CyreneOI Round2 Div2.5

MSANPU OI Contest

Qaaxaap

题解

题目名称	Tribios	SkeMma720	EpieiKeia216	Hyacinthia	NeiKos496	PhiLia093
题目类型	传统型	传统型	传统型	传统型	传统型	传统型
目录	Tribios	Anaxagoras	Castorice	Hyacinthia	Phainon	Cyrene
可执行文件名	Tribios	Anaxagoras	Castorice	Hyacinthia	Phainon	Cyrene
输入文件名 (.in)	Tribios	Anaxagoras	Castorice	Hyacinthia	Phainon	Cyrene
输出文件名 (.out)	Tribios	Anaxagoras	Castorice	Hyacinthia	Phainon	Cyrene
每个测试点时限	1.0 秒	1.0 秒	1.0 秒	2.0 秒	1.0 秒	1.0 秒
内存限制	256 MiB	256 MiB	256 MiB	256 MiB	256MiB	256MiB
测试点数目	10	10	20	20	10	10
测试点是否等分	是	是	是	是	是	是
是否使用子任务	否	是	否	是	是	否

提交源程序文件名 (.cpp)

对于 C++ 语言	Tribios	Anaxagoras	Castorice	Hyacinthia	Phainon	Cyrene

编译选项

对于 C++ 语言	-O2 -std=c++14 -static

注意事项（请仔细阅读）

1. 本场比赛为阶段学习评估，总体难度较小，与 Codeforces-Div2.5 难度相近。
2. 由于评测机性能原因，评测时题目时间限制可能会进行调整。
3. 时间限制、空间限制分别不小于标准程序运行时间和内存使用的 1.5 倍、1 倍。
4. 本场比赛仅支持 C++14 语言。
5. **不保证难度有序。**
6. 本校模拟赛题不得外传，包括但不限于：私自拷题，在任何网站记录题目信息（包括洛谷非公开云剪贴板，非公开题目等），向在线 AI 询问题目，或任何可能造成其他人员获取题目的操作。

Tribios（Tribios）

首先，我们需要理解“平数字序列”的定义。对于任意两个数字 a_i 和 a_j（$i < j$），满足：

$$|a_i - a_j| \leq \min(a_i, a_j)$$

这个条件可以进一步转化为：

$$\max(a_i, a_j) \leq 2\min(a_i, a_j)$$

即

$$\frac{a_i}{2} \leq a_{i+1} \leq 2 \times a_i$$

也就是说，任意两个相邻数字的值不能相差超过一倍。所以我们可以使用动态规划来维护。

我们维护一个动态规划数组 dp，其中 $dp[k]$ 表示以第 k 个数字结尾的最长“平衡数字序列”的长度。我们可以找到一个状态转移方程：

$$dp[k] = \max_{i<k \wedge |a_i - a_k| \leq \min(a_i, a_k)} (dp[i]) + 1$$

但是，直接计算这个状态转移方程的时间复杂度是 $O(n^2)$，显然无法满足 $q \leq 10^5$ 的要求。基于上述性质，我们可以将动态规划的状态转移方程优化为：

$$dp[k] = \max_{a_i \in [a_k/2, 2a_k]} (dp[i]) + 1$$

也就是说，对于每个新插入的数字 a_k，我们只需要在区间 $[a_k/2, 2 \times a_k]$ 内查找之前的最大 $dp[i]$ 值，然后加 1。

为了高效地处理区间查询和动态插入操作，我们可以使用线段树。具体来说：

每次插入一个新数字 a_k，以 k 为下标，插入 $dp[k]$ 的值

查询时，我们需要在线段树树中查询区间 $[a_k/2, 2a_k]$ 的最大值。

复杂度分析

插入操作的时间复杂度为 $O(\log n)$。

查询操作的时间复杂度为 $O(\log n)$。

总体时间复杂度为 $O(q \log n)$。

满足要求

SkeMma720（Anaxagoras）

定义 len 为访问长度，即 $r-l+1$ ，考虑到异或操作在每个二进制位之间互不影响，所以每一位分别讨论贡献，这里给出公式

$$\text{贡献}_k = 2^k \times 2^{\text{len}-1}$$

证明：设区间中有 c 个元素第 k 位上为 1 ，其余 $len-c$ 个元素为第 k 位 0

选择奇数个为 1 的元素，则方式总数为

$$\sum_{t\text{为奇数}} \binom{c}{t} = 2^{c-1}$$

下证 $\sum_{t\text{为奇数}} \binom{c}{t} = 2^{c-1}$

我们考虑二项式定理展开式，则有

$$(1+1)^c = \sum_{t=0}^{c} \binom{c}{t} = 2^c$$

$$(1-1)^c = \sum_{t=0}^{c} \binom{c}{t} (-1)^t = 0 \quad (\text{当} c \geq 1)$$

联立两式，即得

$$(1+1)^c + (1-1)^c = 2 \sum_{t\text{为偶数}} \binom{c}{t},$$

$$(1+1)^c - (1-1)^c = 2 \sum_{t\text{为奇数}} \binom{c}{t}.$$

化简即得 $\sum_{t\text{为奇数}} \binom{c}{t} = 2^{c-1}$

再任意选择 0 ，有 2^{len-c} 种选法

故第 i 位的贡献为

$$2^{c-1} \times 2^{\text{len}-c} = 2^{\text{len}-1}$$

但是最终我们要转换成 10 进制，故需要 $\times 2^k$ ，即得要证的公式 $Q.E.D$

注意到我们刚刚的公式在 $c \geq 1$ 的情况下才成立，所以当该位全是 0 的情况下该位贡献显然为 0

综上，答案 $= \sum_{k=0}^{29} \left([\text{第} k \text{ 位存在 } 1] \times 2^k \times 2^{\text{len}-1} \right) \mod (10^9+7)$

我们只需要使用 30 个带懒标记支持加乘操作的线段树来求和判断每一个二进制位是否存在 1 即可，异或 0 即为不变，异或 1 即为乘 -1 再加 1 ，如此实现即可，时间复杂度 $O(M \log n)$, 常数稍大

EpieiKeia216（Castorice）

形式化题面：给出序列 $a_1, a_2, \ldots, a_n$，从中选出 m 个元素 $a_{j1}, a_{j2}, \ldots, a_{jm}$，对于每个被选出来的元素 a_{jk}，其可以提供的贡献为 $a_{j1}, a_{j2}, \ldots, a_{jm}$ 中与 a_{jk} 的值相等的元素数量

如果一个数字出现了 t 次，其贡献就是 t^2。

我们可以转化一下，如果一个数 a 被选了 x 次，其贡献为 x^2。如果再选一次 a，贡献为 $(x+1)^2$，增加了 $2x+1$。因此，我们应该优先选择出现次数多的数，按照出现次数从大到小贪心即可。

Hyacinthia（Hyacinthia）

虽然这题长得有点像二分，但是好写的做法是 dp。

记 sum_u 表示 u 子树内人数之和，$leaf_u$ 表示 u 子树内叶子节点数量，那么考虑最理想的情况——均匀分配给每个叶子，那么答案至少是

$$\left\lceil \frac{sum_u}{leaf_u} \right\rceil$$

。

那什么情况下不是理想情况呢？记 f_v 表示 v 子树内分配后人数最多的叶子节点的最少数量，那么当 $f_v > \left\lceil \frac{sum_u}{leaf_u} \right\rceil$ 时无法均匀分配。否则，$f_u = \left\lceil \frac{sum_u}{leaf_u} \right\rceil$

$$f_u = \max\left\{\max_v f_v, \left\lceil \frac{sum_u}{leaf_u} \right\rceil\right\}$$

NeiKos496（Phainon）

形式化题面：

给定一个 n, 对于每个不超过 n 的正整数 i，用 n 除以 i，得到余数记为 n_1。用 n 除以 n_1，得到余数记为 n_2 ⋯⋯如此操作，直到出现一个 $n_m = 0$ 为止，此时记 $F_i = m$。

你的任务是对于给定的 n，以及 T 组 L, R，查询 $[L, R]$ 区间中的 $F_k(L \le k \le R)$ 最大值。

这道题的 f_k 的生成方式是不断对原数取余，不难想到递推递推公式是：$f_k = f_{a\%k} + 1$,$O(n)$ 即可接下来，我们要查询 (l, r) 区间中 f_k 的最大值，不难想到使用 ST 表

于是这道题就这么做完了, 时间复杂度 $O(n + t \log n)$

PhiLia093（Cyrene）

形式化题面：

给定一个长为 n 的数字序列 a_i，其中 $1 \le a_i \le n$。

取了 $m+1$ 个前缀，第 i 个前缀是 $1 \sim p_i$，其中 $p_{m+1} = n$。

设 z_i 为数字 i 的出现次数，

求 $(z_1 \times 1) \oplus (z_2 \times 2) \oplus \cdots \oplus (z_n \times n)$。

正解：

考虑维护出每个位置的出现次数，那么一个前缀 p_i 相当于给位置 $1 \sim p_i$ 加 1。这东西只需要维护差分数组，最后再前缀和一下就可以了。时间复杂度 $O(n+m)$，期望得分 100pts。

写在最后（总结）

此次题目难度估计约为蓝、蓝、橙、绿、绿、橙。

这告诉我们在比赛时**先读题估计题目难度再做题**。

另外，T1、T2、T5 都是我之前出过并且讲过的题，300 分里至少应该拿到 100 分，拿不到的应该好好反思。

谢谢倾听