 14 个面。(2 分)</p> <p>每条棱被两个面共用，总共 24 条棱。(2 分)</p>

表 6-2: (续表) LLM 对多面体特征无法理解
(红色表示错误答案，黄色为我们的总结)

gpt-4o	gpt-o1-preview	deepseek
<p>面数：截半立方体由 8 个正三角形和 12 个正方形组成，所以共有 20 个面。</p> <p>边数：使用欧拉公式 $(V - E + F = 2)$，其中 (V) 是顶点数，(E) 是边数，(F) 是面数。先计算顶点数，每个顶点连接 2 个三角形和 2 个正方形，形成一个顶点环，因此有 24 个顶点。</p> <p>代入欧拉公式：$(24 - E + 20 = 2)$，可以计算得：$(E = 42)$</p>	<p>32 triangular faces and 6 square faces.*</p> <p>24 rotational symmetries, forming the *chiral octahedral group*.</p>	<p>思考步骤高达 27 步仍然没有给出数值答案</p> <p>.....</p> <p>步骤 27：具体计算</p> <p>假设多面体有 (n) 个顶点，每个顶点连接两个三角形和两个正方形。每个顶点连接 4 个面，.....可以得到：$(n - \frac{2n}{2} + F = 2)$</p>

因为这类问题需要系统的数学解题能力，需要通过边、顶点和面之间的对应关系，以及欠角和等知识进行逐步推理。更进一步，在进行转动群分析时，往往存在列举不清晰，循环数分析错误等情况。因为这一部分的推理链较长，LLM 在没有推理支撑的情况下，会出现逻辑混乱的情况，甚至跳过作答，如表 7。

表 7: LLM 分析转动群所遇到的困难
(黄色为我们的总结)

题目	正确答案	gpt-4o	gpt-o1-preview
奇怪多面体。现有这样一个多面体，每一面均为正三角形或正方形，每个顶点均与两个三角形和	<p>2) (共 8 分，转动群阶数 1 分，5 种类型每错一种扣 2 分，扣完为止)</p> <p>1. 1 个不动置换，三角形面 $(1)^8$，四边形面 $(1)^6$</p> <p>2. 8 个三角形面心对面心旋转</p>	<p>分析笼统，无得分点</p> <p>截半立方体的对称性和立方体相同，即它的对称群是与立方体同</p>	<p>分析混乱，无得分点</p> <p>Total number of colorings without considering symmetry:</p> <p>...</p> <p>Total colorings: (2^{32})</p>

两个正 方形相 连，三角形和 正方形是交错 排列的。1) 请 给出这个多面 体的边数、面 数和棱数。(6 分) 2) 请以面 为对象给出这 个多面体的转 动群。(8 分	± 120 度,, 三角形面 $(1)^2(3)^2$, 四边形面 $(3)^2$ 3. 6 个四边形面心对面心旋转 ± 90 度, 三角形面 $(4)^2 (1)^2$, 四边形面 $(4)^1$ 4. 3 个四边形面心对面心旋转 180 度, 三角形面 $(2)^4 (1)^2$, 四边形面 $(2)^2$ 5. 6 个顶点对顶点旋转 180 度, 三角形面 $(2)^4$, 四边形面 $(2)^3$	构的群。这个群 是标号为 (O_h) 的群, 排除平移 和反射, 关于旋 转部分是立方体 的旋转群, 和八 面体相同, 有 24 个元素。具体的 转动包括旋转和 面对称等操作。	$\times 4^6$) - Identity Element: Fixes all colorings: $(\text{Fix})(e) = 2^{32}$ $\times 4^6$) - Non-Identity Rotations: ... Total fixed colorings: $(2$ $\times 4 = 8$)
---	---	---	--

因此，我们需要对部分试题赋予 LLMs 系统的数学推理能力，在本次实验中，我们针对解题思路较为固定，答案较为规整的多面体转动群问题进行了 CoT 思维链的方法设计，来提升 LLM 在该类题目上的解题能力。

CoT 方法：赋予 LLMs 系统的数学推理能力

我们针对解题思路较为固定，答案较为规整的多面体转动群问题进行了 CoT (Chain of Thought) 思维链 [7] 的方法设计，我们将多面体转动群问题分解为（附录 3 展现了 CoT 提示语）：

1. 多面体信息理解
2. 转动群分析并列表格
3. 不动点分析并列表格
4. 根据 polya 定理或 burnside 引理列出计算公式

在多步 CoT 的引导下，LLM 应对该类问题的能力有了明显的提升，在表 8 中我们展示了 gpt-o1 模型在 CoT 前后的表现。

表 8-1: CoT 方案应用于 gpt-o1 的案例效果展示

题目	正确答案
奇怪多面体。现有这样一个多面体，每一面均为正三角形或正方形，每个顶点均与两个三角形和两个正方形相连，三角形和正方形是交错排列的。1) 请给出这个多面体的边数、面数和棱数。(6 分) 2)	欠角为 60 度，所以有 12 个顶点。(2 分) 每个顶点都关连两个三角形和两个正方形，所以有 8 个三角形和 6 个正方形，总共 14 个面。(2 分) 每条棱被两个面共用，总共 24 条棱。(2 分) 2) (共 8 分，转动群阶数 1 分，5 种类型每错一种扣 2 分，扣完为止) 1. 1 个不动置换，三角形面 $(1)^8$, 四边形面 $(1)^6$ 2. 8 个三角形面心对面心旋转 ± 120 度,, 三角形面 $(1)^2(3)^2$, 四边形面 $(3)^2$

请以面为对象给出这个多面体的转动群。 (8分	3. 6个四边形面心对面心旋转 $\pm 90^\circ$ 度，三角形面 $(4)^2(1)^2$ ，四边形面 $(4)^1$ 4. 3个四边形面心对面心旋转 180° 度，三角形面 $(2)^4(1)^2$ ，四边形面 $(2)^2$ 5. 6个顶点对顶点旋转 180° 度，三角形面 $(2)^4$ ，四边形面 $(2)^3$
------------------------------------	--

表 8-2: (续表) CoT 方案应用于 gpt-o1 的案例效果展示
(绿色为正确答案，红色为错误答案，黄色为我们的总结)

使用 CoT 前

1)

32 triangular faces and
6 square faces.

24 rotational
symmetries, forming
the chiral octahedral
group.

2)

分析笼统，无得分点

截半立方体的对称性和立
方体相同，即它的对称群
是与立方体同构的群。这
个群是标号为(O_h)的
群，排除平移和反射，关
于旋转部分是立方体的旋
转群，和八面体相同，有
24 个元素。具体的转动包
括旋转和面对称等操作。

使用 CoT 后

1)

.....

共有 (6 \times 4 = 24) 条棱，同样每条棱共享了两次，所以来自正方形的棱数为 (24/2 = 12)。
- 总棱数: (E = 12 + 12 = 24)。
回答：
这个多面体有 12 个顶点，14 个面 (8 个正三角形和 6 个正方形)，以及 24 条棱。

2)已能输出大量正确得分点

(n) 置换类型 | 元素个数 | 正三角形面的循环 | 正方形面的循环 (n).....
(n) 恒等变换 | 1 |(n) 四阶旋转 | 6 | (T 循环 1)(T 循环 2) | (S 固定)(S 循环) | (n)

旋转类型	同类中元素个数	对 8 个三角形的循环分解	对 6 个正方形的循环分解
1. 恒等变换 (Identity)	1	(1^8)	(1^6)
2. 绕面心轴转 180°	3	(2^4)	(2^3)
3. 绕边心轴转 180°	6	(2^4)	(2^3)
4. 绕顶点轴转 120° (含 120° 与 240°)	8	(3^2)(1^2)	(3^2)
5. 绕面心轴转 90° (含 90° 与 270°)	6	(4^2)	(1^2)(4^1)

在实际采用中，我们预先判断该题目是否需要采用该类题目的求解流程，如果需要，则采用 CoT 的方式提升其效果，避免分步 prompt 对其他题目的求解产生干扰。

评测

评测方案：人工评测和 LLM 评测相结合

除了前文展示的优化方案起效的例子，我们也需要系统性的评测方案来展示优化方法的统计性结果。

在这项工作中，我们应用了两种评测方案，兼顾评测的效率和准确性，分别针对两种数据集：

- LLM 评测：包括详细打分规则的测试样例（52 题）

• 人工评测：2017-2019 期末考试题（15 题）

提出 LLM 评测的动机在于人工评测的工作量过大。评测问题的回答需要根据参考答案和打分规则来打分，严格根据参考答案和规则打分的人工工作量是很大的。这是因为参考答案包括 markdown、latex 等多种格式，可读性相对较差；而且打分规则也是具体细分的，需要仔细判断。如表 9 所示，评测一道问题需要在 latex 格式中分析 4 种打分规则。人工打分一道测试样例的耗时是 5min，无法满足我们快速迭代优化方法的需求。

因此，我们额外设计了一种 LLM 评测方案，让 LLM 根据参考答案和打分规则来给回答打分。LLM 评测的输入是问题，回答，参考答案和打分规则；输出是分析过程和分数。我们利用 one-shot learning 的思想，为 LLM 提供了一个如何进行分析并打分的例子。为了保证打分的合理性，分数是在分析过程中根据每个打分规则累加所得的。附录 4 展现了 LLM 评测提示语。有了 LLM 评测方法，我们可以快速评测我们设计的优化方法的有效性。一道题的评测时间由人工评测的 5min 降低到 LLM 评测的 8 秒。值得注意的是，LLM 评测方法需要提供详细的打分规则。如果打分规则没有提供，则只能通过人工评测。

表 9: LLM 评测案例，包括：问题、参考答案、打分规则、回答、分析过程和分数

问题	参考答案	打分规则	回答	分析过程	分数
有 n 个小写英文字母， 通过讨论各 a_i 的奇偶性，求符合要求的字符串数目。	分类讨论： % $\begin{itemize} \item$ 若有两个或以上 a_i 为奇数， 去除这种情况后，答案为 $\binom{n}{a_1} \cdot \binom{n}{a_2} \cdot \dots \cdot \binom{n}{a_k} - \binom{n}{a_1/2} \cdot \binom{n}{a_2/2} \cdot \dots \cdot \binom{n}{a_{(26)/2}}$ 。 $\end{itemize}$	{'rule': '得到总回文串数量', 'score': 1}, {'rule': '多余一个奇数的情况正确', 'score': 3}, {'rule': '恰好一个奇数的情况正确', 'score': 3}, {'rule': '全为偶数的情况正确', 'score': 3}	在这个问题中，我们要计算由 n 个小写英文字母组成的 总数计算简化： 实际上，构成非回文的方式数目相当于总的排列数 ($n! / \prod_{k=1}^{26} a_k!$) 减去构成回文的可能性数目.....	打分分析过程： $\backslash \backslash$ 初始给分为 0。 $\backslash \backslash$ 分析第 1 个批改规则： 因此我们给 4（之前总分）+ 0（不符合该规则，不给分）= 4（当前总分）。 $\backslash \backslash$ 总分： $\backslash \backslash$ 4	4

评测结果

表 10、表 11 和表 12 依次是 RAG 方法，self-correction 方法和 CoT 方法的评测结果。我们的评测指标为依照评分标准并换算成百分制的得分。

RAG 方法有效

由于 OpenAI API 在 o1 模型上不支持 RAG 使用，我们在 GPT-4o 上进行 RAG 方法的统计性实验。我们尝试了两种 RAG 方案：

- RAG (textbook)：以组合数学教材为参考资料
- RAG (homework)：以 2 份课程笔记和 8 份作业为参考资料

结果如表 10 所示，应用 RAG 方法后，GPT-4o 在测试样例上的得分有明显上升，其中 RAG (homework) 使得分由 41 分上升到 48 分。这意味着 RAG 方法成功为 LLMs 提供了组合数学专业知识并提高了回答正确性。然而，在期末试题上 RAG 方法没有带来得分提升，这可能是因为我们为 LLMs 提供的参考资料里包括的是教材或作业题，不包括与期末试题风格相近的题目。另外，我们也发现 RAG (homework) 比 RAG (textbook) 的效果更好，这可能是因为测试样例与作业题相近，便于 LLMs 检索相关知识来回答，而教材篇幅过长，LLMs 难以检索出其中有助于答题的信息。

表 10: GPT-4o 的 RAG 优化结果

数据集	GPT-4o	GPT-4o + RAG(textbook)	GPT-4o + RAG(homework)
测试样例	41	43	48
期末试题	20	20	20

Self-correction 方法有效

我们在 o1 和 GPT-4o 模型上进行 self-correction 方法的统计性实验。结果如表 11 所示，self-correction 方法可以显著提升 LLMs 的组合数学解题能力。例如对于 o1 模型，self-correction 方法可以将期末试题得分从 40 分提升到 54 分。另外，o1 的解题能力整体上是优于 GPT-4o 的（包括 GPT-4o + RAG），因此我们的最终方案可以包括搭载了 self-correction 优化方法的 o1 模型。

表 11: o1, GPT-4o 的 Self-correction 优化结果

数据集	o1	o1 + Self- correction	GPT-4o	GPT-4o + Self- correction
测试样例	67	75	41	51
期末试题	38	43	20	27

CoT 方法有效

我们在 o1 和 GPT-4o 模型上进行 CoT 方法的统计性实验。由于 CoT 方法专属于

处理多面体转动群问题，此小节的评测数据集包括我们从作业题和期末试题中挑出的共 7 道多面体转动群问题。结果如表 12 所示，CoT 方法可以显著提升 LLMs 解决特定问题的能力。例如对于 o1 模型，CoT 方法将得分从 57 分提升至 64 分。另外，由于 o1 的解题能力整体上优于 GPT-4o，且 o1 + CoT 的效果比 o1 + Self-correction 的效果也好，因此在处理多面体转动群问题时，我们的最终方案里可以采用搭载了 CoT 优化方法的 o1 模型。

表 12: o1, GPT-4o 的 CoT 优化结果

数据集	o1	o1 + Self-correction	o1 + CoT	GPT-4o	GPT-4o + CoT
7 道旋转题	57	57	64	29	41

最终结果

综合前 3 小节的结果，我们即可给出期末试题的最优解决方案和评测结果。CoT 方法用于多面体转动群问题，Self-correction 方法用于其他问题。最终结果如表 13 所示，我们的方案在三届期末试题上可以取得 44 分的平均得分，其中 2016-2017 的期末试题上可以得到 58 分的最优得分。

表 13: o1 综合 Self-correction 优化和 CoT 优化的最终结果

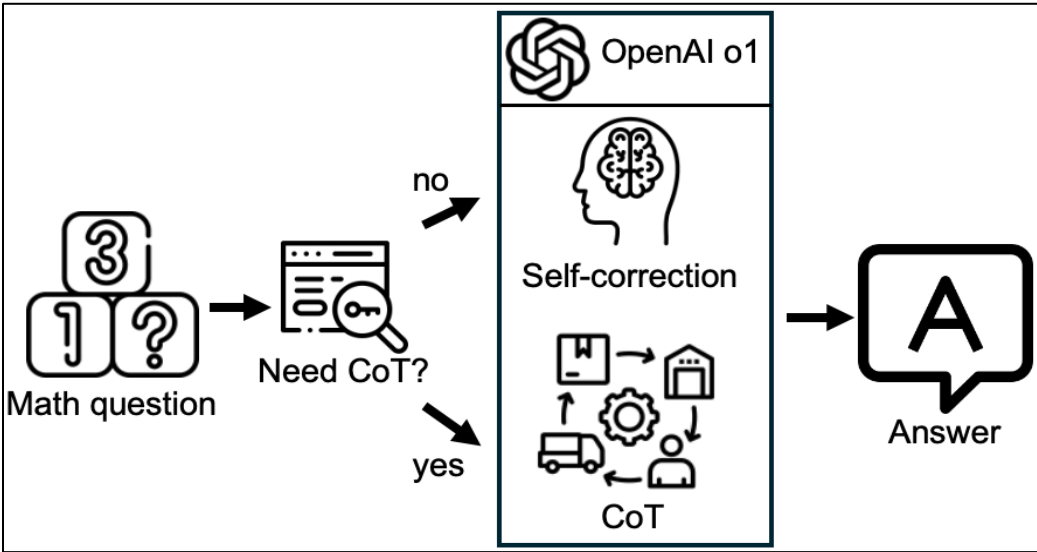
数据集	取值	o1	o1 + CoT + Self-correction
期末试题	平均值	38	44
	最优值	53	58

最终方案：o1 + CoT + Self-correction

结合优化方案和最优评测结果，我们的最终方案如图 3 所示：

- 以 o1 为基础模型。
- 判断问题是否为多面体转动群问题：是否包含关键词“polya”，“burnside”或“转动群”。
- 如果是，采用 CoT 优化方法。
- 如果不是，采用 Self-correction 优化方法。

图 3: 最终方案流程图



总结

针对 LLMs 处理组合数学问题的能力，我们探索了三个难点：

- 难点一：组合数学解题任务涉及高度专业化的知识
- 难点二：LLMs 的回答缺乏一致性
- 难点三：部分组合数学问题需要系统的推理能力

并针对每个难点设计了解决方案：

- RAG 方法：为 LLMs 提供专业知识
- Self-correction 方法：促进 LLMs 输出正确的回答
- CoT 方法：赋予 LLMs 系统的数学推理能力

对这三种解决方案以及他们的组合，我们用两类数据集（测试样例和期末试题）、两种评测方案（LLM 评测和人工评测）进行了详细的评测。

最终，我们依据最优评测结果选定基于 LLMs 解决组合数学问题的最佳方案：**o1+Self-correction+CoT**。该方案在期末试题上的最高得分可达 **58** 分。

参考文献

[1] <https://openai.com/index/learning-to-reason-with-llms/>

[2] <https://www.deepseek.com/>

[3] Hendrycks D, Burns C, Kadavath S, et al. Measuring mathematical problem solving with the math dataset[J]. arXiv preprint arXiv:2103.03874, 2021.

[4] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks[J]. Advances in Neural Information Processing Systems, 2020, 33: 9459-9474.

[5] Ahn J, Verma R, Lou R, et al. Large language models for mathematical reasoning: Progresses and challenges[J]. arXiv preprint arXiv:2402.00157, 2024.

[6] Shinn N, Cassano F, Gopinath A, et al. Reflexion: Language agents with verbal reinforcement learning[J]. Advances in Neural Information Processing Systems, 2024, 36.

[7] Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models[J]. Advances in neural information processing systems, 2022, 35: 24824-24837.

附录

附录 1: RAG 的 1 份参考资料示例

Markdown

该文件主要是 burnside 引理和 polya 定理在题目中的应用。当图案数数量较少，可以数清时，往往采用 burnside 引理。当面对多面体问题的时候，往往采用 polya 定理。需要注意，当多面体上每一面的涂色具有方向性时，可能部分转动群不再具备不动点，特别是存在一阶循环的转动群。

7.1

n 个人围绕圆桌就座,允许旋转圆桌,用 Burnside 引理求方案数

解：

不考虑等效旋转，方案数为 n 个人的全排列： $n!$ ，为引理的图像数。

分析转动情况：

- 恒等置换（1 个）： 和原来保持一致， n 个点全为不动点。
- 顺时针转动（ $n-1$ 个）： $m \times 360/n, m \in [1, n-1]$ ，任何一个图像旋转后没有不动点

每个转动下相同的等价类是同一排列方案，所以根据 Burnside 引理，方案数为：

$$\frac{n!}{1+(n-1)}=(n-1)!$$

7.2

设想将一个大正方形划分为四个边长为原先二分之一的小正方形，使用黑、白两种颜色对每个小正方形染色。允许的对称操作包括：

- (1) 大正方形在其所在平面内绕其中心旋转；
 (2) 将染色方案中的黑色改为白色、白色改为黑色。
 求不等价的染色方案数。

****解： ****

不考虑旋转，染色方案有： $2^4=16$ 种，其分布如下图所示：

考虑旋转的情况

1. 恒等置换：16 个不动点
2. 顺时针 90° /逆时针 90° ： a,i 两个不动点。（两种情况，共 4 个）
3. 旋转 180° ： a,i,h,p 四个不动点
4. 恒等置换+颜色反转：没有不动点
5. 顺时针 90° 、逆时针 90° +颜色反转： h,p 两个不动点（两种情况，共 4 个）
6. 旋转 180° +颜色反转： f,n,,g,o 四个不动点

综上所述，按照 burnside 引理：

\$\$

$$\frac{16+4+4+0+4+4}{1+2+1+1+2+1}=\frac{32}{8}=4$$

\$\$

所以不等价的染色方案数为 4。

7.3

在正四面体的每个面上引一条高，允许旋转，求不等价的方案数。

****解： ****

正四面体每个面都是等边三角形，读了很多次题目，理解为，每个平面上单独做一条高，有多少种不同的方案。等价于三种涂色，对正四面体进行涂色。将面标号为 1,2,3,4，对位置的顶点标号为 a,b,c,d。

分析其置换群，正四面体有三种旋转置换：

1. 不动置换**1 个**： 4 个面都是不动点，循环表示为 $(1)^4$ ，不动图像为 1×3^4
2. 按照顶点和对面中心的连线进行顺时针 120° 和 240° 旋转，四个面引出的高，总共**8 个**置换：每个置换均有底面为 1 个不动点。循环表示为 $(1)(3)$ ，因为旋转中，高的方向一定发生变化，所以没有不动图像。
3. 以对轴进行旋转 180° ：没有不动点，共享旋转轴垂直的楞的两面互换，**3 个**。循环表示为 $(2)^2$ ，旋转中高方向可能不发生改变，不动图像数为 3×3^2 。

根据 polya 定理，方案数为：

\$\$

$$\frac{1 \times 3^4 + 0 + 3 \times 3^2}{1 + 8 + 3} = 9$$

\$\$

7.4

正方形肖像的边长与一个正方体的棱长相同. 将 6 幅相同的此肖像贴在正方体的 6 个面上, 允许旋转, 求不等价的方案数

****解: ****

首先需要考虑到, 肖像的方向问题, 肖像贴在平面上, 有 4 种方向。

下面讨论正方体的置换群:

1. 不动置换****1 个****: 六个面均为不动点, 循环表示为 $(1)^6$, 不动图像数为 1×4^6
2. 面心-面心进行顺时针 or 逆时针 90° 置换****6 个****: 循环表示为 $(1)^2(4)$, 因为转动时一阶循环的肖像方向一定发生改变, 所以不存在不动图像。
3. 面心-面心进行 180° 旋转****3 个****: 循环表示为 $(1)^2(2)^2$, 因为旋转时一阶循环肖像方向一定改变, 所以不存在不动图像。
4. 棱心-棱心进行 180° 旋转****6 个****: 循环表示为 $(2)^3$, 不动的图像数为 6×4^3
5. 以对角线进行旋转 120° , 240° 共****8 个****: 循环表示为 $(3)^2$, 不动的图像数为 8×4^2

根据 polya 定理得到:

\$\$

$$\frac{1 \times 4^6 + 0 + 0 + 6 \times 4^3 + 8 \times 4^2}{1 + 6 + 3 + 6 + 8} = 192$$

\$\$

7.5

给定 12 种不同的彩色. 将一个经典样式的足球的每个正六边形面染成白色, 每个正五边形面则染成给定的彩色之一, 且每种颜色只染一个面; 允许旋转, 求不等价的染色方案数

****解: ****

首先计算有多少个正五边形。

足球一个角由一个正五边形和两个正六边形组成。每个顶点的欠角为: $360 - 2 \times 120 - 108 = 36^\circ$

所以有顶点数: $720 / 36 = 60$

所以正五边形有: $60 / 5 = 12$

六边形有: $60 \times 2 / 6 = 20$ 个

分析置换群:

1. 不动置换, 1 个, $12!$ 个不动点
2. 以相对的五边形为轴进行旋转, $4 \times 6 = 24$ 个, 没有不动点
3. 以相对的六边形为轴进行旋转, $2 \times 10 = 20$ 个, 没有不动点
4. 以相对的正六边形的棱心进行旋转, $3/2 \times 20/2 = 15$ 个, 无不动点

所以根据

burnside 引理有:

\$\$

$$\frac{12!}{60}=7983360$$
 \$\$

7.6

设想一个正八面体，用红、蓝两色对顶点染色，用黄、绿两色对面染色。要求恰有 4 个顶点染为红色（其余为蓝色）、恰有 4 个面染为黄色（其余为绿色），允许旋转，求不等价的染色方案数

解：burnside 引理

正八面体，每个面为等边三角形，6 个顶点，12 个棱。

分析正八面体的置换群为(置换类型，数量，顶点循环表示，面循环表示，不动点个数)：

1. 不动置换，1 个， $(1)^6, (1)^8$ ，不动点个数为 $1 \times \frac{6!8!}{4!2! \times 4!4!} = 15 \times 70 = 1050$ 个
2. 跟根据相对的顶点连线进行顺/逆时针 90 旋转， $6/3 \times 2 = 6$ 个， $(1)^2(4)^1, (4)^2$ ，不动点个数为 $6 \times 1 \times 2 = 12$
3. 根据相对顶点连线旋转 180， $6/2 = 3$ 个， $(1)^2(2)^2, (2)^4$ ，不动点个数为 $3 \times (C_2^2)_3 \times (C_2^2)_4 = 9 \times 6 = 54$
4. 根据相对棱心旋转 180， $12/2 = 6$ 个， $(2)^3, (2)^4$ ，不动点个数为 $6 \times (C_3^2)_3 \times (C_4^2)_2 = 6 \times 3 \times 6 = 108$
5. 根据相对面心旋转 120， $8/2 \times 2 = 8$ 个， $(3)^2, (1)^2(3)^2$ ，不动点个数为 0，因为一定有一组顶点的颜色不相同。

根据 burnside 引理：

\$\$

$$\frac{1050+12+54+108}{1+6+3+6+8} = \frac{1224}{24} = 51$$
 \$\$

解 2：母函数表示法的 polya 定理

相同的置换群，使用 polya 定理的母函数形式进行求解，假设顶点染色为红表示为 R，染为蓝表示为 B；面染色为黄为 Y，染色为绿为 G。

分析正八面体的置换群为(置换类型，数量，顶点循环表示，面循环表示，函数表示)：

1. 不动置换，1 个， $(1)^6, (1)^8$ ， $(R+B)^6(Y+G)^8$
2. 跟根据相对的顶点连线进行顺/逆时针 90 旋转， $6/3 \times 2 = 6$ 个， $(1)^2(4)^1, (4)^2$ ， $6(R+B)^2(R^4+B^4)(Y^4+G^4)^2$
3. 根据相对顶点连线旋转 180， $6/2 = 3$ 个， $(1)^2(2)^2, (2)^4$ ， $3(R+B)^2(R^2+B^2)^2(Y^2+G^2)^4$
4. 根据相对棱心旋转 180， $12/2 = 6$ 个， $(2)^3, (2)^4$ ， $6(R^2+B^2)^3(Y^2+G^2)^4$

5. 根据相对面心旋转 120° , $8/2 \times 2 = 8$ 个, $(3)^2, (1)^2(3)^2$,
 $8(R^3+B^3)^2(Y+G)^2(Y^3+G^3)^2$

根据 polya 定理, $R^4B^2Y^4G^4$ 的系数为:

\$\$

$$\frac{C_6^4 C_8^4 + 6C_2^1 + 3(C_2^0 C_2^1 + C_2^2 C_2^0) C_4^2 + 6C_3^2 C_4^2}{24} = 51$$

\$\$

附录 2: Self-correction 提示语

Python

```
if opts.selfCorrection:
```

```
    self_correction_prompt=f"""
```

请检查关于组合数学的问题的第一次回答的正确性, 并给出第二次回答。如果第一次回答正确, 请直接输出第一次回答; 如果第一次回答错误, 请输出正确的第二次回答。

```
    问题: {problem}
```

```
    第一次回答: {answer}
```

```
    第二次回答:
```

```
    """
```

```
    answer=getAnswer(self_correction_prompt)
```

```
    res['second_answer']=answer
```

附录 3: CoT 提示语

Python

```
cotPrompt="""
```

请根据以下解题思路回答上述问题。

这类问题需要用到 **burnside** 引理或者 **polya** 定理, 具体的解题思路如下

1. 根据欧拉公式, 和棱、面、点之间的数值关系, 求出多面体的信息

2. 根据多面体的形态, 分析其转动群, 并列表出 |置换类型|转动群数量|不同面形的转动循环情况|, 其中循环情况应以转动循环的形式展示

3. 然后根据涂色或者图案进行上色, 涂色没有方向限制, 视为相同方案。但是图案具有方向性, 不同方向的方案不同 (当出现一阶循环的时候不存在不动点, 高阶循环阶乘基数改变)。同样列出表格 |置换类型|转动群数量|不同面形的转动循环情况|不动点个数|, 其中循环情况应以转动循环的形式展示, 不动点个数应以指数相乘的形式展示

4. 最后列出计算公式并求方案数

```

"""
step1prompt="请执行：1.根据欧拉公式，和棱、面、点之间的数值关系，求出多
面体的信息"
step2prompt="请执行：2.根据多面体的形态，分析其转动群，并列表出 |置换
类型|转动群数量|不同面形的转动循环情况|，其中循环情况应以转动循环的形式
展示"
step3prompt="请执行：3.然后根据涂色或者图案进行上色，涂色没有方向限制，
视为相同方案。但是图案具有方向性，不同方向的方案不同（当出现一阶循环的时
候不存在不动点，高阶循环阶乘基数改变）。同样列出表格|置换类型|转动群数量|
不同面形的转动循环情况|不动点个数|，其中循环情况应以转动循环的形式展示，
不动点个数应以指数相乘的形式展示"
step4prompt="请执行：4.最后列出计算公式并求方案数"

stepPrompts=[step1prompt,step2prompt,step3prompt,step4prompt]

```

附录 4: LLM 评测提示语

Python

```

pigaiPrompt=f'''
你是一位组合数学课的老师，请根据参考答案和打分规则来给学生的回答打
分。我将给你问题，参考答案和打分规则，学生回答，请输出一个分数。例如：
问题：

$$\begin{cases} \text{exercise} \end{cases}$$
 设 $\{a_i\}$ 是一长度为
 $60$ 的正整数列（ $1 \leq i \leq 60$ ），其中 $a_1 = 1$ ，
 $a_{60} = 2^{20}$ ，并且数列中每一项都是其后面一项的约数。 %
 $\begin{cases} \text{enumerate} \end{cases}$  [label={\rm (\arabic*)}] \item 求满足要求
的数列 $\{a_i\}$ 的数目； \item 若进一步要求
 $\{a_i\}$ 严格递增，求满足条件的序列数目。
 $\end{cases}$  \end{cases}

参考答案：
"(1)": [
    {
        "answer": " $2^{20} = 2^{40} \cdot 5^{20}$ 。
 $\begin{cases} \text{enumerate} \end{cases}$  [label={\rm (\arabic*)}] \item 每一项中含
有 $2$ 的幂次和 $5$ 的幂次分别单调递增。以 $2$ 的幂次为例，可想象成是走 $58$ 
 $\times 40$ 的格路，路径在 $x = i - 2$ 处的最大 $y$ 坐标即为 $a_i$ 中 $2$ 的幂
次；对于 $5$ 也类似。总的方案数为 $\binom{98}{40}$ 
 $\binom{78}{20}$ ；",
        "rules": [

```

```

        {{
        "rule": "分解 $20^{\{20\}}$ ",
        "score": 1
        }},
        {{
        "rule": "答案正确",
        "score": 3
        }}
    ]
    }}
],
"(2)": [
    {{
        "answer": "\\item 讨论数列
 $\\left\\{\\frac{a_{i+1}}{a_i}\\right\\}$  ( $1 \\leq i \\leq 59$ ), 该数列的组成有三种情况: % \\begin{itemize} \\item
 $38$ 个 $2$ 、 $20$ 个 $5$ 、 $1$ 个 $4$ , 有 $59 \\binom{58}{20}$ 种情况:
\\item  $39$ 个 $2$ 、 $19$ 个 $5$ 、 $1$ 个 $10$ , 有 $59 \\binom{58}{19}$ 种情况; \\item  $40$ 个 $2$ 、 $18$ 个 $5$ 、 $1$ 个
 $25$ , 有 $59 \\binom{58}{18}$ 种情况。 \\end{itemize} % 因此, 总
共有 $59 \\left[ \\binom{58}{18} + \\binom{58}{19} + \\binom{58}{20} \\right]$ 种情况。",
        "rules": [
            {{
            "rule": "讨论数列形式",
            "score": 1
            }},
            {{
            "rule": "情况 1 正确",
            "score": 1
            }},
            {{
            "rule": "情况 2 正确",
            "score": 1
            }},
            {{
            "rule": "情况 3 正确",
            "score": 1
            }},
            {{
            "rule": "答案正确",
            "score": 1
            }}
        ]
    }}
]

```

```
]
  }}
]
```

学生回答:

问题分析

我们有一个长度为 60 的正整数列 $\{a_i\}$, 其中 $1 \leq i \leq 60$, 满足以下条件:

1. $a_1 = 1$
2. $a_{60} = 20^{20}$
3. 对于每一个 i ($1 \leq i \leq 59$), a_i 是 a_{i+1} 的约数。

我们需要解决两个问题:

1. 求满足上述条件的数列 $\{a_i\}$ 的数目。
2. 如果进一步要求数列 $\{a_i\}$ 严格递增, 求满足条件的序列数目。

问题 1: 求满足要求的数列 $\{a_i\}$ 的数目

思路

首先, 我们注意到 $a_1 = 1$, 而 a_i 是 a_{i+1} 的约数。这意味着对于每一个 i , a_{i+1} 必须是 a_i 的倍数。因此, 数列 $\{a_i\}$ 是一个非递减的序列, 且每一项都是前一项的倍数。

为了简化问题, 我们可以考虑将数列 $\{a_i\}$ 看作是从 1 到 20^{20} 的一条路径, 其中每一步都是从前一项乘以一个正整数得到的。

具体步骤

1. **确定每一步的倍数**: 对于每一个 i ($1 \leq i \leq 59$), $a_{i+1} = a_i \times k_i$, 其中 k_i 是一个正整数。
2. **总倍数的确定**: 由于 $a_{60} = 1 \times k_1 \times k_2 \times \dots \times k_{59} = 20^{20}$, 所以 $k_1 \times k_2 \times \dots \times k_{59} = 20^{20}$ 。
3. **分解质因数**: 将 20^{20} 分解质因数, 得到 $20^{20} = (2^2 \times 5)^{20} = 2^{40} \times 5^{20}$ 。
4. **分配质因数**: 我们需要将 2^{40} 和 5^{20} 分配到 k_1, k_2, \dots, k_{59} 中。

计算方法

对于每一个质因数，我们可以独立地分配其指数。具体来说：

- 对于质因数 2 ，我们需要将 40 个 2 分配到 59 个 k_i 中。这相当于将 40 个无区别的球放入 59 个有区别的盒子中，允许空盒。这是一个经典的组合问题，可以用“星与条”方法计算，其方案数为 $\binom{40 + 59 - 1}{59 - 1} = \binom{98}{58}$ 。
- 对于质因数 5 ，我们需要将 20 个 5 分配到 59 个 k_i 中。同样地，方案数为 $\binom{20 + 59 - 1}{59 - 1} = \binom{78}{58}$ 。

由于这两个分配是独立的，总的方案数为 $\binom{98}{58} \times \binom{78}{58}$ 。

问题 2：若进一步要求 $\{a_i\}$ 严格递增，求满足条件的序列数目

思路

如果数列 $\{a_i\}$ 严格递增，那么每一项 a_{i+1} 必须是 a_i 的严格倍数，即 $a_{i+1} = a_i \times k_i$ ，其中 $k_i \geq 2$ 。

具体步骤

1. **确定每一步的倍数**：对于每一个 i ($1 \leq i \leq 59$)， $a_{i+1} = a_i \times k_i$ ，其中 $k_i \geq 2$ 。
2. **总倍数的确定**：由于 $a_{60} = 1 \times k_1 \times k_2 \times \cdots \times k_{59} = 20^{\{20\}}$ ，所以 $k_1 \times k_2 \times \cdots \times k_{59} = 20^{\{20\}}$ 。
3. **分解质因数**：将 $20^{\{20\}}$ 分解质因数，得到 $20^{\{20\}} = 2^{\{40\}} \times 5^{\{20\}}$ 。
4. **分配质因数**：我们需要将 $2^{\{40\}}$ 和 $5^{\{20\}}$ 分配到 k_1, k_2, \dots, k_{59} 中，且每个 k_i 至少为 2 。

计算方法

对于每一个质因数，我们可以独立地分配其指数。具体来说：

- 对于质因数 2 ，我们需要将 40 个 2 分配到 59 个 k_i 中，且每个 k_i 至少包含 1 个 2 。这相当于将 $40 - 59 = -19$ 个 2 分配到 59 个盒子中，这是不可能的。因此，我们需要重新考虑分配方式。

实际上, 由于每个 k_i 至少为 2, 我们可以将 $2^{\{40\}}$ 看作是 $2^{\{40\}} = 2^{\{59\}} \times 2^{\{40 - 59\}} = 2^{\{59\}} \times 2^{\{-19\}}$ 。由于指数不能为负, 这意味着我们需要至少 59 个 2, 但 $40 < 59$, 所以这是不可能的。

因此, 不存在满足条件的严格递增数列。

最终答案

1. 满足要求的数列 $\{a_i\}$ 的数目为 $\binom{98}{58} \times \binom{78}{58}$ 。
2. 若进一步要求 $\{a_i\}$ 严格递增, 则不存在满足条件的序列。

那么, 由于这个例子中的问题和参考答案包含 2 个小题, 因此你需要逐个给每个小题打分。初始给分为 0。

先分析第 1 小题, 对于第 1 个批改规则 "rule": "分解 $20^{\{20\}}$ ", "score": 1, 由于学生回答中包含 "3. **分解质因数**: 将 $20^{\{20\}}$ 分解质因数, 得到 $20^{\{20\}} = (2^2 \times 5)^{20} = 2^{\{40\}} \times 5^{\{20\}}$ 。", 因此根据 "score": 1, 我们给 0 (之前总分) + 1 (该规则给分) = 1 (当前总分)。对于第 2 个批改规则 "rule": "答案正确", "score": 3, 我们找到参考答案中第 1 小题的最终答案是 " $\binom{98}{40} \times \binom{78}{20}$ ", 学生回答的最终答案是 " $\binom{98}{58} \times \binom{78}{58}$ ", 虽然他们形式上不一样, 但是经过一些组合数学的运算是一样的, 即 " $\binom{98}{58} \times \binom{78}{58} = \binom{98}{40} \times \binom{78}{20}$ ", 因此答案正确, 我们给 1 (之前总分) + 3 (该规则给分) = 4 (当前总分)。

再分析第 2 小题, 对于第 1 个批改规则 "rule": "讨论数列形式", "score": 1, 根据参考答案中的 "该数列的组成有三种情况" 可知需要将数列分为三种情况讨论, 但是学生回答中没有分成三种情况讨论, 因此我们给 4 (之前总分) + 0 (不符合该规则, 不给分) = 4 (当前总分)。对于第 2 个批改规则 "rule": "情况 1 正确", "score": 1, 由于学生回答中不包含情况 1, 因此我们给 4 (之前总分) + 0 (不符合该规则, 不给分) = 4 (当前总分)。对于第 3 个批改规则 "rule": "情况 2 正确", "score": 1, 由于学生回答中不包含情况 2, 因此我们给 4 (之前总分) + 0 (不符合该规则, 不给分) = 4 (当前总分)。对于第 4 个批改规则 "rule": "情况 3 正确", "score": 1, 由于学生回答中不包含情况 3, 因此我们给 4 (之前总分) + 0 (不符合该规则, 不给分) = 4 (当前总分)。对于第 5 个批改规则 "rule": "答案正确", "score": 1, 由于学生回答的答案是 "不存在满足条件的严格递增数列", 而参考答案的最终答案是 "总共有 $59 \times \left[\binom{58}{18} + \binom{58}{19} + \right]$ "。

`\\binom{{58}}{{20}} \\right]$种情况"`，因此我们给 4（之前总分）+ 0（不符合该规则，不给分）= 4（当前总分）。

因此总分是 4。

总分：

4

请根据以上例子，依据我下面给出的问题和参考答案给学生回答打分。请根据我提供的例子，首先输出你的打分分析过程，最终输出学生的分数。

问题：

`{problem}`

参考答案：

`{ref_answer}`

学生回答：

`{answer}`

请首先输出你的打分分析过程，最终输出学生的分数。你必须保证你输出的最后一部分遵循此格式：

总分：

分数

...