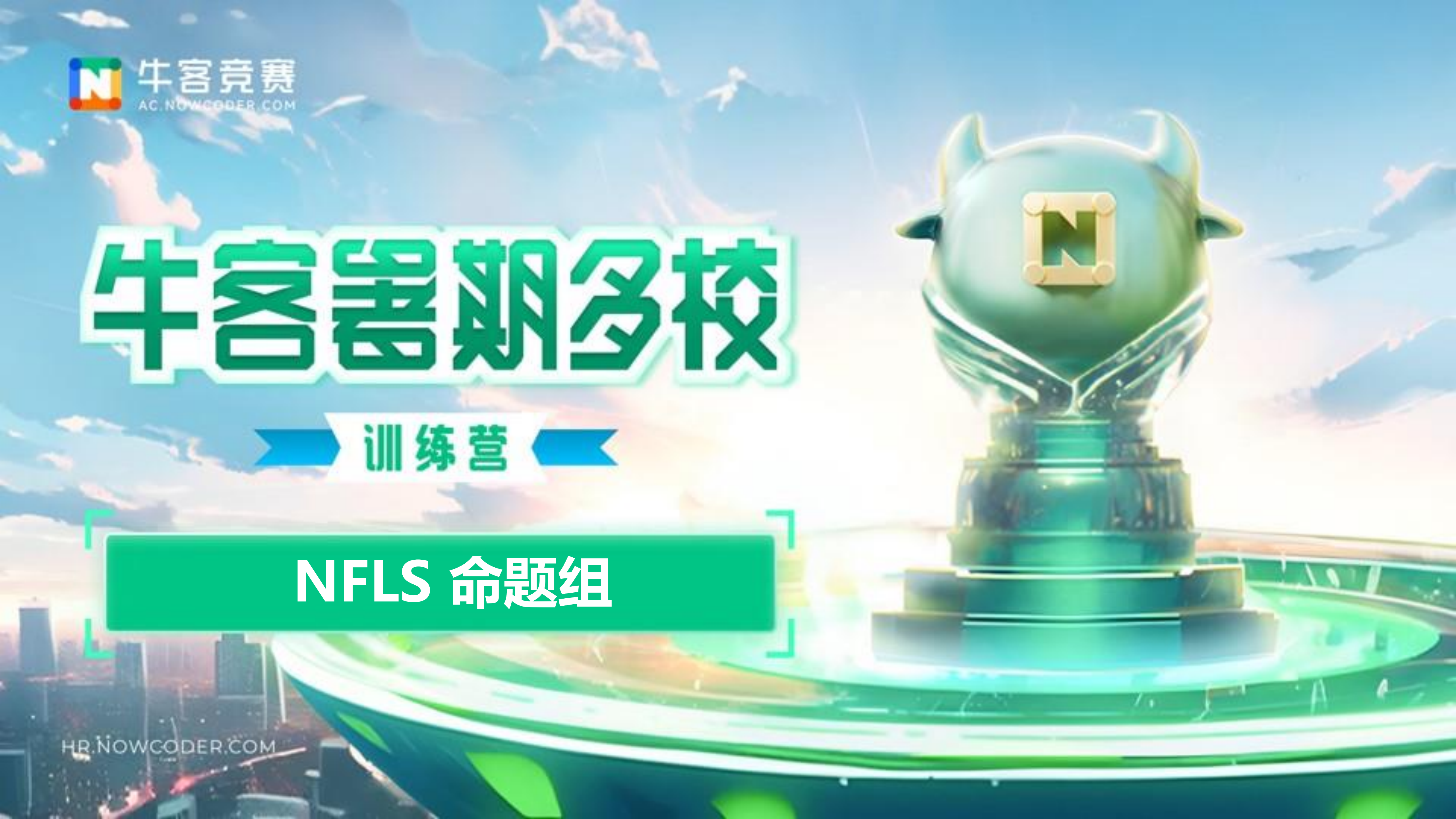
牛客竞赛
AC.NOWCODER.COM
牛客暑期多校
训练营
NFLS 命题组
HR.NOWCODER.COM

A - Random Addition

题目描述

给定 m 个区间 $1 \le l_i < r_i \le n$，满足两两要么包含要么不交。

有一个初始均为 0 的长度为 n 的数组 a，每次随机选择一个 $0 \sim 1$ 之间的实数 x 并将 $a_{l_i} \sim a_{r_i}$ 均加上 x。

t 次询问，每次两个正整数 p, q，表示求 a 中的最大值在 $p \sim q$ 之间的概率，对 998244353 取模。

A - Random Addition

线段不相交，容易发现可以抽象为一棵树。每一个节点对应一个线段，也就是一次 $+x$。

设 $f_i(x)$ 为以 i 节点为根的子树内（即操作完所有 i 子树内的操作后）最大值不超过 x 的概率。不难发现对于叶子节点:

- $0 \le x \le 1$ 时: $f_i(x) = x$。
- $1 < x$ 时: $f_i(x) = 1$。

考虑每个节点 i 如何从子节点转移。当 i 节点选择 y 时，容易发现最大值不超过 x 的概率为:

$$\prod_{v \in son_i} f_v(x-y)$$

容易发现 $0 \le y \le 1$，于是:

$$f_i(x) = \int_{\max(0,x-1)}^{x} \prod_{x \in son_i} f_v(y) \mathrm{d}y$$

归纳可证 $f_i(x)$ 是一个有关 x 的分段函数，且分段点为整数。

由于原题中 $0 \le p < q \le 10$，只需要维护对于区间 $0 \le x \le 1, 1 \le x \le 2, \ldots, 9 \le x \le 10$ 的 $f_i(x)$ 即可。转移时需要多项式乘法，根据树上依赖背包的证明为 $O(10m^2)$ 的。对于每个点需要展开 $\prod f_v(x-1)$，复杂度是 $O(m^2)$ 的（多项式科技可以做到 $m \log m$），于是这部分复杂度是 $O(10m^3)$ 的.

B - Tree Climber

题意简述

给出一棵 n 个点的树和 m 个操作，每个操作给出 l, r, x，表示将原树上 $l, l+1, \cdots, r$ 这些点的虚树中所有边的边权加上 x。求出最终每条边的边权。

$1 \leq n \leq 3 \times 10^5, 1 \leq m \leq 10^6$。

B - Tree Climber

题解

随便取树上一个点作为根，不妨取 1。

一次操作能影响到 x 到 x 的父亲这条边，当且仅当 $[l, r]$ 中既有 x 子树内的点，也有子树外的点。

考虑对于每一条边，统计所有不能影响到它的操作的 x 和，用所有操作的和减去它即是这条边的答案。这样的操作就是，区间内所有的点都在子树内，或都在子树外。

考虑 dsu on tree，过程中维护一个序列 $a_i = 0/1$ 表示 i 是否在当前点的子树内，并对于 a_i 中每一个全 0 或全 1 的连续段，求出左右端点都在这个连续段内的操作的 x 的和，维护所有连续段的这个值的和。

B - Tree Climber

考虑一个 a_i 从 0 变成 1 时，有什么影响。

- 若 $a_{i-1}=a_{i+1}=0$，即分裂了一个全 0 的连续段 $[L,R]$，那么需要减去左端点在 $[L,i]$、右端点在 $[i,R]$ 中的操作的 x 的和，再加上左右端点都为 i 的操作的 x 的和。
- 若 $a_{i-1}=a_{i+1}=1$，即合并了两个全 1 的连续段 $[L,i-1],[i+1,R]$，类似第一种情况，加上左端点在 $[L,i]$、右端点在 $[i,R]$ 中的和，减去 $[i,i]$ 的和。
- 若 $a_{i-1}=1,a_{i+1}=0$，即给全 1 连续段 $[L,i-1]$ 右边添加了一个 1，设右边的全 0 连续段为 $[i+1,R]$，那么加上右端点为 i，左端点在 $[L,i-1]$ 中的和，减去左端点为 i，右端点在 $[i+1,R]$ 中的和。
- $a_{i-1}=0,a_{i+1}=1$ 类似。

B - Tree Climber

总之，我们需要支持的就是一个二维数点，具体来说，对于每个操作在平面上 (l, r) 的位置加上 x，dsu on tree 时要求矩形和。用主席树维护即可。

时间复杂度 $\mathcal{O}(n\log^2 n + m\log n)$。

C - Beautiful sequence

题目描述

你需要求出字典序第 k 小的长度为 n 的单调不降序列 $A=(A_1,A_2,\dots,A_n)$，即 $A_1 \le A_2 \le \cdots \le A_n$，满足:

- 对于每一个 A_i（$1 \le i \le n$），$0 \le A_i < 2^{30}$。
- 对于每一个 $1 \le i < n$ 都满足 $A_i \oplus A_{i+1} = B_i$，其中 B 为给定序列，$\oplus$ 表示按位异或。

C - Beautiful sequence

如果确定了某个 A_i 的二进制下的某一位，那么所有 A_i 的这一位就都知道了。

若 $B_i = 0$，则 $A_i = A_{i+1}$，满足条件。

否则找到 B_i 的最高位 1，可以得出 A_i 的这一位是 0，然后就可以确定所有 A_i 的这一位。矛盾很好判断。

接下来令 c 为还没被确定的位的数量，那么 A 共有 2^c 种构造方法，如果 $2^c < k$ 就直接输出 -1。

从低到高填充 A_1 中还没被确定的位，第 i 次使用 k 的第 i 低的位进行填充。然后整个 A 数组就能确定出来了。

D - Game on Tree

题目大意

- 有一个字符串集合 S，Alice 和 Bob 轮流操作，Alice 先手。
- 每次选择一个 S 中的字符串 s 从 S 中删除，并选择一个在 s 中出现过的字符 c。
- 将 s 沿着 c 出现的位置划分成若干子串，将非空的加入进集合 S。
- 不能进行操作的人输。

D - Game on Tree

题目大意

- 现在给定一棵树，每个节点上有字符。
- 记 $\mathrm{path}(u, v)$ 表示将节点 u 到 v 的最短路径所经过的节点（包括 u 和 v）上的字符依次拼接而成的字符串。
- 共 q 次询问，每次询问给定 u 和 v，求 $S = \{\mathrm{path}(u, v)\}$ 时的胜者。
- 若 Alice 获胜，同时输出第一步有多少种走法。

题解

- 考虑在当前串 s 的末尾添加一个字符 c，会对 SG 值造成什么变化。
- 整串的 SG 值 f 的变化不易描述，因此记 f_i 表示对字符 i 进行操作后，得到的子串集合的 SG 值异或和。
- len_i 表示 i 上次出现的位置之后有多少个字符，未出现则为 -1。
- 显然 $f = \text{mex}_{len_i \neq -1}\{f_i\}$。

题解

- len_i 容易维护。那么考虑加入字符 c 后，f_i 会发生的变化：
- $i = c$，若 $len_i \neq -1$ 则 $f_i \leftarrow f_i$ 无变化。否则，$f_i \leftarrow f$，即修改前整串的 SG 值。
- $i \neq c$，若 $len_i = -1$ 则未定义。否则，这涉及到 s 长度为 len_i 的后缀添加字符 c 后如何变化。

题解

- 记 s_i 表示 s 长度为 len_i 的后缀。记 g_i 表示 s_i 的 SG 值。
- 类似地，记 g_{ij} 表示在 s_i 对字符 j 进行操作后，得到的子串集合的 SG 值异或和。
- 当 $i \neq c$ 时，直接修改 s_i 的异或和显然是正确的。即 $f'_i \leftarrow f_i \oplus g_i \oplus g'_i$。
- 那么 g_i 的修改也是和 f 类似的。
- 注意到只有当 $len_j \neq -1$ 且 $len_j < len_i$ 时，g_{ij} 才有定义，因此 g_i 的修改只依赖于满足上述条件的 g_j，转移不会成环。
- 末尾添加字符，$\mathcal{O}(|\Sigma|^2)$。开头添加字符也类似维护。

题解

- 现在考虑如何合并两个串 s 和 t 的信息。
- 记 h_{ij} 表示 $s_i^{[s]} + t_j^{[p]}$ 的 SG 值。
- 计算时枚举这一步操作的字符 k，发现与 $sg_{ik}^{[s]} \oplus tg_{jk}^{[p]}$ 不同的只有跨过 s 和 t 分界点的那一段 h_{kk} 或 h_{ik} 或 h_{kj}。转移不会成环。
- 计算出 h_{ij} 后，容易计算出新的 f 和 g。
- 复杂度显然是 $\mathcal{O}(|\Sigma|^3)$。

题解

- 考虑最终答案的计算。
- 使用点分治求出 $u \to p$ 和 $p \to v$ 的信息，其中 p 为 u 到 v 在点分树上的 LCA，最后再合并。
- 时间复杂度 $\mathcal{O}(n\log n|\Sigma|^2+q|\Sigma|^3)$，空间复杂度 $\mathcal{O}(n+q|\Sigma|^2)$。
- 计算第一步的走法即 $f_i=0$ 的个数。
- 细节比较多，写的时候要注意下转移顺序。

E - Star wars

题目描述

给你一个 n 个点的无向图，每个点都有权值。需要支持加边删边和权值修改还有查询一个点相邻的点的权值和。

E - Star wars

使用启发式合并的思想可以想到一种神奇的做法。下面用 m 表示边的个数。

考虑给所有边定向。我们维护一个 c 数组，其中 c_i 表示点 i 的出度。

每加入一条新边 (x, y) 时，我们记 x 为 c_x, c_y 的较小值对应的点。我们令这条边是一条从 x 连向 y 的有向边。可以证明 c_i 的最大值不会超过 $\sqrt{m}$。

我们再维护每条边出现的次数和一个 d 数组，d_i 表示所有连接到 i 的点的权值之和。

对于每次加边操作，将这条边的出现次数 $+1$，如果在此操作之前不存在这条边，需要更新 d。如果这条边是第一次出现，按照上面说的做一遍。

对于每次删边操作，将这条边的出现次数 -1，如果在此操作之后不存在这条边，需要更新 d。

对于每次权值修改操作，我们遍历 x 的所有出边 y，令 $d_y \leftarrow d_y + c$。

对于每次查询操作，我们先初始化 $ans \leftarrow d_x + a_x$，然后遍历 x 的所有出边 y，令 $ans \leftarrow ans + a_y$，其中 a_x 表示 x 的兵力。这里的 ans 就是答案。

然后这道题就做完了，时间复杂度 $O(q\sqrt{m})$。

F - Counting sequences

题目描述

计算同时满足下面两个条件的序列 $A = (A_1, A_2, \ldots, A_n)$ 的个数对 998244353 取模后的结果：

- 对于每一个 A_i（$1 \le i \le n$），$0 \le A_i < 2^m$。
- 我们令序列 B 为序列 A 旋转一次后得到的序列，

$$\sum_{i=1}^{n} cnt_1(A_i \oplus B_i) = k$$

其中 $\oplus$ 表示按位异或，一次旋转操作会将 $(A_1, A_2, \ldots, A_{n-1}, A_n)$ 变为 $(A_2, A_3, \ldots, A_n, A_1)$。$cnt_1(x)$ 表示 x 在二进制表示下 1 的个数。

F - Counting sequences

nm 不是 4 的倍数时答案为 0。

$C = A \oplus B$ 固定下来后，如果 $C_1 \oplus C_2 \oplus \cdots \oplus C_n = 0$，那么 A 有 2^m 种构造方法，否则无解。

把 C 在二进制表示下的 m 位分别抽出一个 01 序列，每个都要满足有偶数个 1，总共要有 k 个 1。

令 $f(x) = 1 + \binom{n}{2}x + \binom{n}{4}x^2 + \ldots$

$f^m(x)$ 的 x^k 项的系数就是答案。

多项式快速幂 $O(k \log k \log m)$

G - Cyperation

题目描述

给定一个长度为 n 的环形数组 a 和一个正整数 k，你可以执行下述操作任意多次（可以不执行）：

- 选定两个下标 i 和 j （$1 \le i < j \le n$）满足 $\min(j-i, n+i-j) = k$，即 i 和 j 在环形数组上的最短距离为 k，并且将 a_i 和 a_j 都减去 1。

判断最后能否将整个数组变成 0。如果可以，输出 `YES`；否则输出 `NO`。多测。

G - Cyperation

题解

首先如果 $k > \lfloor \frac{n}{2} \rfloor$，那么我们做不了任何操作，当且仅当所有 a_i 均为 0 的时候才可以，否则不可。

否则显然我们可以将这个数组抽离出若干个环。如：对于 $n = 8, k = 3$ 的情况，可能的 (i, j) 只有 $(1, 4)$，$(2, 5)$，$(3, 6)$，$(4, 7)$，$(5, 8)$，$(1, 6)$，$(2, 7)$，$(3, 8)$ 这几种。那么我们显然就可以把这个东西转化成 $a_1, a_4, a_7, a_2, a_5, a_8, a_3, a_6$，如此操作就变成了对这个环形数组上相邻的两数同时减 1。

考虑对于某个环怎么做。即对于一个长为 m 的环形数组 b，每次可以将相邻两数同时减去 1，求能否全部变成 0。

考虑分 m 为奇数和偶数两种情况讨论。

m 为奇数相对简单。设 1 和 2 减了 x_1，2 和 3 减了 x_2，$\dots$，$m-1$ 和 m 减了 x_{m-1}，1 和 m 减了 x_m。则 $b_1 = x_1 + x_m$，$b_2 = x_1 + x_2$，$b_3 = x_2 + x_3$，$\dots$，$b_{m-1} = x_{m-2} + x_{m-1}$，$b_m = x_{m-1} + x_m$。然后我们发现 $x_1 + x_2 + \dots + x_m = \frac{\sum_{i=1}^{m} b_i}{2}$，据此就可以解出来所有的 x_i。只需要判断 x_i 是否为整数且是否 ≥ 0 即可。

G - Cyperation

那么 m 为偶数也是差不多的，但是你无法直接得出每个 x_i 的值。但还是有一点是需要满足的：$b_1+b_3+...+b_{m-1}=b_2+b_4+...+b_m$。如果不行显然无解。

不妨换一个角度思考问题。用含有 x_1 和 b_i 的式子表示出每个 x_i，看看如果要 $x_i \geq 0$ 需要 x_1 满足什么条件，最后看存不存在非负整数 x_1 满足条件即可。这是容易的：

$x_2=b_2-x_1$。即 $x_1 \leq b_2$。

$x_3=b_3-x_2=b_3-b_2+x_1$，即 $x_1 \geq b_2-b_3$。

同理，$x_1 \leq b_2-b_3+b_4$，如此往后，一直到 x_m。我们求出不等号右边的值，然后对于 $\geq$ 和 $\leq$ 都取最值，最后看 $\max$ 是否 $\leq \min$ 即可。总复杂度 $O(n)$
。

H - Mountain View

题意

- 对于一段折线，定义它的**蓄水面积**为所有前缀 max 向右作水平射线与折线（或紧靠折线右端的竖直线）围成的面积之和。
- 单点修改，区间查蓄水面积。

H - Mountain View

解答

- 注意到问题不弱于 [单点修改，区间查前缀 max 数目]，考虑楼房重建的做法。
- 对横坐标建线段树，结点维护纵坐标的最大值和该区间蓄水面积。
- Pushup 相当于解决这样的子问题：给出折线左侧水面高度查询蓄水量。
- 这一子问题同样可以在线段树上查询，只需讨论左侧水面高度和

[l, mid] 间纵坐标最大值的大小关系即可。

- 递归边界是一个简单的相似三角形求面积。
- 复杂度 $O(n\log^2 n)$（n,q 同阶）

Bonus

- tester 给出了一个复杂度更优的解法。
- 类似【UR #19】前进四 的 trick：本题并未强制在线，且只有单点修改，可以对时间轴建支持区间最值操作的线段树，扫描横坐标做到单 log.

I - We love strings

- 题意简述

有 n 个仅包含 01? 的正则表达式，求有多少个 01 字符串能被至少一个这些正则表达式匹配。

$1 \le n, \sum |s_i| \le 400$

I - We love strings

- 题解

数据范围很奇特，考虑对于正则表达式长度数据分治：

1. 对于 $|s_i| \le B$，我们暴力枚举每一个长度不超过 B 的字符串并检验他们是否合法，复杂度 $\Theta(2^B \sum |s_i|)$。
2. 对于 $|s_i| > B$，我们考虑枚举至少有哪些字符串能匹配上最终的结果，将枚举的字符串合并后答案即为 $2^{问号个数}$，容斥即可。复杂度 $\Theta(2^{\frac{n}{B}} \sum |s_i|)$

平衡一下，求得当 $B = \sqrt{n}$ 时最优，复杂度即为 $\Theta(2^{\sqrt{n}} \sum |s_i|)$。

J - Is it a tree

- 题意简述

给定一棵树和一个点仙人掌，定义 $f(i,j)$ 表示仅考虑仙人掌上编号出现在树上 $i \to j$ 路径的点，会剩下多少个连通块。定义一棵树 S 的权值为 $\sum_{i \in S} \sum_{j \in S} [i \le j] f(i,j)$。

对于给定树的每个子树，求出他的权值。

$1 \le n \le 5 \times 10^5$

J - Is it a tree

- 题解

由于点仙人掌森林的连通块数=点数-边数+环数，我们只需要求出点数总和-边数总和+环数总和即可。分类计算点、边、环的贡献：

1. 点的贡献：即求子树内所有路径上点的数量之和。可以简单树形 DP 求得。
2. 边的贡献：边相当于大小为 2 的环，贡献取反。

J - Is it a tree

3. 环的贡献：环 $a_1, a_2 \ldots, a_k$ 若有贡献，则 $a_1, a_2 \ldots, a_k$ 在树上必然在一条链上，判断点集是否在一条链上可以使用贪心：深度最大的点必然是端点之一，于是距离他最远的点的必定是另一个端点，之后判断剩下的点是否在这两点间的路径上即可。设路径的端点为 u, v。我们分类计算两种环的贡献，祖孙环（任意两点都有祖先后代关系）和普通环。普通环的贡献是好算的，$\texttt{lca}(u, v) \to 1$ 的路径上的点的贡献都会加上 $sz_u \times sz_v$。祖孙环的贡献会略难算，不妨令 $dep_u < dep_v$，设 u 向 v 走一步后到达的点是 w。则对于一个大小为 x 的能够包含这个环的子树，这个环的贡献为 $(x - sz_w)sz_v$ 拆开即得 $sz_v x - sz_w sz_v$ 。这是一个关于子树大小的一次函数，我们将一次项系数和零次项系数分开计算即可。

注意到所有贡献都是 $1 \to u$ 的路径加的形式，容易用树上差分做到线性。

实现中的 LCA 和找 w 的过程可能会多一个 $\log$，但实际上可以用 tarjan LCA 和长链剖分优化掉，所以精细实现是可以做到严格 $\Theta(n)$ 的。

K - Set

题意简述

给定一个长度为 n 的数列 a_i ，求 $\sum_{T\subseteq[1,n]} |T| \max_{i\in T} a_i \min_{i\in T} a_i \bigoplus_{i\in T} a_i$

$1 \le n \le 10^6, 1 \le a_i < 2^{31}$

K - Set

题解

单独处理 $|T| = 1$ 的情况

对 a 排序，此后 $\max\limits_{i \in T} a_i = a_{\max\limits_{i \in T} i}$ ， $\min$ 亦然

枚举 $L = \max_{i \in T} i$ 和 $R = \min\limits_{i \in T} i$

按二进制位分离 $\bigoplus\limits_{i \in T} a_i$

对排序后下标在 (L, R) 内且当前位为 1 的数的数量分三类讨论：有 0 个或有 1 个或至少有 2 个

使用前缀和优化后复杂度$\Theta(n \log a_i)$

L - Misaka Mikoto’s dynamic KMP problem

题意简述

给定模式串 s，需要支持两种操作：

1. s 单点修改；
2. 给出文本串 t，求 s 在 t 中的出现次数和 s 最长 border 的积。

强制在线。

$|s|, m \leq 10^6, \sum |t| \leq 2 \times 10^6$

L - Misaka Mikoto's dynamic KMP problem

注意到 $|s| > |t|$ 时出现次数必定为 0，所以该询问的 $x_i \times y_i = 0$。而当 $|s| \le |t|$ 直接暴力跑 KMP 求出现次数和最长 border 即可。单轮复杂度 $O(|t|)$。

M - Writing Books

题意简述

T 组测试，每次查询 $1 \sim n$ 中所有数十进制下的位数和。

$1 \leq T \leq 10^5, 1 \leq n \leq 10^9$

M - Writing Books

题解

设 n 是 m 位数。

位数小于 m 的所有数都能被统计入贡献，可以扫描每一个 $i < m$，将位数为 i 的数的个数 $9 \times 10^{i-1}$ 乘上 i 加入答案。对于位数为 m 的数，能被计入贡献的有 $n - 10^{m-1} + 1$ 个，乘上 m 加入答案即可。

记得开 `long long`。

时间复杂度 $\mathcal{O}(T \log n)$。

THANKS

AC.NOWCODER.COM