

NOI 2026 第一试《布丁》解题报告

NOI 2026 命题组

2026 年 7 月 20 日

解法一

- 对于测试点 1, w 只有 35 个可能的取值, 考虑用较简单的方式确定答案。
- 只需要询问一次 $[1, 2, 3, \dots, 35]$, 将 w 混入排序后相当于对原本的结果产生了 w 的贡献, 因此将询问结果减去 34 即为 w 的值。
- 对测试点 2, 3 也应用这个做法: 测试点 2 仅询问 $[1, 3000]$ 内所有质数, 测试点 3 询问 $[1, 3000]$ 中的所有数。得分为 $10 + \lfloor 3.66 \rfloor + \lfloor 3.9 \rfloor = 16$ 。
- 一个小优化是不询问最小的候选数, 但不改变得分。

如何合理地分配询问次数？

- 从第 5 次询问开始每次询问要丢掉 30% 的分数，惩罚很重。
- 考虑在第 4 次询问使用解法一，直接确定答案。
- 因此需要在前 3 次询问内尽可能减少 w 的候选取值个数。

约定

- 以下将一次包含 k 个数的询问称为代价 k 的询问。
- 若 p^d 整除 w 而 p^{d+1} 不整除 w ，则称 w 含有 d 个 p 。

解法二

- 一个比较自然的想法是利用询问给出的 $\gcd$ 的信息确定因数组成。

解法二

- 一个比较自然的想法是利用询问给出的 gcd 的信息确定因数组成。
- 可以用一次代价 1 的询问确定 w 是否为某个质数 p 的倍数，
- 进一步地，如果 k 个质数 $p_1, p_2, \dots, p_k$ 的乘积不大于询问参数上界 4500，那么可以用一次代价 1 的询问确定 w 含有其中哪些质因子。
- 更深入地，还可以通过提高次数来确定 w 包含的某个质因子的数量。
- 单从削减取值个数的角度考虑，最坏情况下，确定 w 不是 p_i 的倍数可以将取值个数削减至原本的 $\frac{p_i - 1}{p_i}$ 。

解法二

■ 若在前三次询问时分别选择

- $2 \times 3 \times 5 \times 7 \times 11 = 2310$;
- $13 \times 17 \times 19 = 4199$;
- $23 \times 29 = 667$,

则得分为 $10 + \lfloor 3.66 \rfloor + \lfloor 12.5 \rfloor = 25$ 。

解法二

- 若在前三次询问时分别选择

- $2 \times 3 \times 5 \times 7 \times 11 = 2310;$

- $13 \times 17 \times 19 = 4199;$

- $23 \times 29 = 667,$

则得分为 $10 + \lfloor 3.66 \rfloor + \lfloor 12.5 \rfloor = 25$ 。

- 注意到这个做法在测试点 2 上的表现很差，因此要考虑另外的做法。

解法三

- 考虑询问给出除 $\gcd$ 以外的信息。
- 注意到询问的流程中有这样一步：将 w 插入并按升序排序。
- 这会给出 w 的位置信息，从而可以将 w 的取值上下界压缩。

解法三

- 考虑询问给出除 $\gcd$ 以外的信息。
- 注意到询问的流程中有这样一步：将 w 插入并按升序排序。
- 这会给出 w 的位置信息，从而可以将 w 的取值上下界压缩。
- 具体地，可以通过设计询问序列，使得能够根据将 w 插入后对 $\gcd$ 的扰动确定 w 被插入的位置。
- 以下是三种可能的设计询问序列的方式。

解法三 A

解法三 A

- 首先确定 w 包含 2 的个数 d 。
- 先尝试将取值范围折半。考虑询问 $[2^k, 4096]$ ($k > d$):

解法三 A

- 首先确定 w 包含 2 的个数 d 。
- 先尝试将取值范围折半。考虑询问 $[2^k, 4096]$ ($k > d$):
 - 若 $w \leq 2^k$, 则最终结果增加 2^d ;
 - 若 $w > 2^k$, 则最终结果增加 $2^{d+1} - 2^k$ 。

解法三 A

- 首先确定 w 包含 2 的个数 d 。
- 先尝试将取值范围折半。考虑询问 $[2^k, 4096]$ ($k > d$):
 - 若 $w \leq 2^k$, 则最终结果增加 2^d ;
 - 若 $w > 2^k$, 则最终结果增加 $2^{d+1} - 2^k$ 。
 - 注意到后者是 2^{d+1} 的倍数而前者不是, 因此可以方便地推断 w 的落点。

解法三 A

- 首先确定 w 包含 2 的个数 d 。
- 先尝试将取值范围折半。考虑询问 $[2^k, 4096]$ ($k > d$):
 - 若 $w \leq 2^k$, 则最终结果增加 2^d ;
 - 若 $w > 2^k$, 则最终结果增加 $2^{d+1} - 2^k$ 。
 - 注意到后者是 2^{d+1} 的倍数而前者不是, 因此可以方便地推断 w 的落点。
- 然而候选区间的中点并不总是 2 的次幂, 因此这个时候可以乘上一个质数 p 以贴近候选区间的中点 (需要额外细节检查 w 是 p 的倍数时的情况)。
- 第 1 次询问可以通过类似的方式在确定因数构成的同时将候选区间折半。

解法三 A

- 若第 1 次直接询问 4096 而不做优化，第 2, 3 次询问操作如上，则得分为 $10 + \lfloor 10.2 \rfloor + \lfloor 18.9 \rfloor = 38$ 。

解法三 A

- 若第 1 次直接询问 4096 而不做优化，第 2, 3 次询问操作如上，则得分为 $10 + \lfloor 10.2 \rfloor + \lfloor 18.9 \rfloor = 38$ 。
- 若第 1 次询问 $[3360]$ ($3360 = 2^5 \times 3 \times 5 \times 7$)，确定 2, 3, 5, 7 的因数信息，则得分提升至 $10 + \lfloor 8.8 \rfloor + \lfloor 30.8 \rfloor = 48$ 。

解法三 A

- 若第 1 次直接询问 4096 而不做优化, 第 2, 3 次询问操作如上, 则得分为 $10 + \lfloor 10.2 \rfloor + \lfloor 18.9 \rfloor = 38$ 。
- 若第 1 次询问 $[3360]$ ($3360 = 2^5 \times 3 \times 5 \times 7$), 确定 2, 3, 5, 7 的因数信息, 则得分提升至 $10 + \lfloor 8.8 \rfloor + \lfloor 30.8 \rfloor = 48$ 。
- 若第 1 次询问 $[1680, 3360]$, 其他同上, 折半取值范围, 则得分可进一步提升至 $10 + \lfloor 11.2 \rfloor + \lfloor 39.2 \rfloor = 60$ 。

解法三 B

解法三 B

- 当然，如果想更进一步的话，每次仅折半取值范围是不够的，需要更加精细的划分。

解法三 B

- 当然，如果想更进一步的话，每次仅折半取值范围是不够的，需要更加精细的划分。
- 延续解法三 A 的思路，如果想要将取值区间分为尽可能均匀的多个小段的话，需要将询问序列中的每个数尽可能贴近取值区间的等分点。
- 考虑序列 $[1, 2, 4, \dots, 2^k]$ ，分别乘上不同的质数后对齐至等分点，这样可以根据扰动的结果确定 w 的落点。

解法三 B

- 当然，如果想更进一步的话，每次仅折半取值范围是不够的，需要更加精细的划分。
- 延续解法三 A 的思路，如果想要将取值区间分为尽可能均匀的多个小段的话，需要将询问序列中的每个数尽可能贴近取值区间的等分点。
- 考虑序列 $[1, 2, 4, \dots, 2^k]$ ，分别乘上不同的质数后对齐至等分点，这样可以根据扰动的结果确定 w 的落点。
- 不过实现上需要注意以下的细节：
 - 如果 w 有 d 个 2，则询问序列需要跳过 2^d 。
 - 乘上质数后扰动可能不再唯一，需要仔细把控（选择不会导致扰动重复的质数，或者单独处理结果不唯一的情况）。

解法三 B

- 这一部分视实现情况可以获得不同的分数：

解法三 B

- 这一部分视实现情况可以获得不同的分数：
 - 第 1 次询问不做优化可以获得 55 ~ 80 分；
 - 第 1 次询问优化为 [3360] 可以获得 65 ~ 95 分；
 - 第 1 次询问进一步优化为 [1680, 3360] 可以获得 75 ~ 100 分。

解法三 B

- 这一部分视实现情况可以获得不同的分数：
 - 第 1 次询问不做优化可以获得 55 ~ 80 分；
 - 第 1 次询问优化为 [3360] 可以获得 65 ~ 95 分；
 - 第 1 次询问进一步优化为 [1680, 3360] 可以获得 75 ~ 100 分。
 - 在每个得分区间内的分数取决于你构造的询问序列的精细程度。每次划分越细越均匀则表现通常越好。
 - 标准解法第 2 次询问将取值区间 7 等分，第 3 次询问 3 等分。
 - 事实上，可以进一步改进至 $S \leq 34$ 或更优。

解法三 C

解法三 C

- 这个解法构造的询问序列与 A、B 不同，主要利用了连续正整数的乘积。

解法三 C

- 这个解法构造的询问序列与 A、B 不同，主要利用了连续正整数的乘积。
- 选定左端点 l ，连乘个数 d 与序列长度 k ，直接询问

$$\left[l^{\overline{d}}, (l+1)^{\overline{d}}, \dots, (l+k-1)^{\overline{d}} \right],$$

其中 $n^{\overline{m}} = n \times (n+1) \times \cdots \times (n+m-1)$ ，然后逐个判断候选数会落在哪个区间。

解法三 C

- 这个解法构造的询问序列与 A、B 不同，主要利用了连续正整数的乘积。
- 选定左端点 l ，连乘个数 d 与序列长度 k ，直接询问

$$\left[l^{\overline{d}}, (l+1)^{\overline{d}}, \dots, (l+k-1)^{\overline{d}} \right],$$

其中 $n^{\overline{m}} = n \times (n+1) \times \dots \times (n+m-1)$ ，然后逐个判断候选数会落在哪个区间。

- 这样做的好处是，不需要在第 1 次去确定 w 的因数构成，前三次询问都可以使用这个方法。
- 在前三次询问中分别选定 $k = 5, 3, 2$ ，然后合理选取 l 与 d ，可以获得 98 分。
- 这个做法细节较少，实现起来相对简单，但可能不那么直观。