

2026 NUIT-ACM 新生选拔赛

题解

Kendieer

2026-10-11

预期难度

Easy : K, F, B

Mid — Easy : E, A, C, L

Mid : I, M, J

Mid — Hard : H

Hard : D, G

所有题目时空限制均在标准程序 3 倍及以上。

K. exString

题意

给定一个串 s , 你需要构造一个长度恰好为 n 的字符串 t , 使得 t 存在子串 s 。

数据范围: $1 \leq |s| \leq n \leq 10^4$ 。

Author: Kendieer

K. exString

输出原串，然后手动补齐其他字符即可。

F. 鼠鼠的开销

题意

给定已知预算 $n + m \times k$, 给出 t 个物品的开销 $a_1, a_2, \dots, a_t$, 求按顺序贪心情况下能够支付哪些物品。

数据范围: $0 \leq n, m, k \leq 100, 1 \leq t \leq 100, 1 \leq a_i \leq 100$ 。

Author: hushuqi_qwq

F. 鼠鼠的开销

签到题，按题意模拟即可。

B. Combo

题意

给定 n 个正整数 a_i , 可以选择若干个数, 使得乘积最大化。输出这个答案对 998244353 取模后的结果。

数据范围: $1 \leq n \leq 2 \times 10^5, 1 \leq a_i \leq 10^9$ 。

Author: Kendieer

B. Combo

由于元素都是正整数，所以考虑直接选取所有元素。

由模运算的结合律可得：

$$a \cdot b \bmod p = (a \bmod p) \cdot (b \bmod p) \bmod p。$$

直接输出 $\prod_{i=1}^n a_i \pmod{998244353}$ 的结果即可。

E. 环形操作

题意

已知一个初始大小为 n 且初始值均为 1 的环形数组 a 。每次操作选择一个下标 i ，令 a_{i-1} 和 a_{i+1} 分别 -1 ，令 $a_i + 2$ 。你需要每次操作后保证每个元素都是非负整数。在采用尽可能多操作次数的策略下，求最终数组可能的状态。

数据范围： $3 \leq n \leq 2 \times 10^5, \sum n \leq 2 \times 10^5$ 。

Author: Kendieer

E. 环形操作

当 n 为奇数时, 设第一次操作的下标为 i , 此时我们的操作序列为 $i+1, i-1, i+2, i-2, \dots, i+k, i-k, i+k-1, i-k+1, i+k-2, i-k+2, \dots$, 直到无法操作为止。

(1, 1, 1, 1, 1)
(1, 0, 3, 0, 1)
(0, 2, 1, 2, 0)
(0, 0, 5, 0, 0)

$n = 5$ 情况的示意

所以最终序列为 $(n, 0, 0, 0, \dots)$ 。

E. 环形操作

当 n 为偶数时，我们仍然使用类似奇数的操作序列进行操作：

$(1, 1, 1, 1, 1, 1)$
$(1, 0, 3, 0, 1, 1)$
$(0, 2, 1, 1, 2, 0)$
$(0, 0, 3, 3, 0, 0)$

$n = 6$ 情况的示意

所以最终序列为 $(\frac{n}{2}, \frac{n}{2}, 0, 0, 0, \dots)$ 。

A. X-GCD

题意

进行 T 组查询, 每次查询 $\gcd(a^b, b^a)$ 对 998244353 取模后的结果。

数据范围: $1 \leq T \leq 10^5, 1 \leq a, b \leq 10^6$ 。

Author: Kendieer

A. X-GCD

考虑 gcd 的定义, 设:

$$a = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_n^{\alpha_n}$$

$$b = p_1^{\beta_1} p_2^{\beta_2} \dots p_n^{\beta_n}$$

则有 $\gcd(a, b) = \prod_{i=1}^n p_i^{\min(\alpha_i, \beta_i)}$ 。

由于每个数 x 至多有 $\log(x)$ 个质因子, 所以跑一次线性筛预处理后直接在 $O(\log^2(V))$ 复杂度内质因数分解加暴力枚举, 直接求出 $\prod_{i=1}^n p_i^{\min(a \cdot \beta_i, b \cdot \alpha_i)}$ 即可。

时间复杂度为 $\mathcal{O}(V + T \log^2(V))$, 其中 $V = 10^6$ 。

C. 检查点

题意

给定 n 个检查点 p_i 和 m 个棋子的位置 x_j ，现在可以对每个棋子至多移动一次，移动距离为 d 所需代价为 d^2 。现在已知至多可以使用 k 的代价，求最小的整数 r 使得每个棋子移动后都在检查点距离 r 以内。

数据范围： $1 \leq n, m \leq 2 \times 10^5, 0 \leq k \leq 10^{16}, 0 \leq |x_i|, |p_i| \leq 10^8$ 。

Author: Kendieer

C. 检查点

代码细节题。

考虑二分答案，我们可以将每个棋子最近的检查点距离预处理出来，然后贪心移动到边缘即可。

注意最大代价为 $(2 \times 10^8)^2 \times 2 \times 10^5 > 2 \times 10^{19}$ 。

遇到不符合的情况下要及时退出，否则会出现爆 unsigned long long 的情况。也可以使用 int128 解决这个问题。

时间复杂度为 $\mathcal{O}(n(\log V + \log n))$ ，其中 $V = 2 \times 10^8$ 。

L. MEX 子数组

题意

给定一个由 $0 \sim n-1$ 组成的排列 p , 请求出对于排列 p 所有非空子数组 b 的 MEX 值总和。

数据范围: $1 \leq n \leq 10^6, \sum n \leq 10^6$ 。

Author: Kendieer

L. MEX 子数组

对于每个满足 MEX 值大于等于为 x 的区间中，必然属于 MEX 值大于等于 $x - 1$ 的所有区间。

考虑拆分贡献，我们可以将 $\text{MEX}(a) = x$ 拆分成 $\text{MEX}(a) = \sum_{x \geq 1} [\text{MEX}(a) \geq x]$ 。

由 $\text{MEX}(a) \geq x$ 的区间必然属于 $\text{MEX}(a) \geq x - 1$ 的区间集合，且因为给定的序列是排列，可以根据值为 x 的元素位置继续约束区间范围。

设 $f(x)$ 为 $\text{MEX}(a) \geq x$ 的区间个数，则答案就是 $\sum_{i=1}^n f(i)$ ，而 $f(i) = \min(\text{idx}_1, \text{idx}_2, \dots, \text{idx}_i) \cdot (n - \max(\text{idx}_1, \text{idx}_2, \dots, \text{idx}_i) + 1)$ ，该过程可以线性复杂度内求出。

于是答案可以在 $\mathcal{O}(n)$ 的时间复杂度内求出。

I. 安全传输

题意

通信题。

给定一个正整数 x ，你需要用长度不超过 50 的小写字母进行传输，但得到的小写字母是乱序的。

数据范围： $1 \leq x \leq 10^9$ 。

Author: Kendieer

I. 安全传输

显然，这题不能用简单的二进制进行表示数字，因为 $a \sim z$ 字符集数量不够，考虑优化。

一种简单的优化是将二进制优化成三进制，这样可以用至多 $2 \times 19 = 38$ 个字符完全表示。

另一种优化是考虑压位二进制表示一个数，即将最后 5 位用二进制表示对应状态，此时需要 $20 + 5 \times 3 = 35$ 个字符表示。

M. 巴巴博弈

题意

有 n 堆石子，每堆石子数为 a_i ，两名玩家轮流取石子，每次可以选择若干堆石子，取走其中一颗石子。问先手是否存在必胜策略。
数据范围： $1 \leq n \leq 5 \times 10^5, 1 \leq a_i \leq 10^9$ 。

Author: Kendieer

M. 巴巴博弈

当 n 堆石子数量均为偶数时，先手必败，否则先手必胜。

若存在奇数堆，先手在所有奇数堆各取一颗，局面变全偶；全偶是必败态（后手任意操作必产生奇数堆，先手再同样处理）

时间复杂度为 $O(n)$ 。

J. 划分区间

题意

给定两个长度为 n 的序列 a 和 b , 你可以对该序列划分成若干个区块, 每个区块的权值为 $\sum_{j=l}^r a_j \cdot b_i$.

求恰好划分成 $1, 2, \dots, n$ 个区块时, 最大可能权值之和是多少。

数据范围: $1 \leq n \leq 5000, 0 \leq |a_i|, |b_i| \leq 10^6, \sum n^2 \leq 5000^2$ 。

Author: Kendieer, KAZHYI

J. 划分区间

考虑动态规划，记 $dp_{i,j}$ 表示前 i 个元素恰好划分成 j 个区块下最大权值和。

则有 $dp_{i,j} = \max(dp_{i-1,j}, dp_{i-1,j-1}) + a_i \cdot b_j$ 。

可以发现这个过程可以用滚动数组优化以免内存超限。

时间复杂度为 $\mathcal{O}(n^2)$ ，空间复杂度为 $\mathcal{O}(n)$ 。

H. 野树的曙

题意

给定一棵由 n 个顶点组成的树和一个由 n 个元素组成多重集 S , 现在需要将多重集元素分配到每个点, 作为其点权, 并使得两两路径点权之和最大化。

数据范围: $2 \leq n \leq 2 \times 10^5, \sum n \leq 2 \times 10^5, 1 \leq s_i \leq 10^9$ 。

Author: Kendieer

H. 野树的曙

考虑拆分贡献，将每个节点拆分成该点能被经过的次数，此时我们只需按照顶点被经过的次数从大到小进行填数即可。

求节点被经过的次数有很多，一种简单的方式进行树形 dp，即 $c_u = siz_u(2n - siz_u) - \sum_{v \in son(u)} siz_v^2$ 。

时间复杂度为 $O(n \log n)$ ，瓶颈在排序。

D. 组合数乘积

题意

计算由 n 个组合表达式 $C_{a_1}^{b_1} \cdot C_{a_2}^{b_2} \cdots C_{a_n}^{b_n}$ 的乘积, 对 m 取模。
数据范围: $1 \leq n \leq 10^6, 2 \leq m \leq 10^9, 0 \leq b_i \leq a_i \leq 10^6$ 。

Author: Kendieer

D. 组合数乘积

由于不能保证模数是质数，所以不能使用逆元。

考虑拆分 $\frac{a!}{a!(a-b)!}$ ，手动消除对应项数的因子，最终答案一定是 $\prod_{i=1}^V i^{k_i}$ ，其中 k_i 可以使用差分 + 后缀和完成。

这种做法的时间复杂度为 $O(V \log n)$ 。

D. 组合数乘积

实际上，这题还存在线性复杂度做法。

对于拆分后的因子幂次，我们考虑对每个因子继续质因数分解，即设 $x = p \cdot y$ ，其中 p 为质因子，则可以根据 $x^{k_x} = y^{k_x} \cdot p^{k_x}$ 继续分解。

于是我们只需对质数进行快速幂即可优化到线性复杂度，即 $\mathcal{O}(\frac{V}{\ln V} \log n)$ 。

G. 迟到的色彩刚刚好

题意

给定 n 个顶点组成的树，给定若干个标记点， q 次查询，每次查询 $x \rightarrow y$ 至少经过一个标记点的最短距离。

数据范围： $1 \leq n, q \leq 2 \times 10^5, 1 \leq x, y \leq n$ 。

Author: Kendieer

G. 迟到的色彩刚刚好

考虑对于每个顶点 u , 预处理出该点到达最近的标记点距离 w_u ,
考虑对于每个标记点作为源点, 跑一次多源 BFS 即可完成。

那么对于每次查询 x, y , 其答案为 $\text{dist}(x, y) + \min_{u \in \text{path}(x, y)} 2w_u$,
其中 dist 表示两点之间的距离。

对于该式子, 可以直接维护 LCA 的过程中顺便求出。

实现方式有很多, 可以采用树上倍增或者树链剖分 + ST 表求出。

时间复杂度均为 $\mathcal{O}((n + q) \log n)$, 空间复杂度为 $\mathcal{O}(n \log n)$ 。

后言

致所有人

无论你处在哪个阶段，如果这场比赛没有发挥好，都完全是正常的事情。

一场比赛只是一个人在某个时间点、某个方向上的能力投影，不能定义你的潜力，也不能决定你的未来。请不要因为一次比赛的结果而气馁。坚持训练，认真复盘，持续积累经验，你一定会在一次次练习中逐渐提高。

致谢

本场比赛感谢以下所有工作人员（按字典序排序）

- 出题人: KAZHYI, Kendieer, hushuqi_qwq
- 验题人: 7_divided_by_3, Anoth3r, CirnoNine, Forest5326, LiangBaiKai, Liuguang_Ji, Luyi16_no_head, MaxBlazeResFire, Misumi_Ichika, Silencer76, StarDreamAurora, TRfirst, YESHU1, abovecloud, big-mktx, chouc, hushuqi_qwq, ptrtonull, qianwai, qwynick, tfgkk, usedchang, xiaopacai2.0, xuan3332, yangzhou0629, yuki_lshijo, zhangfengya
- 后勤工作: KAZHYI, Kendieer
- 特别感谢: hushuqi_qwq, usedchang

Thank You!