

薇薇安杯题目讲解

顾何济

2025 年 6 月 3 日

大豆难度预测：IAFL/MBC/KDH/JEG

实际难度分布：AILF/BMK/HCD/EJG

A - Balanced Eating

题意

有两个计数器 s_0 和 s_1 ，初始都为 0。对于每个 a_i ，Varesa 等概率地选择起始索引 $j \in \{0, 1\}$ ，然后将 a_i 块肉按以下规则分配：

第 k 块肉分配给 $s_{(j+k-1) \bmod 2}$ ，即 $s_{(j+k-1) \bmod 2} := s_{(j+k-1) \bmod 2} + 1$ ，其中 $k = 1, 2, \dots, a_i$ 。

求分配完所有肉后 $|s_0 - s_1| \leq 1$ 的概率。 $\sum |a_i| \leq 2 \cdot 10^5$ 。

A - Balanced Eating

题意

有两个计数器 s_0 和 s_1 ，初始都为 0。对于每个 a_i ，Varesa 等概率地选择起始索引 $j \in \{0, 1\}$ ，然后将 a_i 块肉按以下规则分配：

第 k 块肉分配给 $s_{(j+k-1) \bmod 2}$ ，即 $s_{(j+k-1) \bmod 2} := s_{(j+k-1) \bmod 2} + 1$ ，其中 $k = 1, 2, \dots, a_i$ 。

求分配完所有肉后 $|s_0 - s_1| \leq 1$ 的概率。 $\sum |a_i| \leq 2 \cdot 10^5$ 。

现场情况

首个提交：00:02@Farewell

首个 AC：00:02@Farewell

通过队伍：75

A - Balanced Eating

若 a_i 是偶数, 则 s_0 和 s_1 都增加了 $\frac{a_i}{2}$, 对 $s_0 - s_1$ 的值没有影响。

若 a_i 是奇数, 通过简单的模拟可以知道:

- 若 $j = 0$, 则 $s_0 := s_0 + \lceil \frac{a_i}{2} \rceil$, $s_1 := s_1 + \lfloor \frac{a_i}{2} \rfloor$, 这将会使得 $s_0 - s_1$ 增加 1。
- 若 $j = 1$, 则 $s_0 := s_0 + \lfloor \frac{a_i}{2} \rfloor$, $s_1 := s_1 + \lceil \frac{a_i}{2} \rceil$, 这将会使得 $s_0 - s_1$ 减少 1。

A - Balanced Eating

设 a_i 中一共有 m 个奇数，则问题变为：对一个计数器 s 进行 m 次操作，每次操作会等概率地将 s 增加 1 或者减少 1，求 $|s| \leq 1$ 的概率。

一共有 2^m 种操作方式，对 m 进行分类讨论：

- 若 m 是偶数，则必须有 $\frac{m}{2}$ 次增加， $\frac{m}{2}$ 次减少，符合条件的操作方式一共有 $\binom{m}{\frac{m}{2}}$ 种。
- 若 m 是奇数，则增加的次数可以为 $\lfloor \frac{m}{2} \rfloor$ 或者 $\lceil \frac{m}{2} \rceil$ ，于是符合题意的操作方式一共有 $\binom{m}{\lfloor \frac{m}{2} \rfloor} + \binom{m}{\lceil \frac{m}{2} \rceil} = 2\binom{m}{\lfloor \frac{m}{2} \rfloor}$ 种。

在实现的时候，可以将上述分类讨论合并为 $(m \bmod 2 + 1)\binom{m}{\lfloor \frac{m}{2} \rfloor}$ ，最后将这个数乘以 2^m 的逆元即是答案。

总结：简单组合题。

题意

给出两个排列 a 和 b ，问是否存在一颗有根树使得 a 是一个合法 DFS 序， b 是一个合法 BFS 序。 $\sum |a| \leq 2 \cdot 10^5$ 。

I - FST: First Search Traversal

题意

给出两个排列 a 和 b ，问是否存在一颗有根树使得 a 是一个合法 DFS 序， b 是一个合法 BFS 序。 $\sum |a| \leq 2 \cdot 10^5$ 。

现场情况

首个提交: zuiyuyu@00:04

首个 AC: zuiyuyu@00:04

通过队伍: 64

考虑一颗菊花图。由于在 BFS 和 DFS 的伪代码中，都是任选一个儿子的排列，因此只要有 $a_1 = b_1$ ，我们就能构造出以 a_1 为根的菊花图满足题意。

总结：签到题。

题意

给出一个包含大小写字母、数字和 ? 的字符串 s ，你可以将 ? 替换为任意字母或数字。求所有的替换方法，使得：

- 1, 1, I 不能出现在两端。
- 1, 1, I 的两旁必须是与它同类别的字符。

题意

给出一个包含大小写字母、数字和 ? 的字符串 s ，你可以将 ? 替换为任意字母或数字。求所有的替换方法，使得：

- 1, 1, I 不能出现在两端。
- 1, 1, I 的两旁必须是与它同类别的字符。

现场情况

首个提交: 00:15@RAY__

首个 AC: 00:15@RAY__

通过队伍: 62

容易观察出填写字符是无后效性的，因此我们设 $dp[i][j]$ 为当前已经填了 i 个字符， s_{i+1} 是第 j 种字符的填写方案数。我们定义 $j = 0, 1, 2, 3, 4, 5$ 分别为 I，其他大写字母，1，其他小写字母，1，其他数字。则最后的答案为 $dp[n-1][1] + dp[n-1][3] + dp[n-1][5]$ 。

分类讨论当前字母的种类。由于三种字符本质是一样的，这里仅以大写字母为例：

- $j = 0$ ：I 的左边必须是大写字母，因此 $dp[i][0] = dp[i-1][0] + dp[i-1][1]$ 。
- $j = 1$ ：其他大写字母的左边不能是 1 或者 1，因此 $dp[i][1] = 25(dp[i-1][0] + dp[i-1][1] + dp[i-1][3] + dp[i-1][5])$ 。

若 s_{i+1} 不是 ? 而是个字母, 则除了与 s_{i+1} 对应的 j 以外其他的 $dp[i][j]$ 都是 0。而在前页中描述的 $dp[i][1] = 25(\dots)$ 的 25 也要去掉。

边界条件为 $dp[0][0] = dp[0][2] = dp[0][4] = 0$, 其他的 j 值根据 s_1 的值确定。

总结: 一坨史的线性状态机 dp。

题意

交互题。每次可以询问 $a_l, a_{l+1}, \dots, a_r$ 的最长严格递增序列的长度，用 20 次询问找出一个索引 i 和一个数值 x ，使得 $a_i \neq x$ 。 $\sum |a| \leq 10^5$ ， $1 \leq a_i \leq 3$ ，保证 a 中有 1, 2 和 3。

F - Where the West Wind Ends

题意

交互题。每次可以询问 $a_l, a_{l+1}, \dots, a_r$ 的最长严格递增序列的长度，用 20 次询问找出一个索引 i 和一个数值 x ，使得 $a_i \neq x$ 。 $\sum |a| \leq 10^5$ ， $1 \leq a_i \leq 3$ ，保证 a 中有 1, 2 和 3。

现场情况

首个提交: 00:20@Teenbhai

首个 AC: 00:20@Teenbhai

通过队伍: 58

F - Where the West Wind Ends

首先通过一次询问找出整个数组的 LIS 长度。若为 1，则数组必然是 $3, 3, \dots, 3, 2, 2, \dots, 2, 1, 1, \dots, 1$ ，这说明 $a_1 \neq 1$ 。

如果为 2 或者 3，则可以通过二分找到 LIS 长度第一次变成 2 或 3 的下标 i 。容易看出 $a_i \neq 1$ （否则无法贡献 LIS 长度）。

Bonus: 用 $O(n)$ 的时间复杂度实现 interactor。

题意

维护一个数组 a ，支持在右端插入元素和查询 $\sum_{i=l}^r \min_{j=i}^r a_j$ ，强制在线。 $n, q \leq 4 \cdot 10^5$ 。

B - Adventure for Black

题意

维护一个数组 a ，支持在右端插入元素和查询 $\sum_{i=l}^r \min_{j=i}^r a_j$ ，强制在线。 $n, q \leq 4 \cdot 10^5$ 。

现场情况

首个提交：00:17@Farewell

首个 AC：00:31@kawaii xde,ss,mwh

通过队伍：20

B - Adventure for Black

方法一：倍增（firefly）

我们可以模拟一下求解 $\sum_{i=l}^r \min_{j=i}^r a_j$ 的过程。比如对于 $[3, 2, 5, 4, 6]$ ，我们知道后缀最小值之和是 $6 + 4 + 4 + 2 + 2$ 。关注每次加数变化的过程，可以看到是寻找左侧下一个最小值的过程，而每个加数的贡献就是该数距离他下一个最小值的距离。

每次暴力寻找下一个最小值的时间复杂度是 $O(q)$ 的（考虑递增序列即可），我们可以使用倍增来优化跳转的过程。在倍增数组中，我们维护跳转 2^i 次所到达的下标 j ，以及 $(j, r]$ 之间的后缀最小值之和。在查询中二进制拆分跳转次数即可。时间复杂度为 $O((n + q) \log n)$ 。

B - Adventure for Black

方法二：线段树 (Hongrock)

我们可以通过 dp 计算所有的 $\sum_{i=1}^r \min_{j=i}^r a_j$ ，设这个玩意儿是 $dp[r]$ 。

和前页我们观察的一样， $dp[r]$ 同样由一些递减的数字的贡献组成。

对于一个 $[l, r]$ 的查询，我们用线段树维护 a 的最小值的下标，就能求出 $[l, r]$ 中的最小值下标 p 。这样，我们就可以得到 r 一直跳转到 p 的所有最小值的贡献为 $dp[r] - dp[p]$ 。接着， p 的贡献是 $[l, p]$ 这个区间，因此答案就是 $dp[r] - dp[p] + a[p](l - p + 1)$ 。

总结：不要写火箭！

题意

给出一个简化版桥牌的双明手结果，求出双方在最优情况下的最佳定约

题意

给出一个简化版桥牌的双明手结果，求出双方在最优情况下的最佳定约

现场情况

首个提交: `luo_yi@00:30`

首个 AC: `luo_yi@00:30`

通过队伍: 19

总之，我们可以预处理第 i 个人叫到最终定约是 j 所得的分数 $a[i][j]$ 。

方法一：dp (SATSKY)

容易定义状态 $dp[i][j]$ 为第 i 个人当前叫牌，前一个人叫的定约为 j 时第 i 个人的收益。那么此时这个人可以：

- 不叫牌，收益为 $-a[1-i][j]$ 。
- 叫牌，收益为 $-\min\{dp[1-i][j+1], dp[1-i][j+2], \dots, dp[1-i][35]\}$ 。

因此状态转移方程为 $dp[i][j] = \max\{-a[1-i][j], -\min_{k=j+1}^{35} dp[1-i][k]\}$ ，最后结果为 $\max\{dp[0][0], dp[1][0]\}$ 。

方法二：贪心 (firefly)

如果一方开始叫牌，他一定会直接叫到他获利最大的定约，而不会一个一个选择前面的叫品。比如，如果他可以以红桃为将牌获得 9 墩，那么他一定会直接叫 3H。而到另一方之后，他也会作类似的分析，如果不能获得正分，则会叫其他叫品让对方的正分减少。如此贪心循环，直到最后的定约不再变动为止。

以样例为例，3H (+140) $\rightarrow$ 3S (+100) $\rightarrow$ 4C (+130)。

对两方依次作这种贪心，由于每次贪心将牌一定会改变，因此最多贪心五轮。

总结：披着大模拟外衣的算法题。

题意

求一个相邻项互质的正整数序列 $\{a_n\}$, 使得对于 q 组询问 (l, r) , 有 $a_l + a_{l+1} + \cdots + a_r$ 为合数。

K - Pleasure of Hope

题意

求一个相邻项互质的正整数序列 $\{a_n\}$, 使得对于 q 组询问 (l, r) , 有 $a_l + a_{l+1} + \cdots + a_r$ 为合数。

现场情况

首个提交: 00:19@Equinox Beaten Hedian

首个 AC: 00:22@luo_yi

通过队伍: 17

K - Pleasure of Hope

Hint: q 的范围是诈骗, q 可以集中在大约 $\sqrt{n}$ 的范围内, 这表明至少在 $n \leq 200$ 的范围内可以构造出满足所有 (l, r) 均有 $\sum_{i=l}^r a_i$ 为合数的序列。

方法一: (firefly)

如果把这个限制松弛一些, 改为 $l_j < r_j$, 那么事情就简单了, 我们只需要令 $a_i = 2i - 1$, 这样 $S_i = \sum_{k=1}^i a_k = i^2$, 于是就有 $S_r - S_{l-1} = r^2 - (l-1)^2 = (r-l+1)(r+l-1)$ 一定是合数,

秉持着要让 S_i 是平方数, 且 a_i 是合数的思想, 我们注意到 $\sum_{i=1}^n i^3 = (\frac{1}{2}n(n+1))^2$ 。这下就两个条件全满足了! 由于 a_1 也要是合数, 只需要稍微修改一下让 $a_i = (i+1)^3$ 即可。

总结: 一看就是大豆喜欢出的构造题。

H - An AtCoder-style Problem

题意

给出一个数组 A ，对每个前缀 $A[1 \dots i]$ 询问是否存在一个长度为 $i + M - 1$ 的排列 B ，使得 B 所有长度为 $M + 1$ 的子数组的最大值的可重集与 $A[1 \dots i]$ 相同。

H - An AtCoder-style Problem

题意

给出一个数组 A ，对每个前缀 $A[1 \dots i]$ 询问是否存在一个长度为 $i + M - 1$ 的排列 B ，使得 B 所有长度为 $M + 1$ 的子数组的最大值的可重集与 $A[1 \dots i]$ 相同。

现场情况

首个提交: 00:57@kawaii xde,ss,mwh

首个 AC: 00:57@kawaii xde,ss,mwh

通过队伍: 12

H - An AtCoder-style Problem

方法一：二分 + 贪心 (firefly)

考虑对数组 A 如何构造对应的 B 。首先对 A 排序，如果最大值不是 B 的长度那么肯定不行。

我们以样例的 $M = 1$, $A = [3, 3, 5, 5]$ 为例。此时 5 出现了两次，因此我们将 5 贪心地放置在 B_4 这个位置，这样可以最大程度地不影响左侧的数组。

接下来 4 没有出现，这说明 4 完全被 5 挡住了，那么 4 就应该放在 5 的右边，即 B_5 的位置。

类似地，我们可以依次放置 3, 2, 1，得到 $B = [1, 3, 2, 5, 4]$ 。

H - An AtCoder-style Problem

我们称一个前缀 $A[1 \dots i]$ 的最大值等于 $i + M$ 的前缀为有效前缀，只有这些前缀的答案才有可能为 1。接下来我们说明这些前缀具有单调性。

设两个前缀分别为 $A[1 \dots i]$ 和 $A[1 \dots j]$ ($i < j$)，则在处理 $A[i + 1 \dots j]$ 的过程中一定会安排好 $[i + M + 1, j + M]$ 的所有数。而由于我们是从右向左贪心的，这些数一定都会堆积在数组的右侧，这就说明如果 $A[1 \dots j]$ 是可行的，那么 $A[1 \dots i]$ 也是可行的。

所以我们在有效前缀上进行二分，找出第一个不行的有效前缀，将其左的所有有效前缀赋值 1 即可。

H - An AtCoder-style Problem

方法二 (satsky):

注意到: 若 $A_x = k$, 则 $A[1 \dots i]$ 是合法前缀的一个必要条件是 $A[1 \dots i]$ 中至少 $k - M$ 个值满足 $A_j \leq A_x$ 。这是因为在 B 一共 $i + M$ 个数里面, 超过 k 的一共就 $i + M - k$ 个位置, 也就是说就算每个 $> k$ 的数都是滑动窗口的最大值, 也才能给 A 贡献 $i + M - k$ 个 $> k$ 的数, 那么这就表明 $\leq k$ 的数至少有 $i - (i + M - k) = k - M$ 个。

如何找到满足这一条件的下标? 我们将二元组 (A_i, i) 从小到大排序, 就可以离线地求出每个 A_i 对应的前缀范围, 将它们取 $\min$ 就是满足这一条件的所有前缀。

然后, 再加上前文所述的前缀最大值等于 $i + M$, 没有出现 $m + 1$ 次的数, 数组最小值 $> M$ 等条件, 就构成了题设的充要条件。

总结: 非常 AtCoder 的一道题。

题意

一个能跳跃 $1 \times k$ 格的马放置在 n 行 m 列的棋盘上第 i 行第 j 列的格子上。求这只马能跳跃到棋盘中的多少个格子中。

题意

一个能跳跃 $1 \times k$ 格的马放置在 n 行 m 列的棋盘上第 i 行第 j 列的格子上。求这只马能跳跃到棋盘中的多少个格子中。

现场情况

首个提交: 01:16@unsettled

首个 AC: 01:16@unsettled

通过队伍: 9

Fun fact: firefly 曾经认为这是签到题。

C - Knight

不妨设 $n \leq m$ 。首先处理一些特殊情形：

- 若 $k \geq n$ ，或者初始坐标距离棋盘边界均不足 k 格，则这只马无法移动，答案为 1。
- 若 $n < k \leq m$ ，则这只马只能横着跳（走 1 行 m 列）而不能竖着跳。此时马在每一行的可达坐标构成一个公差为 $2k$ 的等差数列。容易看出，奇数行和偶数行的列坐标是相同的，因此我们分别求出奇数行和偶数行的列坐标数目，再累加即可。

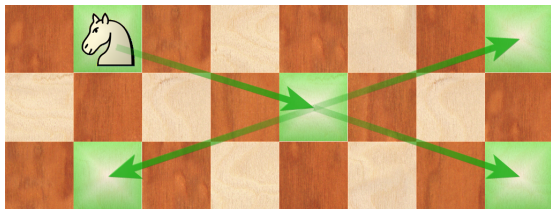


图: $k = 3$ 的一个例子。

C - Knight

若 $k > m$ ，并且马可以移动，此时马就既可以横着跳，也可以竖着跳。

我们先来看一个容易处理的情形，即 k 是偶数时，此时我们可以证明：这个马可以遍历一个 $(k+1) \times (k+1)$ 正方形的边界。 $k=2$ 的示例样例中已经给出，下图是 $k=4$ 的情形。显然，没有标记为绿的区域都是初始无法移动的格子。

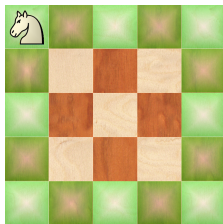


图: 这只马可以遍历边界后返回左上角，这说明只要马位于边界，它就可以遍历整个绿色区域。

C - Knight

若我们当前的区域是一个 $(k+1) \times (k+2)$ 的矩形，由于多了一列，位于左上角的马就可以在跳到右下角 $(k+1, k+1)$ 后再次向上跳，进入 $(1, k+2)$ ，从而进入一个新的矩形边界。这个矩形边界和原有的矩形边界重叠后，未被标记的区域又恰好构成了新的初始无法移动的格子。

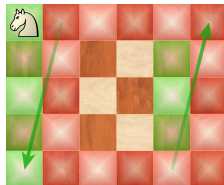


图: $n = 5$, $m = 6$ 的情形。图中的绿色箭头表明绿色矩形边界和红色矩形边界是可以互通的。

这说明：只要这只马初始可以移动，它就一定能访问所有初始可以移动的格子。由此，我们只要求出中间那个初始不能移动的区域的大小，就可以求出这只马能访问的格子数目。

这个矩形的长 $h = \max\{2k - n, 0\}$ ，宽 $w = \max\{2k - m, 0\}$ ，将两者相乘就可以得到答案，在用 nm 减去即可得到答案。

C - Knight

接下来考虑 k 为奇数的情形。我们将棋盘黑白染色，那么马只能访问到所有初始可以移动格子中，与马的初始格黑白颜色相同的格子。

因此，我们需要求出整个 $n \times m$ 棋盘和 $h \times w$ 不可达区域中与马的初始格颜色相同的格子，两者相减得到答案。

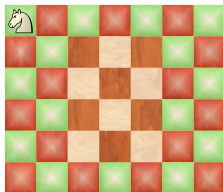


图: $k = 5$, $n = 6$, $m = 7$ 的情形。马只能跳到边界中的绿色格子，而无法进入红色格子。

总结：需要一定思维量的分类讨论题。

题意

给出 n 个引擎和 m 组驱动盘，对每个引擎，找出与最佳驱动盘配套时的伤害期望最大值。

题意

给出 n 个引擎和 m 组驱动盘，对每个引擎，找出与最佳驱动盘配套时的伤害期望最大值。

现场情况

首个提交: 00:55@luo_yi

首个 AC: 01:22@luo_yi

通过队伍: 5

D - Phaethon's Melody

设攻击力为 a ，暴击率为 c ，暴击伤害为 d ，则当造成伤害时：

- 有 c 的概率造成 $a(1 + d)$ 的伤害。
- 有 $1 - c$ 的概率造成 a 的伤害。

因此，期望伤害为 $ca(1 + d) + (1 - c)a = a(1 + cd)$ 。其中 a 并不重要，所以我们就需要最大化 cd 的值。

具体地说，对于第 i 个 1 型引擎，我们希望求得 $\max_j (\min\{P, x_j + b_i\}y_j)$ ，而对于 2 型引擎，我们希望求得 $\max_j (\min\{P, x_j\}(y_j + b_i))$ 。

D - Phaethon's Melody

由于 1 和 2 本质是一样的，我们只介绍如何找出 1 型音擎的最大期望伤害。

方法一：凸包优化 (Satsky)

对于一些 b_i ，我们需要分别求出 $\min\{P, x_j + b_i\}y_j$ 的最小值。这实际上就是由两条线段 $y_j b_i + x_j y_j (b_i \leq P - x_j)$ 和 $P y_j (b_i \geq P - x_j)$ 组成。那么我们已经有了 $2m$ 条线段，就可以在上面构建上凸包，随后在凸包上二分或者双指针就可以求出每个 b_i 对应的最大值了。

方法二：李超树 (firefly)

众所周知李超树插入线段是 $O(\log^2 n)$ 的，会被本题卡掉。因此，为了使得李超线段树能维护一些 $y_j b_i + x_j y_j$ 的 **直线**，我们需要将音擎按照 b_i 从大到小排序，维护一个 $x_j \geq P - b_i$ 所对应 y_j 的集合，这样就能求出这部分 $P y_j$ 的最大值。而李超树中维护 $x_j < P - b_i$ 的直线，这样就能求出这部分 $y_j b_i + x_j y_j$ 的最大值，从而在单 $\log$ 时间内解决了问题。

D - Phaethon's Melody

Fun fact:



cf: -firefly-

24/8/3

不行，我笔试前必须要解决困扰我一周的问题



一个傻瓜

24/8/3

什么问题



cf: -firefly-

24/8/3

给出一些 a_i b_i 和一些查询 c_j ，求 $a_i(c_j+b_i)$ 的最大值

题意

设计一个数据结构，支持单点修改和区间 lcm 的因子个数查询。 $n, q \leq 2 \cdot 10^5$, $V \leq 10^5$ 。

题意

设计一个数据结构，支持单点修改和区间 lcm 的因子个数查询。 $n, q \leq 2 \cdot 10^5$, $V \leq 10^5$ 。

现场情况

首个提交: 00:09@femboys

首个 AC: 02:12@luo_yi

通过队伍: 3

注意到 $\geq \sqrt{V}$ 的质因子只会在 lcm 查询中出现一次，因此我们需要想办法维护这些只出现一次的质因子。

这里有大量其他的做法，如带修莫队 (elijah, 卡了但是没完全卡掉)，KD 树 (StarSilk)，CDQ 分治 (SSerxhs)，在此不展开说明。

我们考虑非常朴素的做法：用 bitset + 线段树来维护 $\geq \sqrt{V}$ 一共 $9592 - 65 = 9527$ 个质因子的出现次数。这样，一个线段树结点需要 $65 \times 4 + \frac{9527}{8} \leq 1451$ 个字节。本题的 n 有 $2 \cdot 10^5$ ，按照 $4n$ 的线段树开法需要 1.1608×10^9 字节，显然已经超过了内存限制。

那么，只要能把线段树的节点数量压缩十倍，就可以把内存降低到 116MB 左右，从而通过本题。这启发我们进行分块，然后在块上建立线段树即可。块内保存每个整数的质因子及其幂，由于每个数的不同质因子不会超过 6 个，因此我们可以随便地用一个 *map* 来保存。

查询时，先查询线段树覆盖的块，再暴力合并左右两边块的边角部分。修改时，修改所在块然后暴力合并即可。

总结：差评如潮。

题意

给出一个数，要求支持路径压缩操作和查询子树 k -MEX。 $n, q \leq 10^6$ 。

题意

给出一个数，要求支持路径压缩操作和查询子树 k -MEX。 $n, q \leq 10^6$ 。

现场情况

首个提交: 02:04@kawaii xde,ss,mwh

首个 AC: 02:50@kawaii xde,ss,mwh

通过队伍: 2

如果是一些区间查询 k -MEX，那自然是十分 trivial 的，我们有多种方法能在 $O(nk \log n)$ 时间内解决。因此我们也想把子树 k -MEX 转化为区间查询，但是树的结构是会变的！也就是说我们需要一种 dfs 的方法，能使得在路径压缩前后 dfs 序不变。

考虑对 $u_2, u_3, \dots, u_m$ 进行路径压缩后， $u_2, u_3, \dots, u_m$ 就变成了兄弟，于是 dfs 序会变成下面的样子：

$$u_2, (u_2 \text{ 的后代}), u_3, (u_3 \text{ 的后代}), \dots, u_m, (u_m \text{ 的后代})$$

在原本的树中， $u_2 - u_3 - \cdots - u_m$ 是一条链，为了确保访问 u_2 后先访问 u_2 的后代，应该在 dfs 访问儿子的序列中，将 u_3 排到最后一位。同理，在 u_3 的儿子中， u_4 也是最后一个被 dfs 的。

这样，通过调整 dfs 时访问儿子的顺序，我们就成功在整个操作中保证了 dfs 序不变。由于在路径压缩过程中子树的大小会发生改变，因此需要额外实时维护子树的大小信息，并将所有查询离线下来保存在 query 数组中。

最后，根据我们的 query 数组，离线地查询每个对应区间的 k-MEX，就可以完成本题了。

总结：本题的 ML 是卡线段树合并的，我不知道为什么会从一开始就写出了正解的 SSerxhs MLE 多次，在此再次致歉。

题意

解线性同余方程组 $a_i(c_i + c_{(i+n-2) \bmod n+1}) + b_i \equiv v \pmod{m}$

题意

解线性同余方程组 $a_i(c_i + c_{(i+n-2) \bmod n+1}) + b_i \equiv v \pmod{m}$

现场情况

首个提交: `luo_yi@03:39`

首个 AC: N/A

通过队伍: 0

前页已经给出了形式化题意，我们首先记 $x_i \equiv c_i + c_{(i+n-2) \bmod n+1} \pmod{m}$ ，那么原方程组就是一组形如 $a_i x_i \equiv v - b_i \pmod{m}$ 的方程组。我们需要每一个方程都有解，这就说明 $v - b_i \equiv 0 \pmod{\frac{m}{\gcd(m, a_i)}}$ 。

记 $g_i = \gcd(m, a_i)$ ，于是我们得到了一组同余方程组 $v \equiv b_i \pmod{\frac{m}{g_i}}$ ，利用 EXCRT 即可求出 v 的通解 $v \equiv v^* + pL \pmod{m}$ ，其中 $L = \text{lcm}(\frac{m}{g_1}, \frac{m}{g_2}, \dots, \frac{m}{g_n})$ 。同样地，我们可以求出 $x_i \equiv x_i^* + t_i \frac{m}{g_i} \pmod{m}$ 。

你以为这就结束了？这才只是开始！

接下来我们需要考虑方程组 $c_i + c_{(i+n-2) \bmod n+1} \equiv x_i \pmod{m}$, 即:

$$\begin{pmatrix} 1 & & \cdots & & 1 \\ 1 & 1 & & \cdots & \\ & 1 & 1 & \cdots & \\ \vdots & \vdots & \vdots & \ddots & \\ & & & 1 & \\ & & & 1 & 1 \end{pmatrix} c \equiv x \pmod{m}$$

我们对这一方程组进行行变换：

$$\begin{pmatrix} & & \cdots & 1 + (-1)^n \\ 1 & 1 & & \\ & 1 & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ & & & 1 \\ & & 1 & 1 \end{pmatrix} c \equiv \begin{pmatrix} \sum_{i=1}^n (-1)^{i-1} x_i \\ x_2 \\ x_3 \\ \vdots \\ x_4 \\ x_5 \end{pmatrix} \pmod{m}$$

G - Navigation Compass

当 n 为奇数时:

此时第一行的方程为 $2c_n \equiv \sum_{i=1}^n (-1)^{i-1} x_i \pmod{m}$ 。若 m 是奇数, 由于 $\gcd(2, m) = 1$, 所以 c_n 一定有解。这就表明可到达的 v 一共有 $\frac{m}{L}$ 个。

若 m 是偶数, 则 RHS 也必须是偶数。容易发现, RHS 的奇偶性和 $\sum_{i=1}^n x_i$ 相同。我们考查 x_i 的通解 $x_i \equiv x_i^* + t_i \frac{m}{g_i} \pmod{m}$, 若存在一个 $\frac{m}{g_i}$ 为奇数, 则说明 x_i 可以在奇数和偶数之间变化, 那么 $\sum_{i=1}^n x_i$ 总可以是偶数, 同上段。

若 $\frac{m}{g_i}$ 均为偶数, 则 x_i 的奇偶性不受后面解系项的影响。我们考察

$x_i^*(v) \equiv \left(\frac{v-b_i}{g_i}\right) \left(\frac{a_i}{g_i}\right)^{-1} \pmod{m}$ 。为了避免复杂的讨论, 我们可以注意到 $x_i^*(v+2L) \equiv x_i^*(v) \pmod{2}$, 因此我们只需要让 v 变化 L , 就能求出符合 RHS 是偶数的 v 数量。

接下来, 求出 c_n 的特解 c_n^* 后, 再逐个回代即可。

G - Navigation Compass

当 n 为偶数时, 此时第一行的方程为 $0 \equiv \sum_{i=1}^n (-1)^{i-1} x_i \pmod{m}$ 。我们设上页中 $\left(\frac{a_i}{g_i}\right)^{-1}$ 为 y_i , 从而我们有:

$$\begin{aligned}\sum_{i=1}^n (-1)^{i-1} x_i &\equiv \sum_{i=1}^n (-1)^{i-1} \left(x_i^* + t_i \frac{m}{g_i} \right) \\ &\equiv \sum_{i=1}^n (-1)^{i-1} \left(y_i \left(\frac{v - b_i}{g_i} \right) + t_i \frac{m}{g_i} \right) \\ &\equiv \sum_{i=1}^n (-1)^{i-1} \frac{y_i}{g_i} (v - b_i) + \sum_{i=1}^n (-1)^{i-1} t_i \frac{m}{g_i} \\ &\equiv 0 \pmod{m}\end{aligned}$$

注意加号右边本质上是 $\frac{m}{g_i}$ 的线性组合, 根据裴蜀定理有

$$\sum_{i=1}^n (-1)^{i-1} \frac{y_i}{g_i} (v - b_i) \equiv 0 \pmod{\gcd\left(\frac{m}{g_1}, \frac{m}{g_2}, \dots, \frac{m}{g_n}\right)}$$

注意到 $\gcd\left(\frac{m}{g_1}, \frac{m}{g_2}, \dots, \frac{m}{g_n}\right) = \frac{m}{L}$ 。我们代入 $v = v^* + tL$, 有:

$$\sum_{i=1}^n (-1)^{i-1} y_i \left(\frac{v^* - b_i}{g_i} \right) + t \sum_{i=1}^n (-1)^{i-1} \frac{L}{g_i} \equiv 0 \pmod{\frac{m}{L}}$$

我们得到了一个全新的，关于 t 的同余方程组。而的确， t 的取值范围确实正是 $Z_{\frac{m}{L}}$ ，这表明我们的分析没有发生矛盾。我们记这个方程为 $A + Bt \equiv 0 \pmod{\frac{m}{L}}$ 。

这个方程有解的条件为 $A \equiv 0 \pmod{\gcd(B, \frac{m}{L})}$ 。若无解，则说明 v 不存在。若有解，我们可以得出 t 的解系大小，它就代表着可用 v 的数量。

什么，我们还要求 $c_1, c_2, \dots, c_n$ ？由于我们已经证明原方程组有解，而原方程组在高斯消元后只有 $n - 1$ 个方程，那我们只需要令 $c_n^* = 0$ 即可回代求出特解。

总结：玩铁道玩的。

The End

希望各位在本次比赛中都有学习到知识！