2025 湖南大学程序设计竞赛 题解

lvle TrainerMarvin Vistarin zmy123456

Hunan University

2025 年 6 月 29 日

Overview

预期难度

Easy:	C, H, F, J
Medium:	E, D, I
Medium-Hard:	A, B
Hard:	G, K
Anti-AK:	L

Overview

实际难度 - 线下

Easy:	C, J
Medium:	F, E, H, D
Medium-Hard:	I
Hard:	A, G, B, K
Anti-AK:	L

实际难度 - 线上

Easy:	C, J, F
Medium:	D, H, E
Medium-Hard:	I, A, G
Hard:	B, K
Anti-AK:	L

C - Echoes of the Runes

题意

给定一个长度为 n 的数组，判断能否使用至多一次交换使得数组升序。

数据范围

$1 \le n \le 1000,\ 1 \le a_i \le 10^9.$

C - Echoes of the Runes

做法 - $O(n \log n)$

将数组升序排序，统计排序后数组和原数组的差异。

不难发现，对有序数组的一次交换会产生两个差异。

若差异数小于等于 2，输出 `Sorted`。否则输出 `Failed`。

时间复杂度 $O(n \log n)$。

C - Echoes of the Runes

花絮

作为签到题，本题放宽了数据范围，$O(n^3)$ 做法也可以通过。

做法 - $O(n^3)$

暴力地交换每一对元素。每次交换暴力检查数组是否有序。

时间复杂度 $O(n^3)$。

H - Simai

题意

给出一些圆内线条，求它们的长度。

数据范围

$1 \leq T \leq 10^5$.

H - Simai

做法

模拟题。按题意模拟即可。

相对麻烦的是计算 $x - y$。圆心角 α 对应的弦长为 $2r \cdot \sin(\alpha/2)$，所以可使用 `std::sin()` 等库函数方便地求得长度。

F - Portal

题意

给定 n 个节点，第 i 个节点需要和自身或其他节点连边 a_i 次。问 1 号点是否一定能到达 n 号点。

数据范围

$1 \le n \le 10^5$，$1 \le a_i \le 10^9$，$\sum_{i=1}^{n} a_i$ 为偶数。

F - Portal

做法

首先特判：$n=1$ 时答案为 Yes。

然后，若数组 a 中只有 a_1 和 a_n 为奇数，则答案为 Yes；反之，答案为 No。时间复杂度 $O(n)$。

证明

由于一条边连接两个点，所以一个连通块的 $\sum a$ 一定是偶数。

若数组 a 中只有 a_1 和 a_n 为奇数，考虑 a_1 所在连通块，不难发现：只要不向 a_n 连边，就不可能满足奇偶性，故答案为 Yes。

反之，我们令每个节点优先向自身连边。此时若 a_1 为偶数，答案显然为 No；若 a_1 为奇数，必然可以找到另一个非 a_n 的奇数与之连接，答案同样为 No。

J - Uniform Random Descent Process

题意

输入 n，问期望执行多少次 $n = \text{rand}(0, n-1)$ 后，n 变为 0。

数据范围

$1 \leq T \leq 10^5$，$0 \leq n \leq 10^5$.

J - Uniform Random Descent Process

做法 - 期望 DP

设 f_i 为 i 期望执行多少次操作，则

$$f_i = \sum_{j=0}^{i-1} \frac{f_j}{i} + 1$$

可以记录 f_i 前缀和来实现 $O(1)$ 转移。$O(n)$ 预处理后每组数据 $O(1)$ 回答询问。

J - Uniform Random Descent Process

做法 - 调和级数

事实上，手玩一下前几项，可以发现答案是调和级数 $\sum_{i=1}^{n} \frac{1}{i}$。

由期望 DP 的转移式，可以归纳得出这一结论。

更直观的解释是：i 有 $\frac{1}{i}$ 的概率随机到 $i-1$，此时比 $i-1$ 多一次操作；在剩下 $1-\frac{1}{i}$ 的情况下，与 $i-1$ 等价。故期望比 $i-1$ 多 $\frac{1}{i}$ 次操作。

同样地，可以 $O(n)$ 预处理并 $O(1)$ 回答。

E - Neuro's New Game

题意

给定一个长度为 n 的 01 串（记为 s），构造一个长度为 n 的排列 a，满足：

- 若 $s_i = 0$, 则 $a_i < a_{(i+1)\%n}$；
- 若 $s_i = 1$, 则 $a_i > a_{(i+1)\%n}$。

如果不存在这样的 a，则输出 -1。

数据范围

$1 \leq n \leq 2 \cdot 10^5$.

E - Neuro's New Game

做法 - 拓扑排序

本题做法很多，比较直观的是使用拓扑排序。

注意到，将小于关系视作入边，将大于关系视作出边，对连边后形成的图做一次拓扑排序，即可得到一段符合条件的排列。

图中有环时无解。从图的拓扑关系进一步观察到，当且仅当 s 全为 0 和 1 时才会成环。

E - Neuro's New Game

做法 - 离散化

一种更为暴力的做法是：

- 当 $s_i = 0$ 时，令 $a_i := a_{i-1} - 1$；
- 当 $s_i = 1$ 时，令 $a_i := a_{i-1} + 10^9$。

不难发现这样构造出的 a_i 一定不会重复。

构造完成后，将数据离散化即可得到答案。

D - Keine's Prefix Sum

题意

定义数组 a 的前缀和 $s_i = \sum_{j=1}^{i} a_j$。如果所有 $s_i \geq 0$，则称数组 a 为 **good** 数组。

对数组 a 进行 n 次循环左移操作。输出 n 次操作中 **good** 数组的总数。

数据范围

$1 \leq n \leq 10^5$, $-10^3 \leq a_i \leq 10^3$.

D - Keine's Prefix Sum

做法

给定数组 $[a_1, \ldots, a_n]$。令 S 为前缀和，P 为前缀和的前缀 min，Q 为前缀和的后缀 min。即

- $S_i = \sum_{j=1}^{i} a_j$
- $P_i = \min_{0 \le j \le i} S_j$
- $Q_i = \min_{i \le j \le n} S_j \; (Q_{n+1} = +\infty)$

将数组左移 k 次后，数组变为 $[a_{k+1}, \ldots, a_n, a_1, \ldots, a_k]$。

可以发现，它是 **good** 数组当且仅当

$$Q_{k+1} \ge S_k \text{ 且 } P_k \ge S_k - S_n.$$

预处理 P, Q 用时 $O(n)$，每次检查 $O(1)$，总时间复杂度 $O(n)$。

I - The Dream-Quest

题意

给你一棵根为 1，大小为 n 的树。初始时刻，Vistar 在树根，每个叶子上有一个警察。

一个回合内，Vistar 可以选择移动到相邻节点（或保持静止），然后所有不在根节点的警察向根节点移动一次。如果任何时候 Vistar 与警察在同一节点，则被抓住。

判断 n 回合后 Vistar 是否可能未被抓住。如果可能，输出任意一个行动方案。

数据范围

$1 \leq n \leq 2000.$

I - The Dream-Quest

做法

本题有多种做法。一种简单的做法是建分层图并 DFS。

设当前位置为 u，时间为 t，相邻节点为 v，则 (u,t) 转移到 $(v,t+1)$，当且仅当 t 时刻和 $t+1$ 时刻中 v 节点都没有警察。

我们可以 $O(n^2)$ 预处理出每个时刻警察的位置。预处理后，按照限制进行 DFS，同时使用栈维护行动的路径。如果进入一个 $t=n$ 的状态，就说明此时的方案合法。

时间复杂度 $O(n^2)$。

A - A Slide B Slide

题意

给定两个数组 $a = [a_1, a_2, \ldots, a_n]$ 和 $b = [b_1, b_2, \ldots, b_m]$。

你需要选择 a 的一个子序列 $[a_{c_1}, a_{c_2}, \ldots, a_{c_p}]$ 和 b 的一个子序列 $[b_{d_1}, b_{d_2}, \ldots, b_{d_q}]$，满足以下条件：

- $c_1 \le 2$, $d_1 \le 2$;
- $\forall i \ge 2$, $c_i - c_{i-1} \le 2$, $d_i - d_{i-1} \le 2$;
- $c_p = n$, $d_q = m$.

问：是否无论如何选择 $[c_1, c_2, \ldots, c_p]$，都存在 $[d_1, d_2, \ldots, d_q]$，使得 $[b_{d_1}, b_{d_2}, \ldots, b_{d_q}]$ 是 $[a_{c_1}, a_{c_2}, \ldots, a_{c_p}]$ 的子序列。

数据范围

$1 \le n, m \le 10^5$, $1 \le a_i, b_i \le 10^9$.

A - A Slide B Slide

做法

无论 a 选什么，都必须选择一个 b 的子序列与其匹配。

所以，对于每一个 a_i，我们应该贪心地考虑 **最坏情况下** b 匹配到的最远位置。

可以证明，这样不会错过最优解。

做法

考虑一个基于贪心的 DP。

设 dp_i 为：选择 a_i 时，贪心策略下 b 能够匹配的最远位置。若最终 $dp_n = m$，则说明总可以完成匹配。转移如下：

首先

$$dp_i = \min(dp_{i-1}, dp_{i-2}),\ k := dp_i$$

然后有

$$dp_i = \begin{cases} k+2, & \text{if } a_i = b_{k+2} \\ k+1, & \text{else if } a_i = b_{k+1} \\ k, & \text{otherwise} \end{cases}$$

时间复杂度 $O(n+m)$。

B - Bricked Blast Furnaces

题意

使用不超过 $3 \cdot 10^5$ 个耐火砖 (F)，构建出 n 个砖高炉结构。耐火砖 (F) 可以共用。

砖高炉结构如下图所示：

F	F	F
F	L	F
F	B	F

F	F	F
F	L	B
F	F	F

F	B	F
F	L	F
F	F	F

F	F	F
B	L	F
F	F	F

Not even two BBFs are fast enough...Can you figure out how to save the most bricks making 4?

PS: If you can't, this pack may not be for you.

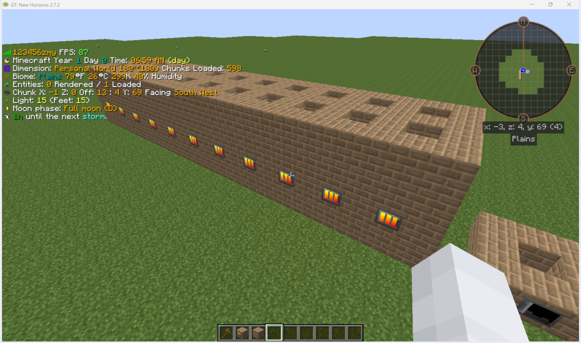
数据范围

$1 \leq n \leq 10^5$.

B - Bricked Blast Furnaces

做法

最简单的构造方法是摆成两排，但是显然，这样不能通过此题（需要 $3 \cdot 10^5 + 5$ 个）。



B - Bricked Blast Furnaces

做法

考虑将构造从两排扩展成多排。如果再密铺一些就会非法。

F	F	F	F	F	F	F
F	L	F	L	F	L	F
F	B	F	B	F	B	F
F	L	F	L	F	L	F
F	B	F	B	F	B	F
F	L	F	L	F	L	F
F	B	F	B	F	B	F

问题在于，砖高炉的主方块 (B) 会占用其它结构里耐火砖 (F) 的位置。

B - Bricked Blast Furnaces

做法

我们可以去掉一些砖高炉结构（得到一些空位），并将旁边结构的主方块 (B) 放在一个十字的四条边上。

F	F	F	F	F	F	F
F	L	F	L	F	L	F
F	B	F	B	F	B	F
F	L	F		F	L	F
F	B	F	F	F	B	F
F	L	F	L	F	L	F
F	B	F	B	F	B	F

$\rightarrow$

F	F	F	F	F	B	F
B	L	F	L	F	L	F
F	F	F	B	F	F	F
F	L	B		B	L	F
F	F	F	B	F	F	F
F	L	F	L	F	L	B
F	B	F	F	F	F	F

这样构造后，对于每个砖高炉结构，(B) 的旁边必须存在至少一个空位。我们需要最小化空位数量。

B - Bricked Blast Furnaces

做法

考虑如何在二维平面上选择这些空位。

构造方法比较简单，直接按照日字形放空位即可。注意：为展示方便，下图将距离缩短至实际的一半。

X					X	
		X				
				X		
	X					X
			X			
X					X	
		X				

B - Bricked Blast Furnaces

做法

下图更加直观地呈现了 (F) 的分布。

```
. . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . F F F . . . . . . . . . . . .
. . . . . . . . . . F . F . . . . . . . . . . . .
. . . . F F F . F F F . F F F F F . . . . . . . .
. . . . F . F . F . .   . . F . F . . . . . . . .
. . F F F . F F F F F . F F F . F F F F F . . . .
. . F . .   . . F . F . F . .   . . F . F . . . .
. . F F F . F F F . F F F F F . F F F . F F F . .
. . . . F . F . .   . . F . F . F . .   . . F . .
. . . . F F F F F . F F F . F F F F F . F F F . .
. . . . . . . . F . F . .   . . F . F . F . . . .
. . . . . . . . F F F F F . F F F . F F F . . . .
. . . . . . . . . . . . F . F . . . . . . . . . .
. . . . . . . . . . . . F F F . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . .
. . . . . . . . . . . . . . . . . . . . . . . . .
```

B - Bricked Blast Furnaces

证明

假设初始放置 $x \cdot y$ 个砖高炉结构，那么至多需要 $k = \lfloor \frac{xy}{5} \rfloor \leq \frac{xy}{5}$ 个空位。

砖高炉结构的数量为 $xy - k$，(F) 的数量为 $(2x+1) \cdot (2y+1) - 2xy + k = 2xy + k + 2x + 2y + 1$。

由于

$$\frac{2xy + k + 2x + 2y + 1}{xy - k} = \frac{3xy + 2x + 2y + 1}{xy - k} - 1$$
$$\leq \frac{11}{4} + \frac{10x + 10y + 5}{4xy}$$

我们知道一个砖高炉结构大约使用 $\frac{11}{4}$ 个 (F)，故该构造合法。

G - Portal 2

题意

给定 n 个节点，第 i 个节点需要和其他节点连边 a_i 次。问 1 号点是否一定能到达 n 号点。

数据范围

$2 \leq n \leq 10^5$，$1 \leq a_i \leq 5000$，$\sum_{i=1}^{n} a_i$ 为偶数。

G - Portal 2

做法

如果集合 $\mathbb{S}$ 存在一种合法配对方案，我们就称 $\mathbb{S}$ 是合法的。

引理 1

$\mathbb{S}$ 合法 $\Leftrightarrow$ $\mathrm{sum}(\mathbb{S})$ 为偶数且 $\mathrm{sum}(\mathbb{S}) - 2\max(\mathbb{S}) \geq 0$。

G - Portal 2

引理 1

$\mathbb{S}$ 合法 $\Leftrightarrow$ $\mathrm{sum}(\mathbb{S})$ 为偶数且 $\mathrm{sum}(\mathbb{S}) - 2\max(\mathbb{S}) \geq 0$。

证明 - 引理 1

右侧条件必要性显然。下面只证明充分性。

我们将不断采取以下构造步骤，直至配对完成：

- 选取当前集合中最大的两个非 0 元素 a_1 和 a_2。
- 在二者之间配对一次。这可以等价为 $a_1 := a_1 - 1,\ a_2 := a_2 - 1$。

可以证明，这样的构造一定能完成配对。故引理 1 得证。

G - Portal 2

做法

可以发现，如果原数组能被划分成若干个合法集合，且 1 和 n 不在同一集合中，答案即为 No。反之即为 Yes。

我们称这样的划分为 **合法划分**。

引理 2

若存在合法划分，则必存在一种合法划分，使得数组恰好被划分为两个集合。

G - Portal 2

引理 2

若存在合法划分，则必存在一种合法划分，使得数组恰好被划分为两个集合。

证明 - 引理 2

只需证明将两个合法集合合并，得到的依然是合法集合。

设 $\mathbb{S}$ 和 $\mathbb{T}$ 为两个合法集合。则

$$\operatorname{sum}(\mathbb{S}+\mathbb{T}) \geq 2\max(\mathbb{S}) + 2\max(\mathbb{T}) \geq 2\max(\mathbb{S}+\mathbb{T})$$

故引理 2 得证。

G - Portal 2

做法

由引理 2，可以使用 01 背包检查合法划分的存在性。

设 $f_{0/1,i,j}$ 表示：总和为偶数/奇数，考虑前 i 个点，总和为 j 时是否可行。我们将 a_2 至 a_{n-1} 排序，并钦定最大值在哪个集合中。

不妨设最大值和 a_n 在同一集合。定义：

- $\mathbb{A} = [2 \cdot \max(a_1, a_i),\ \sum_{i=1}^{n} a_i - 2 \cdot \max_{i=1}^{n}(a_i)]$
- $\mathbb{B} = \{\max(a_1, a_i),\ \sum_{i=1}^{n} a_i - \max_{i=1}^{n}(a_i)\}$

只需检查以下条件是否成立：

$$\exists j \in \mathbb{A} - \mathbb{B},\ f_{0,i,j} = 1$$

G - Portal 2

做法

朴素背包会超时，使用 bitset 可以优化到 $O\left(\frac{n\cdot\sum_{i=1}^{n} a_i}{\omega}\right)$，但仍然无法通过。

然而，本题有如下性质：

引理 3

数组中只有 a_1 和 a_n 为奇数时，答案为 Yes；

否则，$\sum_{i=1}^{n} a_i \geq 6\max(a)$ 时，答案一定为 No。

G - Portal 2

做法

由引理 3，背包的值域只需开到 $O(\max(a))$。

时间复杂度 $O\left(\frac{n\cdot\max(a)}{\omega}\right)$。

花絮

上述做法足以通过本题，不过还可以进一步常数优化：

- 若在值域 $\leq 4\max(a)$ 内未发现合法划分，则答案为 Yes。反之为 No。
- 或者，在上一个性质的基础上只跑 $3\max(a)$，也能得到正确答案。

G - Portal 2

证明 - 引理 3

将 a_2 至 a_{n-1} 降序排序，得到的新数组记作 b。我们取 b_1, b_2 分入集合 $\mathbb{S}$，此时有

$$|\mathrm{sum}(\mathbb{S}) - 2\max(\mathbb{S})| = |b_1 - b_2|$$

考虑整数 j 为最小的、满足 $\sum_{i=3}^{j} b_i \geq |b_1 - b_2|$ 的整数。则

$$\sum_{i=3}^{j} b_i \leq |b_1 - b_2| + \min(b_1, b_2) \leq \max(a)$$

将 b_3 至 b_j 并入集合 $\mathbb{S}$，此时即有 $\mathrm{sum}(\mathbb{S}) - 2\max(\mathbb{S}) \geq 0$。

$\mathrm{sum}(\mathbb{S})$ 未必是偶数。是奇数时，我们再向其中加入一个奇数以满足奇偶性。不妨设此数为 b_k。

G - Portal 2

证明 - 引理 3

上述步骤完成后，有

$$\mathrm{sum}(\mathbb{S}) \le b_1 + b_2 + \max(a) + b_k \le 4\max(a)$$

将剩下的数分入集合 $\mathbb{T}$，$\mathrm{sum}(\mathbb{S}) + \mathrm{sum}(\mathbb{T}) \ge 6\max(a)$ 时，必有 $\mathrm{sum}(\mathbb{T}) - 2\max(\mathbb{T}) \ge 0$。容易验证边界无法取到，故性质得证。

不难发现 $\mathrm{sum}(\mathbb{T}) \le 4\max(a)$。故 DP 时只需做两次检查，每次检查到 $4\max(a)$ 。进一步的分讨可以优化到只检查 $3\max(a)$。

K - Yet Another Connecting Problem

题意

给定 n 个节点的有向图，节点编号从 1 到 n。连边情况如下：

- 如果节点 u 是一个完全平方数，则存在一条从 u 指向 $\sqrt{u}$ 的有向边。
- 如果 u 不是完全平方数且 $u \le n-3$，则存在一条从 u 指向 $u+3$ 的有向边。

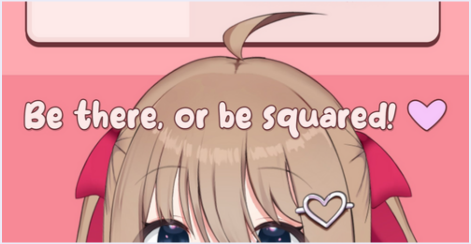
给定 q 个询问，每个询问包含一个目标节点 x。对于每个 x，计算图中有多少个节点能够到达 x。

数据范围

$1 \le T \le 100$, $1 \le n \le 10^{12}$, $1 \le q \le 10^6$.

K - Yet Another Connecting Problem

做法



Be **there**, or be **squared**!

K - Yet Another Connecting Problem

做法

首先形式化题意。让我们设起始点为 a_0，不考虑边界有

$$a_{i+1} = \begin{cases} \sqrt{a_i}, & a_i \text{ 是完全平方数} \\ a_i + 3, & \text{otherwise} \end{cases}$$

不难发现：

- $a_0 = 1$ 时，a_i 恒等于 1；
- $3 - 6 - 9 - 3$ 是一个循环。

K - Yet Another Connecting Problem

做法

排除 $a_0 = 1$ 的特例，分 $a_0 \equiv 0, 1, 2 \pmod 3$ 三种情况讨论。考虑 n 趋于无穷，可以证明：

引理 1

- $a_0 \equiv 0 \pmod 3$ 时，$\forall i \in \mathbb{N}, a_i \equiv 0 \pmod 3$；
- $a_0 \equiv 1 \pmod 3$ 时，$\exists i \in \mathbb{N}, a_i \equiv 2 \pmod 3$；
- $a_0 \equiv 2 \pmod 3$ 时，$\forall i \in \mathbb{N}, a_i \equiv 2 \pmod 3$；
- $3 - 6 - 9 - 3$ 是唯一可能的循环。

K - Yet Another Connecting Problem

做法

先处理单次询问。考虑这张图的反图，以 x 为起点在反图上搜索，能到达的点数即为答案。

设 l 为小于 x 且与 x 同余的最大完全平方数。x 可以直接到达 $(l, x]$ 内与 x 同余的点。注意：$x \equiv 2 \pmod 3$ 时，这样的完全平方数不存在，x 可直接到达 $[2, x]$。此时不妨定义 $l = -1$。

接下来，对区间内所有编号平方，并继续考虑这些点能到达哪些点。重复这一过程直到计算出所有贡献。

可以证明除 $3-6-9-3$ 外，贡献是独立的，可以直接加起来得到答案。

K - Yet Another Connecting Problem

做法

形式化上面的过程。设 $f(x)$ 为 x 的答案，有

$$f(x) = \frac{x-l}{3} + \sum_{i:\ i \equiv x \pmod 3,\ l < i \le x} f(i^2)$$

时间复杂度为 $O(\sqrt{n})$。

K - Yet Another Connecting Problem

做法

考虑多次询问，我们可以分段预处理 $\sqrt{n}$ 以内的答案：

$$f(i) = \begin{cases} f(3), & i \in \{6, 9\} \text{ 且 } n \geq 9 \\ f(i^2) + 1, & i - 3 \text{ 是完全平方数} \\ f(i-3) + f(i^2) + 1, & \text{otherwise} \end{cases}$$

预处理时，对 f 按拓扑序搜索，或者记忆化搜索，复杂度仍然是 $O(\sqrt{n})$。事实上，不做任何优化直接搜索，复杂度只退化到 $O(\sqrt{n}\log\log n)$，同样可以通过。

预处理后单次询问 $O(1)$。时间复杂度 $O\left(\sum(\sqrt{n}+q)\right)$，约 $1e7$。

K - Yet Another Connecting Problem

花絮

- 题目灵感来自 IMO 2017 Problem 1 和 Neuro-sama。
- 完整的证明较为繁琐。这里略过不表，留作读者练习。
- 本题有较为复杂的边界情况。可以考虑对拍小数据来检查。~~出题人和验题人都没有一遍写对本题的正解。~~
- 时限约为 std 的 6 倍。如果做法正确但是 TLE，很可能是因为实现不精细（例如：多一个 $\log$，过量使用 `double` 和 `std::sqrt()`，etc.）。

L - Yet Another Another Connecting Problem

题意

给定 n 个节点的有向图，节点编号从 1 到 n。连边情况如下：

- 如果节点 u 是一个完全平方数，则存在一条从 u 指向 $\sqrt{u}$ 的有向边。
- 如果 u 不是完全平方数且 $u \leq n-3$，则存在一条从 u 指向 $u+3$ 的有向边。

给定 q 个询问，每个询问包含一个目标节点 x。对于每个 x，计算图中有多少个节点能够到达 x。

数据范围

$1 \leq T \leq 10^5$，$1 \leq n \leq 10^{18}$，$1 \leq q \leq 10^6$.

L - Yet Another Another Connecting Problem

做法

设 $g(x)$ 为 x 本身和通过 x^2 开根到达 x 的点的数量，排除 $x \in \{1,3,6,9\}$ 之后:

$$g(x) = f(x^2) + [x \leq n]$$

$$f(x) = \sum_{i:\ i \equiv x \pmod 3,\ l < i \leq x} g(i)$$

L - Yet Another Another Connecting Problem

做法

考虑对 $g(x)$ 的 3 个同余类分别进行前缀和，记为:

- $s_0(x) = \sum_{i=0}^{x} g(3i) + C$;
- $s_1(x) = \sum_{i=0}^{x} g(3i+1) + C$;
- $s_2(x) = \sum_{i=0}^{x} g(3i+2) + C$.

令 $t = x\%3$，则 $f(x) = s_t(x) - s_t(l)$。

L - Yet Another Another Connecting Problem

做法

对 $x \equiv 0, 1, 2 \pmod 3$ 进行分讨，令 $k = \lfloor x/3 \rfloor$：

$$g(x) = f(x^2) + 1$$
$$= \begin{cases} s_0(3k^2) - s_0(3k^2 - 6k + 3) + [x \le n], & x = 3k \\ s_1(3k^2 + 2k) - s_1(3k^2 - 2k) + [x \le n], & x = 3k + 1 \\ s_1(3k^2 + 4k + 1) - s_1(3k^2 + 2k) + [x \le n], & x = 3k + 2 \end{cases}$$

可以从大到小依次求出 $g(x)$ 和 $s(x)$。

L - Yet Another Another Connecting Problem

做法

观察发现，$g(x)$ 和 $s(x)$ 可以表示为分段多项式函数。

例如：

$$g(3k) = \begin{cases} 1, & O(\sqrt{n}) \sim O(n) \\ 6k - 2, & O(n^{1/4}) \sim O(\sqrt{n}) \\ 108k^3 - 162k^2 + 114k - 29, & O(n^{1/8}) \sim O(n^{1/4}) \\ \dots \end{cases}$$

这可以使用数学归纳法证明。

L - Yet Another Another Connecting Problem

做法

注意特殊点：当一个点的求和区间跨过了两段不同的解析式时，需要单独计算。

对于每个 n，预处理出 $g(x)$ 和 $s(x)$ 分段的端点和特殊点的值，在询问时直接代入求值。

L - Yet Another Another Connecting Problem

实现及复杂度分析

选定一个阈值 K，假设使用多项式计算超过 $3K$ 的部分，剩余位置直接计算出数值。

此时，我们需要计算到约 $\log_K n$ 次的多项式。

因为每段的多项式系数和 n 无关，可以打表或者在最开始时插值 $O(\log_K^2 n)$ 计算得到。

L - Yet Another Another Connecting Problem

实现及复杂度分析

对于每个 n 需要计算：

- $3K$ 以内所有点的值：$O(K\log_K n)$；
- 每段的端点：依次开根得到，需要微调，$O(\log\log_K n)$；
- 特殊点的值：有 $\log\log n$ 个特殊点，总共需要计算的多项式次数之和为 $O(\log_K n)$。

询问时，代入解析式计算即可，$O(\log_K n)$。

时间复杂度 $O(\log_K^2 n + TK\log_K n + \sum q\log_K n)$，std 取 $K=5$ 时运行时间约 750ms。

空间复杂度 $O(K+\log_K n)$。

THANK YOU!