数据结构

Larunatrecy

目录

CF1764H Doremy's Paint 2

题目描述

有一个长度为 n 的序列 a，初始时 $a_i = i$。
有 m 个操作，第 i 个操作将数列里 $a_{l_i} \sim a_{r_i}$ 都赋值成 a_{l_i}。
对于 $i = 1 \ldots m$，输出依次执行第
$i, i+1 \ldots (i+k-1) \bmod m + 1$ 个操作后序列里的颜色数。
$1 \le n, m \le 2 \times 10^5$。

CF1764H Doremy's Paint 2

破环成链，倒着考虑每个操作，维护 t_i 表示 i 位置的数字会存在到哪个时刻。

加入操作 i 时，产生的影响即是对于 $[l_i+1, r_i]$ 内的位置 j，$t_j = i$，而 $t_{l_i} = \max\limits_{j \in (l_i, r_i]} t_j$。

因为只有区间覆盖操作，所以用 set 维护连续段即可，维护答案可以再用一个树状数组，维护 $t_i \le v$ 的 i 的个数，复杂度 $O((n+m)\log n)$。

「2024 集训队互测」连通块

题目描述

给定一棵树，和一个初始为空的集合 S，先向 S 加入 m 次连通块，每次给定了 k_i 条边，把这些边断掉后形成的若干连通块都加入 S。接下来由 q 次询问，每次给定 u, r，询问有多少 S 内的连通块满足连通块内所有点到 u 的距离都 $\leq r$。

$1 \leq n \leq 10^5, \sum k_i, q \leq 3 \times 10^5$。

「2024 集训队互测」连通块

连通块内所有点到 u 的距离都 $\leq r \Leftrightarrow$ 到连通块的两个直径端点的距离均 $\leq r$。
而两个以直径端点为中心的半径为 r 的邻域的交是一个以直径中点为中心的半径为 $r - l/2$ 的邻域，其中 l 是直径长度。
使用点分树维护，每次是一个一维数点。

QOJ8028 Octasection

题目描述

给定 n 个三维立方体 $(x_l, y_l, z_l) - (x_r, y_r, z_r)$，你要从每一维选出一个坐标 x_p, y_p, z_p，使得每个立方体至少在一维上的坐标区间覆盖了对应的 $x_p/y_p/z_p$。

$n \leq 10^5$

QOJ8028 Octasection

题目描述

给定 n 个三维立方体 $(x_l, y_l, z_l) - (x_r, y_r, z_r)$，你要从每一维选出一个坐标 x_p, y_p, z_p，使得每个立方体至少在一维上的坐标区间覆盖了对应的 $x_p/y_p/z_p$。

$n \leq 10^5$

QOJ8028 Octasection

枚举 x_p，用数据结构维护 (y,z) 平面，这样每个立方体会在 x_l 处被删除，x_r+1 处被加入，而在 (y,z) 上对应的就是一个十字，我们要判断十字有没有交。

转化为求出补集的并，而补集的并可以看成是四条单调轮廓线，而每个立方体出现在轮廓线上的时刻是 $O(1)$ 个区间，因此可以用 set 维护轮廓线，用线段树维护每个 y 上的上轮廓线-下轮廓线，每次线段树二分找到第一个 >0 的位置输出即可。

[Ynoi1998] Marchen

题目大意

给定一个排列 a，每次询问区间 $[l,r]$ 内满足 $l \le i < j < k \le r \wedge a_i < a_j < a_k$ 的 (i,j,k) 数量。

$n, m \le 10^5$，强制在线。

[Ynoi1998] Marchen

分块，在不考虑空间的情况下，不妨先封装一个 $O(n\sqrt{n}) - O(1)$ 的可持久化分块，用来做在线二维数点。
设块长为 B，块 b 的左右端点是 st_b ed_b，i 所在块编号是 bel_i。

- 散块—散块—散块
  枚举 j，查询左边比 a_j 小的乘上右边比 a_j 大的，容易用可持久化分块 $O(1)$ 查询。
- 散块—散块—整块 / 整块—散块—散块
  和上面的一样的。
- 散块-整块-散块
  对于固定的 $i, k, a_i < a_k$，符合要求的 j 的个数是中间整块里值域在 $[a_i, a_k]$ 的数的个数，可以求出 $\leq a_k$ 的个数 v_k，减掉 $\leq a_i$ 的个数 v_i，故我们将两个块的元素从小到大归并起来后，把贡献拆开即可 $O(B)$ 计算。

[Ynoi1998] Marchen

- 散块-整块-整块 / 整块-整块-散块
  两种是对称的，只考虑第一种，对于每个 i 预处理 $ans_{i,b}$ 表示 i 和 $[bel_i+1,b]$ 组成的答案。
  固定 i，从小到大枚举 b，考虑 k 位于块 k 的数量，那么此时预处理 j 和 k 同块的方案数，这个很容易做到 $O(nB)$，对于 j 和 k 不同块的方案数，我们用 $a_j<a_k$ 的数量减掉 $a_j<a_i<a_k$ 的数量，减掉 $a_j<a_k<a_i$ 的数量，就是 $a_i<a_j<a_k$ 的数量。
  $a_j<a_k$ 的数量可以提前枚举 b 后扫一遍预处理，$a_j<a_k<a_i$ 用两次可持久化分块查二维数点即可。对于第三种，枚举 b，然后把所有值离散化到 $O(B)$ 种，从右到左扫一遍即可。

[Ynoi1998] Marchen

- 整块-整块-整块
  使用上面的 ans 数组容易求出来。

然后有个问题是开不下那个可持久化分块的数组，因此我们考虑把这一部分替换掉，可以发现除了散块内部的贡献，剩下的都是某个整块区间的查询，只需要开一个表查询即可。
对于散块内部的查询，用 i 左边同块内比 a_i 小的数的数量减掉比 a_i 小的数中 $< l$ 的即可，前者可以预处理，后者在排序后的数组上扫一遍即可。
最后复杂度是 $O((n+m)\sqrt{n})$，有亿点点卡常。

[UR28] 环环相扣

题目大意

给定一个元素两两不同的序列 a，m 次询问区间 $[l,r]$，你要选出 $l \le i,j,k \le r$ 且两两不同的三个下标 i,j,k，最大化 $a_i \bmod a_j + a_j \bmod a_k + a_k \bmod a_i$。

$1 \le n \le 2 \times 10^6, 1 \le m \le 8 \times 10^5, 1 \le a_i \le 10^{18}$。

[UR28] 环环相扣

首先讨论一下 a_i, a_j, a_k 的大小关系，可以发现，只有两种可能的形态：

- $a_i < a_j < a_k$，贡献是 $a_i + a_j + a_k \bmod a_i$。
- $a_i > a_j > a_k$，贡献是 $a_i \bmod a_j + a_j \bmod a_k + a_k$。

下面的若干性质证明会用到性质：若 $a_i > a_j$，则 $a_i \bmod a_j + a_j \le (a_i - a_j) + a_j = \min(a_i, 2a_j - 1)$

很容易猜到下面的结论：

引理 (Key Observation)

最优方案下，区间内的最大值和次大值一定被选。

[UR28] 环环相扣

证明.

不妨假设区间内的数字从小到大排列为 $w_1, w_2 \dots w_m$。
那么选择前三大的数就可以得到一组解为 $w_{m-2} + w_{m-1} + w_m \bmod w_{m-2}$，假设最优解为 (i, j, k)，那么在方案 $a_i + a_j + a_k \bmod a_i$ 中 $\leq a_k + a_j$，只有当 $k = m$ 时可能大于之前的解，此时最优的 j 时 $m - 1$；而 $a_i \bmod a_j + a_j \bmod a_k + a_k \leq a_i \leq \min(a_i, 2a_j - 1)$，如果 $i < m$ 或 $j < m - 1$，那么一定不优。 □

容易根据该引理得到一个 $O(nq)$ 的做法。

[UR28] 环环相扣

设区间内的最大值下标是 i，次大值下标是 j，设 $f([l,r],i)$ 表示从区间 $[l,r]$ 选择一个 k，满足 $k \neq i$，且 a_k 不是 $[l,i) \cup (i,r]$ 内的最大值，使得 $a_i \bmod a_k + a_k$ 最大。
那么两种答案可以分别写成
$f([l,r],j) + a_i \bmod a_j, f([l,r],i) + a_j$。
现在把问题转化为求 $f([l,r],i)$，我们把它拆成 $f([l,i],i)$ 和 $f([i,r],i)$ 两部分，注意这里可能会让原本的次大值不被考虑到，我们需要求一下然后更新答案。

[UR28] 环环相扣

两种是对称的，我们现在假设要求 $f([i,r],i)$，固定 i 扫 r 即可做到 $O(n^2+q)$。

注意根据查询的形式，这里的 i 一定是 $[i,r]$ 内的最大值或次大值，考虑 $a_i \bmod a_k + a_k \le a_i$，并且取等的条件是 $a_i - a_k < a_k$ 即 $a_k > \frac{a_i}{2}$。

因此固定 i，有用的 r 一定不超过第二个 $> \frac{a_i}{2}$ 的元素，再往后扫都不变了。

分析一下此时的总支配对数量，我们按照值域倍增分块，则对于每个值域块内的元素来说，最多向左/向右扫 2 个段，故总的扫描次数是 $O(n)$，加上外层的倍增分块就是 $O(n\log V)$。

复杂度 $O(n\log V + q\log n)$。

[UR28] 环环相扣

还不够好！
考虑固定扫 r 的过程中，现在扫到了某个元素 a_k，如果 $[r, k-1]$ 的元素里至少有两个元素 $\geq 2a_k$，那么 k 就是没用的，这是因为 $a_i \bmod a_k + a_k < 2a_k$ 所以选择 $\geq a_k$ 的元素一定是更优。
我们从大到小枚举 i，如果出现一个满足上述条件的 k，那么对于更小的 i 也是没用的，因此我们可以用链表维护没有被删除的数字，每次扫一遍没被删除的数字，如果扫到的 $a_k \leq \frac{i}{2}$，那么最多被扫两次就被删掉了；否则根据上一个性质，扫到第二个这种元素就停止了，那么很容易说明总的扫描次数是 $O(n)$ 的，我们直接顺便出来每个 $f([i, r], i)$ 的值发生变化的位置，询问的时候直接二分即可。
复杂度是 $O(n + q \log n)$

[UR28] 环环相扣

还不够好！
考虑固定扫 r 的过程中，现在扫到了某个元素 a_k，如果 $[r, k-1]$ 的元素里至少有两个元素 $\geq 2a_k$，那么 k 就是没用的，这是因为 $a_i \bmod a_k + a_k < 2a_k$ 所以选择 $\geq a_k$ 的元素一定是更优。
我们从大到小枚举 i，如果出现一个满足上述条件的 k，那么对于更小的 i 也是没用的，因此我们可以用链表维护没有被删除的数字，每次扫一遍没被删除的数字，如果扫到的 $a_k \leq \frac{i}{2}$，那么最多被扫两次就被删掉了；否则根据上一个性质，扫到第二个这种元素就停止了，那么很容易说明总的扫描次数是 $O(n)$ 的，我们直接顺便出来每个 $f([i, r], i)$ 的值发生变化的位置，询问的时候直接二分即可。
复杂度是 $O(n + q \log n)$

[CTS2022] 普罗霍洛夫卡

题目大意

给定序列 $a_1, \ldots, a_n$，共 m 次询问，每次询问给出 l, r，查询所有满足 $l \le L \le R \le r$ 的 (L, R) 的权值的按位异或和，二元组 (L, R) 的权值是 $|\{a_i \mid L \le i \le R\}|$。

$n, m \le 4 \times 10^5$。

[CTS2022] 普罗霍洛夫卡

题目大意

给定序列 $a_1, \ldots, a_n$，共 m 次询问，每次询问给出 l, r，查询所有满足 $l \le L \le R \le r$ 的 (L, R) 的权值的按位异或和，二元组 (L, R) 的权值是 $|\{a_i \mid L \le i \le R\}|$。

$n, m \le 4 \times 10^5$。

[CTS2022] 普罗霍洛夫卡

离线，对 r 扫描线，维护每个 $[l,r]$ 的权值，也就是颜色数。设 a_i 上一次出现的位置是 lst_{a_i}，那么就是对 $(lst_{a_i},i]$ 做一次区间加 1。同时要维护历史异或和 h_i，然后扫到一个询问的右端点时查询历史异或和的区间异或和。

将 w_i 写成 $(0\oplus 1)\oplus(1\oplus 2)\ldots(w_i-1)\oplus w_i$ 的形式，这样每次修改操作的贡献可以拆开了。

具体来说，当我们扫到 R 的时候对 i 这个位置做了 $+1$，那么如果在扫到 r 时查询，如果 r 和 R 奇偶性不同，那么可以认为这次操作产生了 $w_i\oplus(w_i+1)$ 的贡献，否则贡献为 0。

[CTS2022] 普罗霍洛夫卡

将奇偶分开，形式化地描述一下我们现在的操作：

维护一个序列 w，初始时全为 0，同时维护序列 h_0, h_1，每次让区间内的 w_i 加一，并让 $h_{o,i} = h_{o,i} \oplus w_i (\oplus w_i + 1)$，其中 o 是当前操作编号 $r \bmod 2$，查询区间异或和。

设 w_i 最低的 0 所在位是是 2^t，那么 $w_i \oplus (w_i + 1)$ 就是 $2^{t+1} - 1$。

[CTS2022] 普罗霍洛夫卡

分块，考虑在进行一个整块操作时，对于每个 t 计算会有多少数的最低的 0 是这一位。对于每个 t 预处理一个桶 $B_{t,2^t}$ 表示块内 $\mathrm{mod} 2^t \equiv 0 \sim 2^t - 1 (\mathrm{mod} 2^t)$ 的个数，然后维护块内被整体加的次数 tag，那么我们只需要查询 $B_{t+1}, (2^t - 1 - tag) \bmod 2^{t+1}$ 的个数即可，同时也可以通过这个来打标记。

但是这样桶需要开到 $O(n)$ 级别，这样就炸了，我们考虑对 t 根号分治一下，前 b 位我们按照上面的方法处理。对于一个 w_i，其在 b 位产生贡献的次数最多不超过 $O(n/2^b)$，这样桶就只需要处理 $O(\sqrt{n})$ 即可，大一的部分也开个桶每次暴力找到需要改的改一下即可。

复杂度 $O(n\sqrt{n}\log n)$

[CTS2022] 普罗霍洛夫卡

发现在于低位的处理不太行，而这部分有用的 tag 模意义下显然只有 2^b 种，直接预处理一个 val_i 表示 $tag \equiv i$ 时的贡献，这个直接每次加上一位去求贡献，总的状态数就是 $2^0 + 2^1 + \ldots 2^b = 2^{b+1}$ 种，故预处理也不带 log。

最后的复杂度是 $O(n\sqrt{n})$。

Thanks!