The 2024 ICPC ShaanXi Provincial Contest

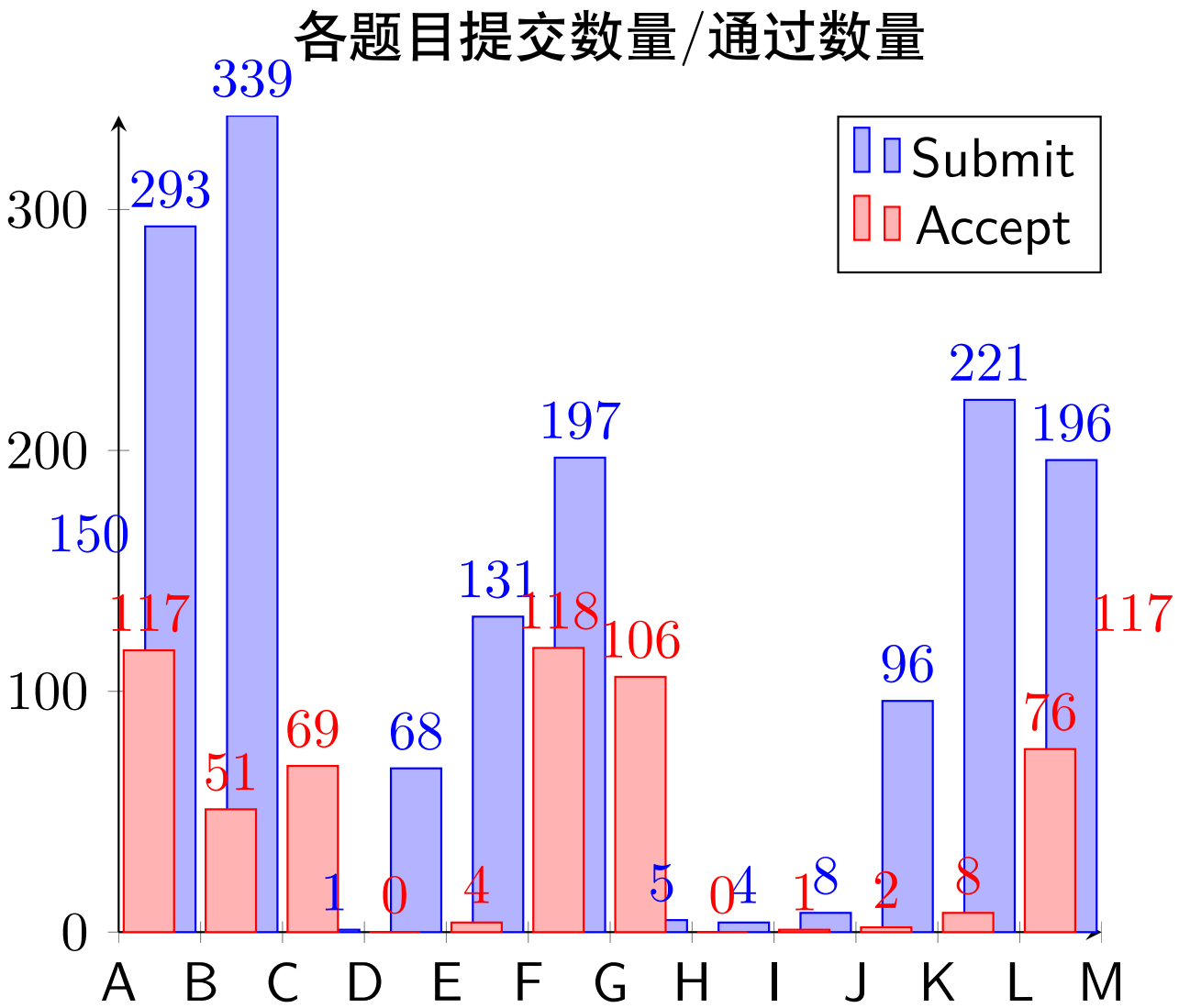
西安交通大学程序设计竞赛校队

2024.6.2

预期题目难度顺序

- 签到题：AFM
- 铜牌题：GCL
- 银牌题：BE
- 金牌题：DK
- 捧杯：IJ
- 防 AK 题：H

实际通过数量


各题目提交数量/通过数量
Submit
Accept
300
200
150
100
0
A
B
C
D
E
F
G
H
I
J
K
L
M
293
339
197
131
221
196
96
68
69
117
51
1
0
4
118
106
5
0
4
1
8
2
8
76
117

A Chmod

题意

给定 chmod 模式串，输出对应的权限字符串。

- Idea：Rhodoks
- Solution：Corycle
- Std & Data：Corycle
- Verifier：Luan_233，MCPlayer542

A Chmod

按照题意模拟即可。

F 写都写了，交一发吧

题意

给定一个长度为 n 的序列 g_i，求 $gi\&gj$，$1 \leq i, j \leq n$ 的最大值。

- Idea：MCPlayer542
- Solution：MCPlayer542
- Std & Data：MCPlayer542
- Verifier：Rhodoks，Lrefrain

F 写都写了，交一发吧

注意到代码可以重复提交，容易发现最优解为每次都提交得分最高的代码，得分即为单次提交所能获得的最高分。

遍历数组取得最大值输出即可。

M 窗花

题意

求 n 个整点坐标，对角线长为 2cm 且与坐标轴平行的正方形窗花的总覆盖面积。

- Idea：Rhodoks
- Solution：Rhodoks
- Std & Data：HCCH
- Verifier：Shirost，Rhodoks

M 窗花

解法 1:

考虑旋转 45°，则所有正方形各边均平行于坐标轴，可以很容易计数。

解法 2:

.
考虑把每个正方形拆分成四个小三角形，对每个窗花记录占用了哪些小三角形。

解法 3:

考虑按照 x 坐标从小到大，相同则 y 坐标从小到大排序依次加入正方形，注意到此时新加入的窗花只会和位于 $(x-1,y)$ 和 $(x,y-1)$ 的窗花重叠且 $(x-1,y)$ 和 $(x,y-1)$ 之间不互相重叠。减去重叠部分面积，计算实际的面积增量即可。

G 消失的数字

题意

1-9 中的一个数字（及包含其作为数位的数）消失了，问 $n(1 \leq n \leq 10^{18})$ 现在是第几个存在的自然数。

- Idea：Rhodoks
- Solution：Rhodoks
- Std & Data：uuku
- Verifier：Luan_233，Lrefrain

G 消失的数字

先考虑 $x = 9$ 的情况，我们有 8 之后是 10，188 之后是 200。不难发现，这其实就是‘逢九进一’，也就是说新的数列其实就是九进制数，那我们想求它是第几个其实就是把这个九进制数转换成十进制数。

那么再看 $x \neq 9$ 的情况，其实和上一种情况是一样的，把所有大于 x 的数位都减一是不改变数的顺序的，那我们将所有大于 x 的数位都减一后，新的数列也是九进制数，转成十进制即可。

要注意的是，自然数是包括 0 的，所以最后答案要加一。

C 换座位

题意

给出长为 n 数组 a_i, 你需要找到长为 n 的数组 b_i $(1 \le i \le n, 1 \le b_i \le 2n)$ 满足 $\forall i \ne j, b_i \ne b_j$ 且 $\forall i, b_i = i$ 或 $b_i = a_i$。最大化 $b_i = a_i$ 的个数并输出最大值。

- Idea：Rhodoks
- Solution：Rhodoks
- Std & Data：yyf_0404
- Verifier：Veritas，Shirost

C 换座位

考虑建图，i 向 a_i 连边，因为每个点至多有一条出边，所以这是一个内向基环树森林 + 树森林。

对于基环树，只有环上的点可以同时移动。

对于树，应该取最长的以树根为终点的路径，移动所有路径上的点。

考虑拓扑排序求最长路，如果发现是树，则答案加上最长路的边数。最后答案加上未入队的点数（也就是环上的总点数）。

时间复杂度 $O(n)$。

L 下棋

题意

初始有 x 个棋子，两个人轮流行动拿走棋子，拿走的棋子数量 y 要求满足在 k 进制下各位数的乘积是 y 的因子，寻找一个最小的 k 使得先手必胜。

- Idea：Luan_233，Rhodoks
- Solution：Luan_233
- Std & Data：Luan_233
- Verifier：Aquamoon，Veritas

L 下棋

注意到无论在何种进制下，y 一定满足最低位非 0。并且任何一个个位数 y 都是合法的。

类似于巴什博奕，注意到当先手拿到的 $(x)_k$ 末位为 0 时，无论怎么拿都会拿成末位非 0。所以策略很显然，后手只需要拿走个位上的数字个棋子就可以保证先手拿到的 $(x)_k$ 末位为 0。必败态 $(x=0)$ 必然存在于 x 末位为 0 的集合内。

所以末尾一位为 0 的时候先手必败，反之则必胜。那么令 $(x)_k$ 的末位不为 0，k 只需要取最小的非因子即可，最小非因子是对数级别的，可以通过暴力枚举获得。

B 表达式矩阵

题意

构造一个 $n \times m$ 的矩阵，其中每个位置要么是 1，要么是 $+$ 或者 $*$，使得在满足每一行每一列均为一个合法表达式的前提下，最小化每一行、每一列表达式的值之和。

- Idea：Rhodoks
- Solution：Rhodoks，Lrefrain
- Std & Data：Aquamoon
- Verifier：uuku，HCCH

B 表达式矩阵

对于行列大小均为偶数的情况，以 6×8 为例答案形如

```
1 1 1 1 1 1 1 1
1 * 1 * 1 * 1 1
1 1 * 1 * 1 * 1
1 * 1 * 1 * 1 1
1 1 * 1 * 1 * 1
1 1 1 1 1 1 1 1
```

注意到每行/每列恰有一个 11。由于无法构造出只含 1 的解，显然使用乘号连接 11 和 1 是最优的。

B 表达式矩阵

对于列大小为奇数的情况，以 6×9 为例：

1 1 1 1 1 1 1 1 1
1 * 1 * 1 * 1 * 1
1 1 * 1 + 1 * 1 1
1 * 1 * 1 * 1 * 1
1 1 * 1 * 1 + 1 1
1 1 1 1 1 1 1 1 1

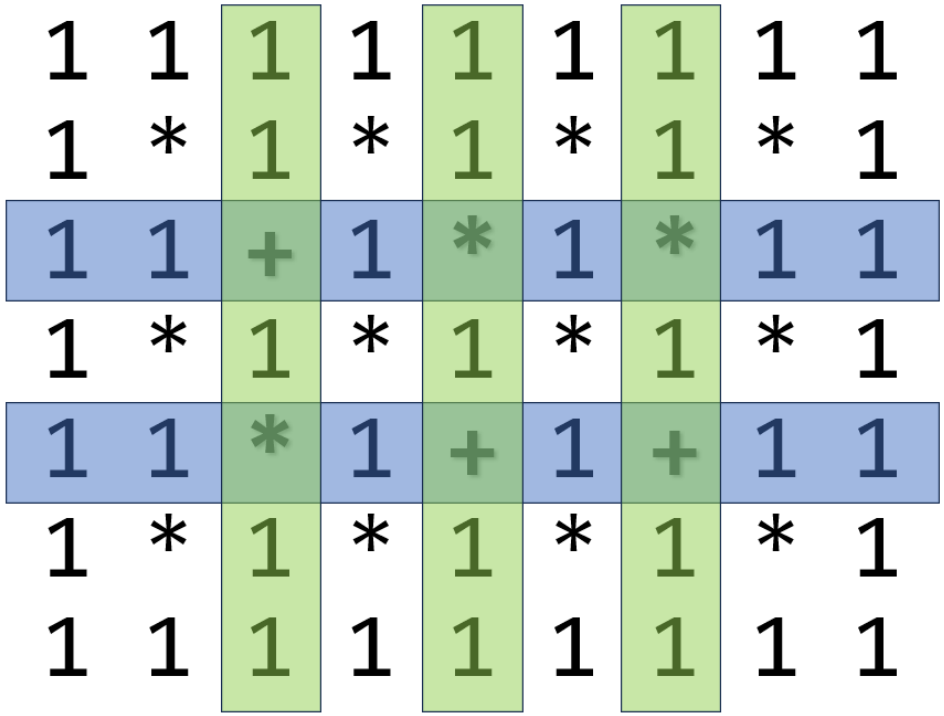
所有包含两个 11 的行都必须使用一个 $+$，否则会产生 11×11 的结果，这是我们不能接受的。

同时，除此之外每多使用一个 $+$，都会使答案增加，所以只需要在这些行中选恰好一个运算符改为 $+$ 即可。

行大小为奇数的情况类似。

B 表达式矩阵

对于行列大小均为奇数的情况，以 7×9 为例：



所有包含两个 11 的行都必须使用一个 $+$，所有包含两个 11 的列都必须使用一个 $+$。

如图所示，观察位于交点的六个运算符，需要保证每行都有一个 $+$，每列都有一个 $+$。可以考虑斜着分配。

B 表达式矩阵

通过搜索和合理的剪枝，能在可接受的时间内算出答案。

考虑答案空间只有 9×9，可以搜出所有的答案硬编码在代码中（打表）。

E 商路

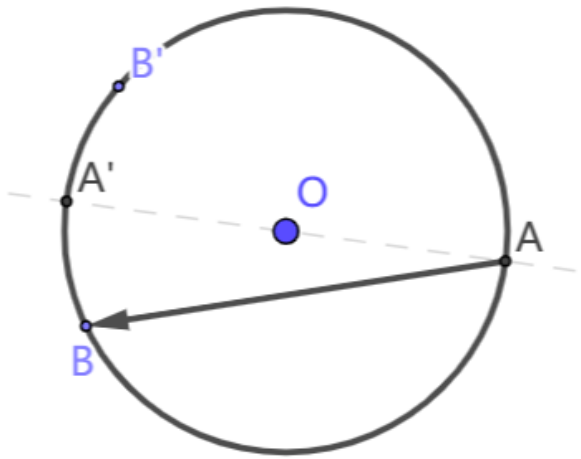
题意

圆上有若干点，每个点只能向离它最远的点连有向线段，保证线段之间不相交不重合的前提下最多可以连几条线段（端点可以相交）

- Idea：Rhodoks
- Solution：Rhodoks
- Std & Data：Veritas
- Verifier：Corycle，Aquamoon

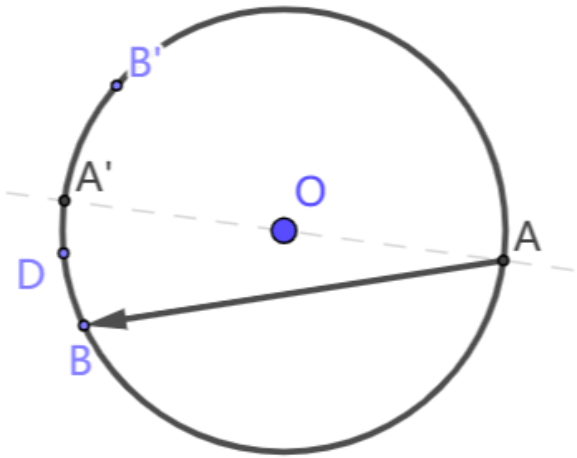
E 商路

引理 1：不存在市场 A, B, C, D，两两互不相同，且商路 $\vec{AB}$ 与 $\vec{CD}$ 同时存在。



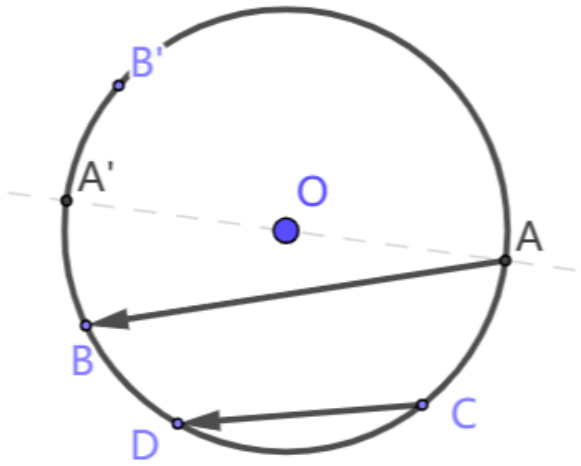
如图所述，假设存在一条商路 $\vec{AB}$，作 AO 直线交圆于另外一点 A'，作点 B 关于直线的对称点 B'。我们假设存在与点 A, B 不同的两点 C, D，$\vec{CD}$ 是另外一条商路。

情况 1： 点 D 位于弧 $B\overset{\frown}{A'B'}$ 上。



此时我们发现距离点 A 最远的点变成了点 D，矛盾。

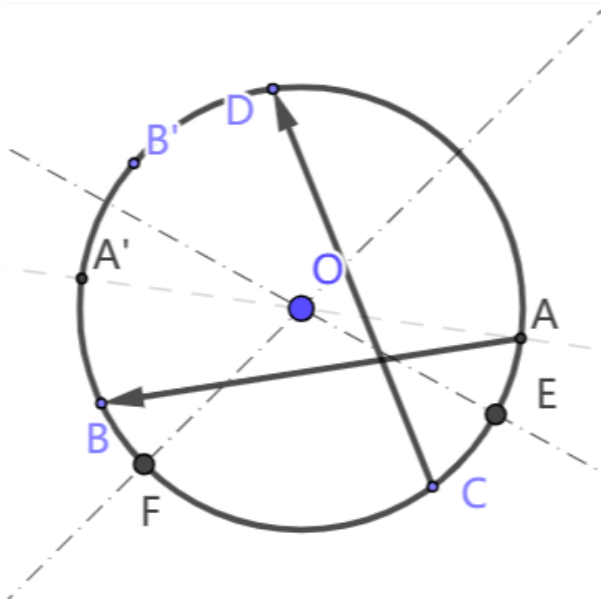
情况 2： 点 D 位于弧 $\overset{\frown}{AB}$ 上。



由于线段 CD 和线段 AB 不能相交，所以点 C 也在弧 $\overset{\frown}{AB}$ 上，我们注意到，因为弧 $\overset{\frown}{AB}$ 不是优弧，所以线段 CD 一定比线段 CA, CB 的较长者短。矛盾。

E 商路

情况 3： 点 D 位于弧 $\overset{\frown}{B'A}$ 上。



作线段 BD, AD 的中垂线交弧 $\overset{\frown}{AB}$ 于点 E, F，因为点 D 是点 C 的最远点，所以点 C 必须在弧 $\overset{\frown}{EF}$ 上，则此时两商路相交，矛盾。

综上所述，引理得证。

E 商路

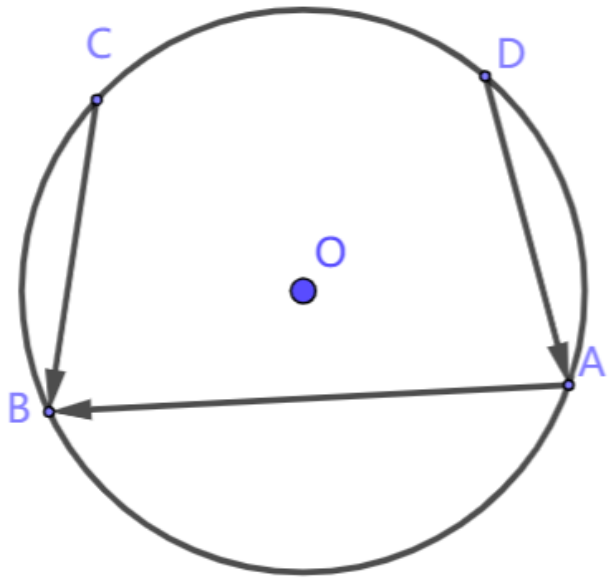
引理 2：若存在商路 $\vec{AB}$，则其余商路必与 $\vec{AB}$ 交且仅交于 A 或 B。

由引理 1 可自然推得。

E 商路

引理 3：若存在商路 $\vec{AB}$，则其余商路只可能：

- 均与 $\vec{AB}$ 交于点 A。
- 均与 $\vec{AB}$ 交于点 B。
- 数量恰好为两条，分别与 $\vec{AB}$ 交于点 A,B，且之间交于圆上一点 C。



若存在两商路，端点分别为 A, B, C, D(如上图所示)，则点 C, D 必定重合，否则违反引理 1。

E 商路

最终题解

通过双指针求出距离每个点 i 最远的点，考虑两种情况：

1. 商路 $\vec{AB}, \vec{AC}, \vec{BC}$，此时需要点 B, C 均为点 A 的最远点且点 C 是点 B 的最远点。三个点构成一个等腰三角形。

2. 所有商路要么以点 A 为起点或以点 A 为终点。
枚举哪个点为引理 3 中前两种情况的交点。所有距点其最远的点中包含点 A 的点向点 A 连商路，对于距离点 A 最远的点，如果其没有向点 A 连商路，则自点 A 向其连商路。

时间复杂度 $O(n)$

D 双子序列

题意

给定 S、s_1、s_2，定义一个 S 的子串的反感度为 s_1、s_2 作为子序列在子串内出现次数的乘积，求所有 S 的子串反感度之和。

- Idea：Luan_233
- Solution：Rhodoks
- Std & Data：Luan_233
- Verifier：Electro_Master，yyf_0404

D 双子序列

考虑贡献转化，转化为对每个 s_1、s_2 构成的 pair，能够在多少个区间内产生贡献，显然是包含这两个串的所有区间，

也就是：左端点在 S 开头到两个串的第一个字符，右端点在两个串最后一个字符到 S 末尾，这些区间均可产生贡献。

考虑设计一个 DP，DP 的时候记录所有匹配方案的左端点：

设 $f[i,j,k]$ 表示当前处理到 S 的第 i 个字符，s_1 匹配完前 j 个字符，s_2 匹配完前 k 个字符时，所有的匹配方案中的 s_1 和 s_2 的起始位置之和。

显然有转移如下：

D 双子序列

$f[i-1,j-1,k] \to f[i,j,k] \quad if\ S_i = s_{1j}$
$f[i-1,j,k-1] \to f[i,j,k] \quad if\ S_i = s_{2k}$
$f[i-1,j-1,k-1] \to f[i,j,k] \quad if\ S_i = s_{1j}\ and\ S_i = s_{2k}$
$f[i-1,j,k] \to f[i,j,k]$

边界条件是：当 j 或者 k 这两维从全 0 变成至少有一个是 1 的时候，表示两个串第一个字符在此出现，此时 $f[i,j,k]+=i$。

统计答案的时候，计算 $f[i,|s_1|,|s_2|]$ 相比于前一位的增量，也就是恰好在这一位结束的匹配方案，乘上 $|S|-i+1$ 后计入答案。

容易发现上面求解的是 f 的差分数组与后缀长度之积，其实等于 $f[i,|s_1|,|s_2|]$ 之和，枚举求和即可。

时空复杂度 $O(|S||s_1||s_2|)$。

K 致命公司

题意

有 n 个通道，会有 m 个弹簧头在 t_i 时刻在 x_i 通道距离你 y_i 米出现，并在出现后的每时刻朝你移动 k 米。每时刻你可以选择看一个通道里的弹簧头，则该时刻该通道里的弹簧头将不会移动。问你最晚可以活到哪个时刻。

- Idea：Rhodoks
- Solution：Shirost
- Std & Data：Shirost
- Verifier：Corycle，HCCH

K 致命公司

答案具有二段性，考虑二分答案，判断能否活到第 $T-1$ 个时刻。

对于第 i 个弹簧头，其在第 $T-1$ 个时刻前不到达原点的条件是：在时刻 $[t_i, T)$ 内，至少向通道 x_i 凝视了 $\lceil \frac{k(T-t_i)-y_i}{k} \rceil = (T-t_i) - \lfloor \frac{y_i}{k} \rfloor$ 个时刻。

这样我们可以得到若干形如在时刻 $[t_i, T)$ 内，至少向通道 x_i 凝视了 m_i 个时刻的限制条件。

注意到这些条件的时间区间的右端点是固定的，所以在较短的区间内进行的凝视，肯定也在较长的区间内起效。

考虑按照起点 t_i 倒序排序所有的要求，贪心地把时间分配给对应的通道，如果可以分配，则说明可以活到第 $T-1$ 个时刻。

时间复杂度 $O((n+m)(\log m + \log y_i))$

J 猜质数 II

题意

给定一个长为 n 的测试点序列 a_i，q 次询问，每次给出 u, l，求一个 r 使得 $\sum_{i=1}^{1e6} score(i, l, r)$ 的值最大，有多个 r 输出较小的。
定义 $score(x, l, r) = \sum_{i=l}^{r} f(x, a_i)$
其中

$$f(x,y) = \begin{cases} u-y, & x=1 \\ u, & 1<x\le y, gcd(x,y)=1 \\ -x\cdot y, & x\neq 1, gcd(x,y)=x \\ 0, & \text{otherwise..} \end{cases}$$

- Idea：MCPlayer542
- Solution：Luan_233，MCPlayer542
- Std & Data：MCPlayer542
- Verifier：yyf_0404，Electro_Master

J 猜质数 II

出题人在打 Div3 的时候看错题导致没能 AK 耿耿于怀，并再缝合了一点点数论成分以加强该题。

简单推一下式子：

$$\begin{aligned}\sum_{i=1}^{10^6} score(i,l,r) &= \sum_{i=1}^{10^6}\sum_{j=l}^{r} f(i,a_j) = \sum_{j=l}^{r}\sum_{i=1}^{10^6} f(i,a_j)\\ &= \sum_{i=l}^{r} u\cdot\varphi(a_i) - a_i\cdot\sigma(a_i)\\ &= \sum_{i=1}^{r} u\cdot\varphi(a_i) - a_i\cdot\sigma(a_i) - \sum_{i=1}^{l-1} u\cdot\varphi(a_i) - a_i\cdot\sigma(a_i)\end{aligned}$$

其中 $\varphi(x)$，$\sigma(x)$ 分别为欧拉函数和除数函数。由于 l 是给定的，显然后一项是常数，因此我们只需最大化 $\sum_{i=1}^{r} u\cdot\varphi(a_i) - a_i\cdot\sigma(a_i)$ 即可。

J 猜质数 II

令 $\Phi(x)=\sum_{i=1}^{x}\varphi(a_i)$，$\Psi(x)=\sum_{i=1}^{x}a_i\cdot\sigma(a_i)$，则我们需要最小化 $ans=u\cdot\Phi(x)-\Psi(x)$。

因此问题可转化为过任意一个可行的 r 的决策点 $(\Phi(r),\ \Psi(r))$ 的一次函数 $y=ux-ans$ 的截距最小是多少。容易发现我们只需在所有可行 r 决策点的下凸包上二分即可。

由于 $\Phi(x)$ 和 $\Psi(x)$ 是单调的，我们只需将询问按 l 离线，倒着逐步将每个 $(\Phi(r),\ \Psi(r))$ 加入单调栈维护下凸包，并对每个 l_i 与当前新加入的 r 一致的询问，用该询问的 u_i 在当前凸包上二分即可。其中 $\Phi(r)$ 和 $\Psi(r)$ 可以用线性筛求得。

注意在有多个 r 可取得最大值时需要输出最小的 r，且 $u\cdot\Phi(r)$ 的大小可能会接近 long long 的上界 (实际数据中并没有刻意去卡)。

J 猜质数 II

也可以考虑将询问按 u 离线，初始时认为每个起始点 l 的最优答案为自身 (即 $u = 0$ 的情况)，在 u 每次增加的时候考虑哪个点不再优于其右侧点，并且用一些数据结构对区间进行合并，因而能计算新的 u 下的答案。

验题中也有使用李超线段树通过该题的队伍。

猜质数

题意

给定 n, k，要求进行 n 次询问，每次可以询问 a_i，得到 $g((p^k)^{a_i})$。要求恰好 n 次询问后自由选择 m，求出 $(p^k)^{a_i} mod (m \cdot a_i)$。

其中 $g(x) = f(f(f(...f(x))))$，$f(x)$ 是 x 的各位数字之和。

- Idea：MCPlayer542
- Solution：MCPlayer542，Aquamoon
- Std & Data：MCPlayer542
- Verifier：uuku，Electro_Master

猜质数

构造题。

题目的条件乍看之下很毒瘤，给出的信息十分有限 (一次 $a_i = 1$ 的查询即可获得所有 x 下的 $g(q^x)$，在输入包含 $a_i = 1$ 的情况下多余的查询是没用的)，实际上只需要进行简单的构造就可以水过本题。

首先经过简单的推导可知 $g(x)$ 对题目范围内涉及的任何数 x 而言都等价于 $(x - 1 \bmod 9) + 1$，即我们可以获得一个数 $\bmod 9$ 的值。

同时考虑 $n = 1$ 时我们必须通过单次询问得到 $q^{a_1} \bmod (m \cdot a_1)$，且有 $m \geq 35$，因此必须确定另一个 q 的取模恒等式。

猜质数

这里的取模恒等式我们可以从两个方向考虑。

首先可以通过欧拉定理得知 $q^{\phi(m)} \bmod m = 1$ 在 $\gcd(q, m) = 1$ 时成立，且若上式成立，可以推广到 $q^{\phi(m \cdot p_0)} \bmod (m \cdot p_0) = 1$ 在 $p_0 \mid m$ 时成立，此时 q 的指数增长速度和模数增长速度相同。

因此可以考虑采用欧拉定理并尽可能地构造优秀的 m 使其满足题目条件且为偶数，则对于所有的 $p \nmid m$ 均能得到正确答案。于是在足够多的 m 里随机选取一个并不断乘二倍增，并利用 $g(x)$ 提供的 $\bmod 9$ 和 $q^{2^i} \bmod (2^{i+1}) = 1$ 进行 CRT 即可。

然而事实是打表之后发现恰当的 m 非常少，因此无法避免在目前数据范围内产生碰撞，被出题人卡掉。

其次可以考虑分圆多项式 $\Phi_{p_0^s}(1) = p_0$ 对任意质数 p_0 和正幂 s 成立，$\Phi_1(1) = 1 - 1 = 0$，且 $q^{p_0^s} - 1 = \prod_{i=0}^{s} \Phi_{p_0^i}(q)$。显然在 $q \bmod p_0 = 1$ 时有 $\Phi_{p_0^i}(q) \bmod p_0 = \Phi_{p_0^i}(1) \bmod p_0 = 0$。

可得 $q^{p_0^s} - 1$ 是 p_0^{s+1} 的倍数。

因此 $q^{p_0^s} \bmod p_0^{s+1} = 1$ 成立，q 的指数增长速度和模数增长速度相同。

我们只需取以上两个结论中较简单的一个特殊情形，即均取 $p_0 = 2$ 即可。

猜质数

考虑 p 是奇质数，即 q 是奇数，因此 $\gcd(q, 2^i) = 1$ 且 $q \bmod 2 = 1$ 成立；并且我们还可以发现对 $p_0 = 2$ 有一些更强的性质，因为我们可以得到 $q^2 - 1 = (q-1)(q+1)$ 是 8 的倍数，故 $q^2 \bmod 8 = 1$。

由此我们可以倍增得到 $q^4 \bmod 16 = 1$，$q^8 \bmod 32 = 1$，$\ldots$

将该恒等式与查询到的 $\bmod 9$ 的结果 CRT 即可得到 $q^{2^i} \bmod 36 \times 2^i$。

最初这题其实需要发现 $q^2 \bmod 12 = 1$，由于 $g(x)$ 的性质而被降级到了目前的版本。

H 最大流

题意

给定一个 n 个点 m 条边的有向无环图，对每个点 i 求 $min(Maxflow(1,i),k)$。

- Idea：Cretaceous
- Solution：Cretaceous，MCPlayer542
- Std & Data：Rhodoks
- Verifier：Lrefrain

H 最大流

我们提出了一个随机算法可以以 $O((n+m)k^2)$ 的复杂度通过本题，算法流程大致描述如下：

对任意边 $e=(u_e,v_e)$ 计算一个向量 $w_e \in \mathbb{Z}_p^k$，定义如下：

若 e 的起始点 $u_e=s$，则 w_e 在 $\mathbb{Z}_p^k$ 中独立均匀随机生成；

若 $u_e \neq s$，则记 u_e 的入边向量组为 M_{u_e}，w_e 是 M_{u_e} 的随机线性组合。

最后对每个汇点 $t \neq s$ 计算其入边对应向量组的秩 $rank(M_t)$，若 $rank(M_t)<k$ 则认为 $rank(M_t)$ 是该汇点的最大流；否则报告最大流至少为 k。

H 最大流

一个直观的理解如下：

由最大流最小割定理，最大流等于最小割，假设我们找到一组最小割 $e_1, e_2, \cdots, e_x$，那么最终汇点 t 的入边向量组均为割集边向量 $w_{e_1}, w_{e_2}, \cdots, w_{e_x}$ 随机生成，且割集边向量大概率线性无关。

所以 t 的入边向量组的秩，大概率是最小割的值。特别的，如果最小割的值大于 k，则我们的算法无法判断具体的最小割值。

H 最大流

因为图是 DAG，可以使用拓扑序推进算法。

注意到生成出边时，在入度和出度为 $O(m)$ 时，朴素的实现会退化成 $O(m^2k)$ 复杂度。
解决方法是，对于每个点维护一个大小为 $k \times k$ 的线性基，将所有入边向量插入线性基，单次插入复杂度为 $O(k^2)$，点的入边向量组的秩，即是线性基中向量的个数。生成出边时，仅使用线性基中的至多 k 个向量随机生成，单次生成复杂度也是 $O(k^2)$。

对于单个答案的错误概率，我们给出一个上界为 $O(\frac{n}{p})$。理论上需要跑至少 2 次以保证所有点的正确性，但每个点的概率并不互相独立，实测单次运行的结果正确率非常高。

H 最大流

FOCS 2011 的论文 Graph Connectivities, Network Coding, and Expander Graphs 给出了一个和我们的算法极其相似的做法，可以在 $O(mk^{\omega-1})$ 的时间复杂度内进行求解，其中 $\omega \approx 2.38$ 是矩阵乘法的复杂度指数。

其相对我们做法的主要区别在于：

1. 使用 Expander Graphs and Superconcentrators 方法将原图转换成一个最大流和原图相等，但是点入度限制为常数的新图。从而取代线性基。
2. 使用复杂度更优的矩阵求秩方法，使 $a \times b$ 的矩阵求秩可以在 $O(ab^{\omega-1})$ 的复杂度内完成。

欢迎感兴趣的选手阅读。