

[Ynoi2007] rdiq

range different inversions query

给定一个长为 n 的序列 a ， m 次查询区间 $[l, r]$ 本质不同逆序对数。

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

Solution

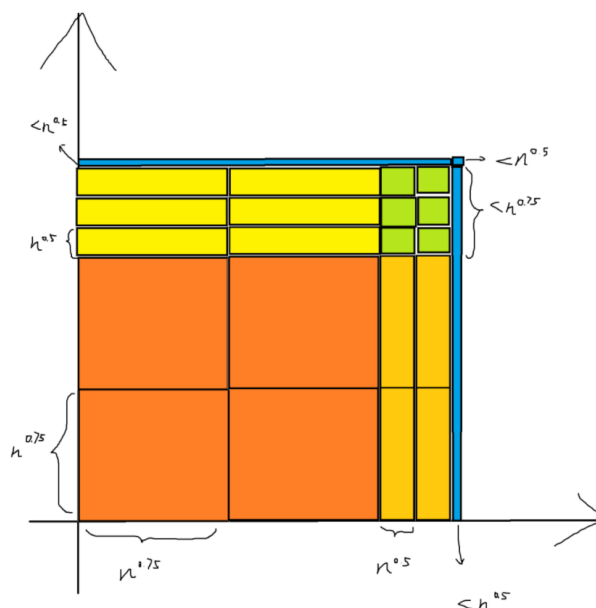
考虑二次离线莫队，记 pre_i, suf_i 表示 i 左右第一个等于 a_i 的位置。

以 r 的移动为例，从 $[l, r]$ 转移到 $[l, r+1]$ ，增加量为所有在 $[nxt_{r+1}, r+1]$ 区间内满足 $a_i > a_{r+1}$ 且 a_i 不在 $[l, nxt_{r+1}]$ 区间出现的个数，不妨记为 $f(l, r+1)$ 。相邻询问间的 r 移动是连续的，因此可表示成区间的形式，即 $f(l, [r+1, qr])$ 。

之后跑扫描线。为方便起见，将 a 序列翻转，则 $f(l, [x, y]) \rightarrow f(n-l+1, [n-y+1, n-x+1])$ 。

对于同个数保留最靠前的位置。修改操作为对 $x \in [1, i], y \in [1, a_i]$ 的位置全体 $+1$ ，查询操作为求 (i, a_i) 的值。

修改操作有 $\mathcal{O}(n)$ 次，而查询操作有 $\mathcal{O}(n\sqrt{m})$ 次。考虑设计一个 $\mathcal{O}(B)$ 修改， $\mathcal{O}(1)$ 查询的分块：



考虑将 $(1, 1)$ 到 (i, a_i) 的矩形劈成这样 5 类，记橙色块为 S_1 ，浅橙色块为 S_2 ，黄色块为 S_3 ，绿色块为 S_4 ，蓝色块先不管。

设橙色块长为 B^3 ，绿色块长为 B^2 ，则橙色块一共有 $\mathcal{O}((\frac{n}{B^3})^2)$ 块，浅橙色和黄色块一共有 $\mathcal{O}(\frac{B^3}{B^2} \cdot \frac{n}{B^3}) = \mathcal{O}(\frac{n}{B^2})$ 块，绿色块一共有 $\mathcal{O}((\frac{B^3}{B^2})^2) = \mathcal{O}(B^2)$ 块。

蓝色块不方便区间加，考虑它对所有落在蓝色区域的 $\mathcal{O}(B^2)$ 个询问的贡献，这是好求的。

时间复杂度 $\mathcal{O}(n\sqrt{m} + n((\frac{n}{B^3})^2 + \frac{n}{B^2} + B^2))$ ，空间复杂度 $\mathcal{O}((\frac{n}{B})^2)$ 。

取 $B = n^{0.25}$ ，得最优时间复杂度 $\mathcal{O}(n\sqrt{m})$ ，空间复杂度 $\mathcal{O}(n)$ 。

[Ynoi2007] rfplca

range father pair LCA

给定一棵 n 个节点的树, a_i 与 i 连边 ($i \geq 2$), 有两种操作:

- **1 l r x**: 对于 $l \leq i \leq r$, $a_i = \max(a_i - x, 1)$ 。
- **2 u v**: 查询当前 a 数组下, 树上 u, v 的 LCA。

本题强制在线。

$2 \leq n, q \leq 4 \times 10^5, a_i < i (i \geq 2), 1 \leq x \leq 4 \times 10^5$

时间限制 2500ms, 空间限制 64MB。

Solution

简单序列分块题, 时间复杂度 $\mathcal{O}(n\sqrt{n})$, 空间复杂度 $\mathcal{O}(n)$ 。

题解有空一定补!!!

[Ynoi2007] rgxsxrs

给定一个长为 n 的序列 a , 有两种操作:

- **1 l r x**: 区间 $[l, r]$ 中 $> x$ 的元素减去 x ;
- **2 l r**: 询问区间 $[l, r]$ 的和, 最小值, 最大值。

$1 \leq n, m \leq 5 \cdot 10^5, 1 \leq a_i, x \leq 10^9$, 强制在线。

Solution

哎, 我想鸽 (

https://www.bilibili.com/video/BV1qA411c7K7?spm_id_from=333.999.0.0

1h40min 左右的位置。

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

题解有空一定补!!!

[Ynoi2009] rprsvq

题目链接: [洛谷 P6108](#)。

Range Plus Range Subsequence Variance Query

给定一个长为 n 的序列 a , 初始全 0, 有两种操作:

- **1 l r a**: 区间 $[l, r]$ 整体加 a ;
- **2 l r**: 求区间 $[l, r]$ 所有非空子序列的方差之和, 答案对 998244353 取模。

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

时间限制 1500ms, 空间限制 512MB。

Solution

对于一个数列 x , 其方差为:

$$s^2 = \frac{1}{n} \left(\sum_{i=1}^n (x_i - \bar{x})^2 \right) = \frac{n-1}{n^2} \sum_{i=1}^n x_i^2 - \frac{2}{n^2} \sum_{1 \leq i < j \leq n} x_i x_j$$

因此：

$$\begin{aligned} ans &= \sum_{S \subseteq \{1, \dots, n\}, |S| \neq 0} \left(\frac{|S|-1}{|S|^2} \sum_{u \in S} x_u^2 - \frac{2}{|S|^2} \sum_{u, v \in S, u < v} x_u x_v \right) \\ &= \sum_{t=1}^n \binom{n-1}{t-1} \frac{t-1}{t^2} \sum_{i=1}^n x_i^2 - \sum_{t=2}^n \binom{n-2}{t-2} \frac{2}{t^2} \sum_{1 \leq i < j \leq n} x_i x_j \\ &= \sum_{t=2}^n \binom{n-2}{t-2} \frac{n-1}{t^2} \sum_{i=1}^n x_i^2 - \sum_{t=2}^n \binom{n-2}{t-2} \frac{2}{t^2} \sum_{1 \leq i < j \leq n} x_i x_j \\ &= \sum_{t=2}^n \frac{\binom{n-2}{t-2}}{t^2} \left((n-1) \sum_{i=1}^n x_i^2 - 2 \sum_{1 \leq i < j \leq n} x_i x_j \right) \end{aligned}$$

注意到 $\sum_{t=1}^n \frac{\binom{n-2}{t-2}}{t^2} = \frac{1}{n(n-1)} \sum_{t=1}^n \frac{\binom{n}{t} (t-1)}{t} = \frac{1}{n(n-1)} (2^n - 1 - \sum_{t=1}^n \frac{\binom{n}{t}}{t})$

$$\sum_{t=1}^n \frac{\binom{n}{t}}{t} = n \sum_{t=1}^n \frac{\binom{n-1}{t-1}}{t^2} = n \sum_{t=0}^{n-1} \frac{\binom{n-1}{t}}{(t+1)^2}$$

令 $f(n) = \sum_{t=0}^n \frac{\binom{n}{t}}{t+1}$, $g(n) = \sum_{t=0}^n \frac{\binom{n}{t}}{(t+1)^2}$, 则：

$$\begin{aligned} f(n) &= \sum_{t=0}^n \binom{n}{t} \int_0^1 x^t dx = \int_0^1 (1+x)^n dx = \frac{2^{n+1} - 1}{n+1} \\ g(n) &= \frac{1}{n+1} \sum_{i=0}^n \binom{n+1}{i+1} \frac{1}{i+1} \\ &= \frac{1}{n+1} \int_0^1 \sum_{i=0}^n \binom{n+1}{i+1} x^i dx \\ &= \frac{1}{n+1} \int_0^1 \frac{1}{x} ((1+x)^{n+1} - 1) dx \\ &= \frac{1}{n+1} \sum_{i=0}^n \int_0^1 (1+x)^i dx \\ &= \frac{1}{n+1} \sum_{i=0}^n f(i) \end{aligned}$$

线性预处理 f, g , 时间复杂度 $O(q \log n)$, 空间复杂度 $O(n)$ 。

[Ynoi2009] rla1rmdq

range level ancestor 1 range minimal depth query

给定一棵 n 个节点的树，带边权，根为 rt 。

定义 $fa(x)$ 表示 x 的父亲（特别地， $fa(rt) = rt$ ）。

定义 $dep(x)$ 表示 x 到 rt 简单路径上的边权和。

给定一个长为 n 的序列 a ，有两种操作：

- 1 l r ：对于 $l \leq i \leq r$ ， $a_i = fa(a_i)$ ；
- 2 l r ：对于 $l \leq i \leq r$ ，查询 $\min\{dep(a_i)\}$ 。

$1 \leq n, m \leq 2 \times 10^5, 1 \leq a_i \leq n$, 边权 $\in [0, 10^9]$ 。

时间限制 3000ms , 空间限制 64MB 。

Solution

序列分块简单题。

时间复杂度 $O(n\sqrt{n})$, 空间复杂度 $O(\frac{n\sqrt{n}}{\omega})$ 。

然鹅我写了 $O(n\sqrt{n\log n})$ /you!

题解有空一定补!!!

[Ynoi2011] 遥远的过去

题目链接: [洛谷 P5310 \(洛谷月赛\)](#)。

有一种字符集超大的 Z 语言, 这种语言的特点是:

- 字符集超大, 有 2^{31} 种字符;
- 每个单词由一系列**两两不同**的字符组成;
- 字符既能比较相同和不同, 也能比较大小;
- 两个单词只要离散化后相同, 则本质相同。

给定两个字符串 A, B , 求 B 作为子串在 A 中被匹配的次数。

有 q 次修改, 每次修改 B 中的一个字符, 计算匹配次数。

$n, m, q \leq 10^5$, 保证字符串时刻合法。

时间限制 1000ms, 空间限制 512MB。

Solution

对于 A 的每连续 m 个数字, 我们可以将其离散化后记为 $\sum_{i=1}^m a_i base^{m-i}$ 。

从 $[i, i+m-1]$ 到 $[i+1, i+m]$, 相当于删去下标 i , 加上下标 $i+m$, 这可以用 Fhq_Treap 快速维护。

同理维护 B 即可, 每次修改即先 `erase` 一个数, 再 `insert` 一个数。

时间复杂度 $O(n \log n)$ 。

空间复杂度 $O(n)$ 。

[Ynoi2012] NOIP2016 人生巅峰

题目链接: [洛谷 P5527](#) 。

给定一个长为 n 的序列 a , 数的值域 $[0, v)$, 有 2 种操作:

- **1 1 r** : 询问 $[l, r]$ 内是否可以选出两个**非空下标集合** X, Y , 满足:
 - X 和 Y 没有交集;
 - $\sum_{i \in X} (a_i + 1) = \sum_{i \in Y} (a_i + 1)$ 。
- **2 1 r** : 对于所有 $l \leq i \leq r$, $a_i \leftarrow a_i^3 \pmod{v}$ 。

$1 \leq n, m \leq 10^5, 1 \leq v \leq 1000$ 。

时间限制 500ms, 空间限制 128MB。

Solution

区间内的子集数为 2^{r-l+1} , 每个区间的 sum 不超过 $1000(r-l+1)$ 。

若 $2^{r-l+1} > 1000(r-l+1)$, 即 $r-l+1 > 13$, 由抽屉原理知必有两集合 sum 相等, 且不相交。

否则 $r-l+1 \leq 13$, 考虑 $dp[i][j] = 0/1$ 表示前 i 个数能否拼成 j 。

如果在某个 i 满足 $dp[i][j] = 1$ 且 $dp[i][j - a[i] - 1] = 1$ 则符合条件。这一过程可以用 *bitset* 优化。

区间修改直接用树状数组维护一下幂次即可, 注意 $v \leq 1000$, 可以很方便预处理 $f[i][j]$ 表示 $i^{3^j} \bmod v$ 的值。

时间复杂度 $\mathcal{O}(m(\log n + \frac{16900}{\omega}))$, 跑不满。

空间复杂度 $\mathcal{O}(n)$ 。

[Ynoi2013] 无力回天 NOI2017

题目链接: [洛谷 P5607](#)。

给定一个长为 n 的序列 a , 实现两种操作:

- 区间 $[l, r]$ 内的所有数都 xor 上 v ;
- 查询区间 $[l, r]$ 内选任意个数 (包括 0 个) 的最大 xor 和是多少。

$1 \leq n, m \leq 5 \times 10^4, 0 \leq a_i, v \leq 10^9$ 。

时间限制 4000ms, 空间限制 128MB。

Solution

区间修改线性基并不好维护, 因此考虑差分。

定义 $b_i = a_i \text{ xor } a_{i-1}$, 则 a_l, a_{l+1}, \dots, a_r 的线性基等于 $a_l, b_{l+1}, b_{l+2}, \dots, b_r$ 的线性基 (因为 $a_i \text{ xor } a_j \text{ xor } a_j = a_i$)。

对于一次修改, 相当于在 b_l 和 b_{r+1} 上分别插入一个 v , 查询即得到 a_l 的值以及 b_{l+1}, \dots, b_r 的线性基, 插入 a_l 后求一次 *getans*(x) 即可。

时间复杂度 $\mathcal{O}(q \log n \log^2(\max\{a_i\}))$, 跑不满。

空间复杂度 $\mathcal{O}(n \log(\max\{a_i\}))$ 。

[code](#)

[Ynoi2013] 文化课

题目链接: [洛谷 P5608](#)。

鸽了。

[Ynoi2013] 对数据结构的爱

题目链接: [洛谷 P5609](#), [CF1172F Nauuo and Bug](#)。

定义函数

$$\text{add}(x, y) = \begin{cases} x + y - p & \text{if } x + y \geq p \\ x + y & \text{otherwise} \end{cases}$$

函数 $f(l, r)$ 的伪代码:

```
1 result <- 0
2 for i <- 1 to r do
3   result <- add(result, a[i])
4 return result
```

给定一个长为 n 的序列 a 以及模数 p , 每次查询区间 $f(l, r)$ 的值。

$1 \leq n \leq 10^6, 1 \leq m \leq 2 \times 10^5, 1 \leq p \leq 10^9, -10^9 \leq a_i \leq 10^9$ 。

时间限制 2000ms, 空间限制 512MB。

Solution

区间 $[l, r]$ 的答案一定可以表示为 $\text{sum} - xp$, x 表示减去 p 的次数, 且显然有 $0 \leq r - l + 1 \leq p$ 。

考虑线段树, 但发现初始值可能会对 x 产生影响, 所以我们对于每个节点都记录 $c[i]$ 表示区间 $x = i$ 的最小初值。

显然, 区间减去 x 的初始值范围为 $[c[x], c[x + 1] - 1]$ 。

分析节点 u 的合并, 记两儿子为 ls 和 rs , 则有 $c[x + y] = \min\{\max(c[x], c[y] - (\text{sum}[ls] - xp))\}$ 。

考虑什么情况下的 x, y 是合法的: 如果 ls 最大初始值到都到达不了 rs 的最小初始值, 那么就不合法, 即 $(c[ls][x + 1] - 1) + (\text{sum}[ls] - xp) < c[rs][y]$, 否则合法。

目前合并是 $O(\text{len}^2)$ 的, 考虑如何优化。

首先, 对于合法的 (x, y) 以及 $(x + 1, y - 1)$, (x, y) **不劣于** $(x + 1, y - 1)$, 因此随着 x 的递增, y 单调不减。

其次, 因为要比较不合法的情况, 所以还需满足对于任意 x , 有 $c[ls][x + 1] - c[ls][x] \geq p$, 可感性理解, 因为要多减一次就需要整体抬 p 。

这样, 就可以通过单调队列优化到 $O(\text{len})$ 。

建树过程时间复杂度 $O(n \log n)$, 空间复杂度 $O(n \log n)$;

查询的话, 直接 *query* 了, 查到对应区间二分一下, 找到对应**减去 p 次数**的区间即可, 这里时间复杂度 $O(q \log^2 n)$ 。

时间复杂度 $O(n \log n + q \log^2 n)$ 。

空间复杂度 $O(n \log n)$ 。

[code](#)

[Ynoi2013] 大学

题目链接: [洛谷 P5610](#)。

给定一个长为 n 的序列 a , 支持两种操作:

- `1 l r x`: 将区间 $[l, r]$ 中所有 x 的倍数除以 x ;

- `2 1 r`: 查询区间 $[l, r]$ 的和。

本题强制在线。

$$1 \leq n, m \leq 10^5, 0 \leq a_i \leq 5 \times 10^5。$$

Solution

问题瓶颈在于如何快速找到该除的位置，因为每个位置最多被除 \log 次。

我们可以预处理，然后对于每个因子开个 dsu，用 dsu 实现快速寻找下一个位置即可。

时间复杂度 $\mathcal{O}(n(\sum d(a_i))\alpha(n))$ 。

空间复杂度 $\mathcal{O}(n \sum d(a_i))$ 。

[Ynoi2013] D2T2

鸽了。

[Ynoi2013] Ynoi

题目链接：[洛谷 P5612](#)。

给定一个长为 n 的序列 a ，支持三种操作：

- `1 1 r x`: 将 $[l, r]$ 中的数异或 x ；
- `2 1 r`: 将 $[l, r]$ 排序；
- `3 1 r`: 查询区间 $[l, r]$ 异或和。

$$1 \leq n, m \leq 10^5, 0 \leq a_i, x < 10^8。$$

时间限制 1500ms，空间限制 32MB。

Solution

区间排序和区间异或会将原序列分成一些连续的段，其中一次排序操作会把区间内改为同一个新的连续段，一次异或操作会将区间端点处的段分裂为两段。显然在 m 次操作后，至多产生 $2m$ 个新的连续段。

对同一个连续段，每次排序会覆盖掉以前排序的效果，而相邻几次异或操作的效果可以叠加，所以一个连续段的信息可以用一个已排好序的序列异或上一个数表示。

为了处理连续段的变化，可以用平衡树维护序列中的连续段，通过打标记来处理操作。对每个连续段内部，用二进制 trie 维护序列。这里的 trie 较为特殊，为了节省空间，只维护叶节点构成的虚树，这样对一个长度为 k 的连续段，只需用一棵 $2k-1$ 个点的 trie 表示。在 trie 上维护异或标记和排序标记（表示对这棵子树先排序后异或），其中异或标记的效果是对整棵子树内的值异或上一个数，而排序标记的效果是将整棵子树内由于异或而左子树大于右子树的点交换左右子树。打上排序标记前要先下传异或标记。在排序时会合并多个连续段，这时将每棵 trie 打上排序标记，利用 trie 的合并操作得到合并后的连续段。分裂连续段时可以用 trie 的分裂操作实现。

[Ynoi2014] 在太阳西斜的这个世界里

题目链接：[洛谷 P5062](#)。

维护一棵初始为空的红黑树，进行两种操作：

- 在红黑树上插入一个整数 x （对于重复的整数，插入时间晚的小）；
- 给定 x ，询问将 1 到 x 的整数分别（独立）插入到红黑树，造成的旋转次数总和。

$$1 \leq n \leq 2 \times 10^5, 1 \leq x \leq 10^8。$$

时间限制 1000ms，空间限制 128MB。

还没学过红黑树，怎么办（

[Ynoi2014] 置身天上之森

题目链接：[洛谷 P5063](#)。

你有一个根节点区间是 $[1, n]$ 的线段树，支持两种操作：

- `1 l r a`：对每个 $x \in [l, r]$ ，将 $[x, x]$ 对应的叶子权值 $+a$ ，非叶节点的权值相应变化；
- `2 l r a`：询问有多少个线段树节点被 $[l, r]$ 包含，且权值 $\leq a$ 。

$$1 \leq n, m \leq 10^5, -10^5 \leq a \leq 10^5。$$

时间限制 3000ms，空间限制 128MB。

Solution

首先，假设这两个操作放在序列上，只能做到 $O(m\sqrt{n \log n})$ （取块长 $B = \sqrt{n \log n}$ ），所以我们尽量往这方面凑。

容易发现，线段树中本质不同的线段长度只有 $2 \log n$ 种，并且任意两根线段要么不交要么包含，所以我们将每种长度的线段提出来，分别执行上述做法即可。这样复杂度是 $O(q\sqrt{n \log n \log n})$ 。

用分散层叠算法优化上述流程，可以做到时间复杂度 $O(q\sqrt{n \log n})$ 。

时间复杂度 $O(q\sqrt{n \log n})$ 。

空间复杂度 $O(n)$ 。

[Ynoi2014] 等这场战争结束之后

题目链接：[洛谷 P5064](#)。

给你一张图，每个点有点权 a_i ，最开始没有边。有三种操作：

- 添加一条 x 与 y 之间的双向边；
- 回到第 x 次操作后的状态；
- 查询 x 所在连通块能到的点中，点权第 y 小值。

$$1 \leq n, m \leq 10^5, 0 \leq a_i \leq 10^9。$$

时间限制 500ms，空间限制 20MB。

Solution

注：本做法十分奇怪。

考虑根据 2 操作建出操作树，然后在树上用可撤销 dsu 维护连通性，同时对于每个连通块，我们值域分块，记 `cnt[x]` 表示第 x 块中数的个数。

这样时间复杂度 $O((n+m)\sqrt{n})$ ，空间复杂度 $O(n\sqrt{n})$ 。

事实上，你把块长调到 5000，那么 `cnt` 数组可以开 `short` 类型，空间就卡进去了。。。

正经的做法是树分块，能把空间复杂度优化至 $O(n)$ 。

[Ynoi2014] 不归之人与望眼欲穿的人们

题目链接: [洛谷 P5065](#)。

给定一个长为 n 的序列 a , 有两种操作:

- **1 p x**: 将 a_p 修改成 x ;
- **2 x**: 查询最短的区间 $[l, r]$, 满足 $\bigvee_{i=l}^r a_i \geq x$, 无解输出 **-1**。

$1 \leq n, m \leq 5 \times 10^4, 0 \leq a_i, k \leq 2^{30}$ 。

时间限制 1000ms, 空间限制 128MB。

Solution

记 $w = \log a_i$, 固定左端点 l , 对于 $r \in [l, n]$ 只有 $O(w)$ 个本质不同的 or 值。

用 $dp[i][j]$ 表示左端点为 i , 最小的满足第 j 位是 1 的右端点, $rdp[i][j]$ 表示右端点为 i , 最小的左端点。

令 $f(i) = \{(j, dp[i][j])\}$, 二元组按 $dp[i][j]$ 从小到大排序, 用归并实现可 $O(nw)$ 求。

直接暴力做是 $O(mnw)$ 的, 考虑序列分块, 记 S 表示块大小。

对于第 i 块, 记左端点为 L_i , 右端点为 R_i , $pre(i) = \{(j, dp[L_i][j])\}$, $suf(i) = \{(j, rdp[R_i][j])\}$ 。

注意这里的 dp, rdp 都是相对于块内元素而言的。

维护 $sumor_i$ 表示区间 or 值, $mx(i)$ 表示长度为 i 的区间的最大 or 值。

对于修改操作, 暴力重构该块。 dp, rdp 同上, mx 可以在维护 dp 时顺便计算, 复杂度 $O(wS)$ 。

对于查询操作:

- 若左右端点位于同一块, 则在 mx 暴力二分, 复杂度 $O(\frac{n}{S} \log S)$ 。
- 否则, 假设区间穿越了第 i 块, 则可以通过第 $i-1$ 块的 suf 和第 $i \sim \frac{n}{S}$ 块的 pre 计算。后缀维护 pre 即可, 然后 two pointers 做到单次 $O(w)$, 总复杂度 $O(\frac{n}{S} w)$ 。

时间复杂度 $O(m(wS + \frac{n}{S} \log S + \frac{n}{S} w))$, 取 $S = \sqrt{n}$ 得最优复杂度 $O(m\sqrt{n} \log a)$ 。

空间复杂度 $O(n \log a)$ 。

[Ynoi2014] 人人本着正义之名

题目链接: [洛谷 P5066](#)。

给定一个长为 n 的 01 序列 a , 实现以下 7 种操作:

- **1 l r**: 区间 $[l, r]$ 赋值为 0;
- **2 l r**: 区间 $[l, r]$ 赋值为 1;
- **3 l r**: 将 $[l, r-1]$ 内所有 a_i 变成 a_i 或 a_{i+1} (同时进行);
- **4 l r**: 将 $[l+1, r]$ 内所有 a_i 变成 a_i 或 a_{i-1} (同时进行);
- **5 l r**: 将 $[l, r-1]$ 内所有 a_i 变成 a_i 与 a_{i+1} (同时进行);
- **6 l r**: 将 $[l+1, r]$ 内所有 a_i 变成 a_i 与 a_{i-1} (同时进行);
- **7 l r**: 查询区间 $[l, r]$ 的和。

本题强制在线。

$$1 \leq n, m \leq 3 \times 10^6, 0 \leq a_i \leq 1.$$

时间限制 8000ms, 空间限制 512MB。

Solution

我们将原序列看做若干个值连续段, 以 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1 为例, 序列被划分为 $[0, 0], [1, 1, 1], [0, 0, 0], [1], [0], [1, 1]$ 。

开一棵平衡树, 每个节点代表一个值连续段。

对于 1/2 操作, 我们类似珂朵莉树做就好;

对于 3/4/5/6 操作, 它们本质相同, 我们以 3 操作为例:

序列变成 $[0], [1, 1, 1, 1], [0, 0], [1, 1], [], [1, 1, 1]$ 。

每个 0 连续段的右端点 -1 , 每个 1 连续段的左端点 -1 。这个都可以通过打懒标记做到。注意到, 我们可能把一个连续段搞成空的, 所以在操作结束以后, 要把空区间清掉垃圾。

总体来说就是这样, 细节有亿点点多。。各显神通吧。

时间复杂度 $\mathcal{O}(q \log n)$ 。

空间复杂度 $\mathcal{O}(n)$ 。

[Ynoi2014] 长存不灭的过去、逐渐消逝的未来

题目链接: [洛谷 P5067](#)。

鸽了。懂的都懂。

[Ynoi2015] 我回来了

题目链接: [洛谷 P5068](#)。

给定 n , 支持以下两种操作:

- 加入一个数 $h \in [1, n]$
- 查询 $[L, R]$, 计算 $\sum_{x=L}^R f(x)$, 其中 $f(x)$ 表示最小的满足不存在 $h \in [(f(x) - 1) \times x + 1, f(x) \times x]$ 的值。

$$1 \leq n \leq 10^5.$$

时间限制 1000ms, 空间限制 256MB。

Solution

容易发现, 区间总个数是 $\frac{n}{1} + \frac{n}{2} + \dots + \frac{n}{n} = \mathcal{O}(n \log n)$ 的, 所以瓶颈在于快速找到这些需要移动的区域。

我们可以通过线段树打个 `vector` 实现 (这个 `vector` 相当于一个“闹钟”, 在触发的时候我们 check 一下区间), 同时用一个树状数组维护答案即可。

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

空间复杂度 $\mathcal{O}(n \log^2 n)$, 但常数小的可怜 (

[Ynoi2015] 纵使日薄西山

题目链接: [洛谷 P5069](#)。

给定一个长为 n 的序列 a , 我们重复执行以下操作:

- 选定序列最大值 (如有多个, 选下标最小的), 将 a_{i-1}, a_i, a_{i+1} 减 1 (减到 < 0 仍为 0)

你需要计算操作次数, 使得序列变成全 0。

现在有 q 次修改, 每次将 a_x 改为 y , 求答案。

$$1 \leq n, q \leq 10^5, 1 \leq x_i \leq n, 1 \leq a_i, y_i \leq 10^9。$$

时间限制 1000ms, 空间限制 128MB。

Solution

假设这次操作了 x , 那么容易发现 x 位置不能受 $x-1, x+1$ 的影响 (因为 a_x 始终比 a_{x-1}, a_{x+1} 大), 所以 x 必定被自己减了 a_x 次。据此, 我们可以从大到小依次枚举数, 做到单次 $O(n)$ 。

让我们换一种视角去考虑它。假如序列是单调的, 那么一定取的是最大值、次大值、次次大值、次次次大值.....也就是奇偶性相同的位置的和。所以, 我们将序列划分成若干段单调区间, 对于每个单调区间是容易维护的, 注意相邻两个单调区间的交点 (即原序列的极值点) 可能会算重, 我们容易快速计算它们。奇偶性位置和, 可以开两个树状数组记录。

对于将 $a_x = y$, 它可能会导致 $x-1, x, x+1$ 三个位置的“极值点”性, 这些都是可以 $O(\log n)$ 快速维护的。

时间复杂度 $O(q \log n)$ 。

空间复杂度 $O(n)$ 。

[Ynoi2015] 即便看不到未来

题目链接: [洛谷 P5070](#)。

给定一个长为 n 的序列 a , 每次询问一个区间 $[l, r]$, 求长度为 $1, 2, \dots, 10$ 的极长值域连续段个数 (答案模 10), 定义如下:

- 记区间的数排序并去重后为 b_1, b_2, \dots, b_m , 则 $[x, y]$ 为极长连续段当前仅当 $b_{x-1} + 1 \neq b_x, b_y + 1 \neq b_{y+1}$ 且 $\forall i \in [x, y), b_i + 1 = b_{i+1}$ 。

$$1 \leq n, m, a_i \leq 10^6。$$

时间限制 3000ms, 空间限制 512MB。

Solution

简单题, 将询问离线, 树状数组维护, 每次暴力更新极长连续段即可。

时间复杂度 $O(kn \log n)$, 其中 $k = 10$ 。

空间复杂度 $O(n + q)$ 。

[Ynoi2015] 此时此刻的光辉

题目链接: [洛谷 P5071](#)。

给定一个长为 n 的序列 a , m 次询问, 每次查询区间 $d(\prod_{i=l}^r a_i) \bmod 19260817$, $d(x)$ 表示 x 的约数个数。

$1 \leq n, m \leq 10^5, 1 \leq a_i \leq 10^9$ 。

时间限制 2500ms, 空间限制 512MB。

Solution

对于一个 $x \leq 10^9$, 它至多有两个 $> x^{1/3}$ 的质因子, 所以预处理 ≤ 1000 的质数的指数前缀和。

对于 > 1000 的质数, 每个 a_i 最多有两个, 因此莫队优化即可。

实测 ≤ 1000 的质数有 168 个, 所以可以通过。

时间复杂度 $\mathcal{O}(168n + m(168 + \sqrt{n}))$ 。

空间复杂度 $\mathcal{O}(168n)$ 。

[code](#)

[Ynoi2015] 盼君勿忘

题目链接: [洛谷 P5072](#)。

给定一个长为 n 的序列 a , q 次查询。

每次给定 l, r, p , 求区间 $[l, r]$ 的所有子序列去重后的和, 对 p 取模。

$1 \leq n, m, a_i \leq 10^5, 1 \leq p \leq 10^9, 1 \leq l \leq r \leq n$ 。

时间限制 3000ms , 空间限制 512MB。

Solution

考虑莫队, 权值 x 的贡献显然是 $(2^{r-l+1} - 2^{\text{非}x\text{个数}}) \cdot x$ 。

区间内 `非x个数` 只有 $\sqrt{2n}$ 种取值, 所以对于每组询问用光速幂预处理 2^k , 然后暴力计算即可。

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

[Ynoi2015] 世上最幸福的女孩

题目链接: [洛谷 P5073](#)。

给定一个长为 n 的序列 a , 实现两种操作:

- `1 x`: 全局加 x ;
- `2 l r`: 查询区间 $[l, r]$ 最大子段和, 可以不选。

$1 \leq n \leq 3 \times 10^5, 1 \leq m \leq 6 \times 10^5, |a_i| \leq 2 \cdot 10^9, |x| \leq 5 \cdot 10^7$ 。

时间限制 1000ms, 空间限制 128MB。

Solution

对于不带修改的情况, 我们可以线段树维护四个标记 `pre, suf, ans, sum` 表示前缀最大值、后缀最大值、区间最大子段和、区间和。

加上修改操作，以 `pre` 为例，我们将所有点按照 (i, pre_i) 排布在坐标系上，有贡献的一定在上凸壳上，因此我们可以构建凸包去维护信息。

考虑如何合并。

对于 `pre` 而言：`pre[u] = max(pre[lis], sum[u] + pre[rs])`，那么我们将 `pre[rs]` 整体加上 `sum[u]`，然后对应位置取 `max` 即可。`suf` 同理。

对于 `ans` 而言：`ans[u] = max(ans[lis], ans[rs], suf[lis] + pre[rs])`，前两个是类似的，第三个是将两个上凸壳合并，直接闵可夫斯基和即可。

因此，构建整个线段树的复杂度是 $T(n) = 2T(\frac{n}{2}) + O(n)$ ，即 $O(n \log n)$ 的。

对于一组询问，我们可以对应到线段树的 $\log n$ 的节点上，在上凸壳二分找到最优决策位置。因此整体复杂度 $O(q \log^2 n)$ 。

由于可以离线，我们考虑将所有询问按照偏移量（即之前 1 操作的 x 之和）排序，然后线段树每个节点开一个指针，可以发现指针是单调右移的，所以时间复杂度均摊 $O(n \log n)$ 。

但以上做法空间都是 $O(n \log n)$ 的（线段树里的每个凸包都开了个 `vector`）。事实上我们可以隐式建线段树，即递归建，然后 `dfs` 函数每次返回 `pre, suf, ans, sum`，因为每时每刻都能保证每个位置只占用了 $O(1)$ 空间，并且当前 `dfs` 结束后，这些空间全被释放了，所以空间复杂度降至 $O(n)$ 。

时间复杂度 $O(n \log n)$ ，空间复杂度 $O(n)$ 。

[Ynoi2016] 掉进兔子洞

题目链接：[洛谷 P4688](#)，[LOJ 6199](#)，[BZOJ 4939](#)。

给定一个长为 n 的序列 a ，每次询问三个区间 $[l1, r1, l2, r2, l3, r3]$ ，把三个区间中同时出现的数一个一个删掉，求最后三个区间剩下的数的个数和。

$$1 \leq n, q \leq 10^5, 1 \leq a_i \leq 10^9。$$

时间限制 3000ms，空间限制 512MB。

Solution

考虑莫队，将所有数按照 (a_i, i) 二元组离散化。

对于一组询问，用 `bitset` 记录哪些数在区间内，三个区间取并即可。

注意到空间被卡了，于是每 $\frac{q}{3}$ 个询问一捆整体跑即可，常数 $\frac{1}{3}$ 。

时间复杂度 $O(\frac{nq}{w} + n\sqrt{q})$ ，空间复杂度 $O(\frac{nq}{w})$ ，跑不满。

[Ynoi2016] 这是我的发明

题目链接：[洛谷 P4689](#)，[LOJ 6200](#)，[BZOJ 4940](#)。

给定一棵 n 个节点的树，初始根为 1，有两种操作：

- `1 x`：将树根换为 x ；
- `2 x y`：从 x 子树中选一个点， y 子树中选一个点，求点权相等的方案数。

$$1 \leq n \leq 10^5, 1 \leq m \leq 5 \times 10^5, 1 \leq a_i \leq 10^9。$$

Solution

简单题，抠出 dfn 序列，直接莫队即可。

时间复杂度 $\mathcal{O}(n\sqrt{m})$ ，空间复杂度 $\mathcal{O}(n)$ 。

[Ynoi2016] 镜中的昆虫

题目链接：[洛谷 P4690](#)，[LOJ 6201](#)，[BZOJ 4941](#)。

给定一个长为 n 的序列 a ，有两种操作：

- `1 l r x`：将区间 $[l, r]$ 的值赋值为 x ；
- `2 l r`：询问区间 $[l, r]$ 出现多少种不同的数。

$1 \leq n, q \leq 10^5, 1 \leq a_i \leq 10^9$ 。

时间限制 1500ms，空间限制 64MB。

Solution

先考虑如何快速计算操作 2，令 pre_i 表示最大的 $j < i$ 满足 $a_j = a_i$ ，那么答案即为 $\sum_{i=l}^r [pre_i < l]$ ，所以我们需要动态维护 pre 数组，这是一个二维偏序状物。

看见区间赋值，不难想到用珂朵莉树维护。具体地，我们对每个颜色开一个 set，同时全局维护一个 set，这样每次区间赋值，可以在全局 set 上修改，并快速对应到 pre 并修改。

加上时间轴这一维，整个问题是一个三维偏序问题，可以树套树解决，但是空间有些吃力。我们改成离线跑 cdq 分治即可。

时间复杂度 $\mathcal{O}((n+q)\log^2 n)$ 。

空间复杂度 $\mathcal{O}(n+q)$ 。

[Ynoi2016] 炸脖龙 I

题目链接：[洛谷 P4118](#)，[BZOJ 5394](#)。

给定一个长为 n 的序列 a ，有两种操作：

- `1 l r x`：区间 $[l, r]$ 加 x ；
- `2 l r p`：查询 $a[l]^{a[l+1]^{a[l+2] \cdots a[r]}} \bmod p$ 。

$1 \leq n, m \leq 500000, a[i], x \in [1, 2 \times 10^9], p \leq 2 \times 10^7$ 。

时间限制 2500ms，空间限制 512MB。

Solution

指数依次模 $\varphi(p)$ ，最多 $\mathcal{O}(\log)$ 次就出现 $p^{(k)} = 1$ 。

因此暴力查询即可，时间复杂度 $\mathcal{O}(p + m \log n \log p)$ ，跑不满。

空间复杂度 $\mathcal{O}(n)$ 。

[Ynoi2016] 谁的梦

题目链接：[洛谷 P4692](#)，[BZOJ 5395](#)。

给定 n 个序列，第 i 个序列长度为 len_i 。

定义一个序列的权值为不同数字个数。我们在每个序列里选择一个连续非空子串，拼接起来。求所有选法得到的序列的权值和。

有 q 次修改，每次给定 x, y, z ，将第 x 个序列的第 y 个元素改成 z 。

$1 \leq n, m, \sum len_i \leq 10^5$ 。

时间限制 1500ms，空间限制 128MB。

Solution

正难则反，我们考虑每种数字不出现的方案，显然是 \prod 每个序列选择不含 x 的子串数，而这个容易通过 x 的下标集合快速计算。

因此，我们对每个数开个 set，记录下标。每次修改即修改某一项，是平凡的。

注意在实现的时候，可能会出现乘以 0 再除以 0 的情形，所以我们需要把贡献表示为 $0^a \times b$ 的形式，这样方便处理 0 的情形。

记 $len = \sum len_i$ ，则时间复杂度 $\mathcal{O}(len \log len)$ ，空间复杂度 $\mathcal{O}(len)$ 。

[Ynoi2016] 炸脖龙 II

题目链接：[洛谷 P4693](#)，[BZOJ 5396](#)。

鸽了。

[Ynoi2017] 由乃的玉米田

题目链接：[洛谷 P5355](#)，[BZOJ 4810](#)。

给定一个长为 n 的序列 a ，有 4 种询问：

- 查询区间 $[l, r]$ 是否可以选出两个数，满足它们的和/差/乘积/整除（无余数）为 c 。

$1 \leq n, q \leq 10^5, 1 \leq c \leq 10^5$ 。

时间限制 1000ms，空间限制 128MB。

Solution

先忽略除法，我们考虑莫队：开一个 bitset s ，第 i 位为 1 表示存在数 i 。

对于 $x - y = c$ 而言，即满足 s 与 $s \gg x$ 有交；

对于 $x + y = c$ 而言， $n - x = y + n - c$ ，因此再开一个 bitset s' ，第 i 位为 1 表示存在数 $n - i$ ，满足 s 与 $s' \gg n - c$ 有交；

对于 $x \times y = c$ 而言，直接暴力枚举 c 的因数即可。

现在加上除法，我们采取根号分治：

- 若 $c > \sqrt{n}$ ，我们类似乘法去暴力枚举；
- 若 $c \leq \sqrt{n}$ ，枚举每种 c ，然后扫一遍数组即可得到 $maxl[r]$ 表示右端点为 r 时左端点最右能多少。

时间复杂度 $\mathcal{O}(\frac{nq}{\omega} + n(\sqrt{q} + \sqrt{n}))$ 。

[Ynoi2017] 由乃的 OJ

题目链接: [洛谷 P5354 \(洛谷月赛\)](#), [BZOJ 4811](#)。

给定一棵 n 个节点的树, 每个节点有一个位运算符 opt 和一个权值 x , opt 有 $\&$, $|$, \wedge 三种。

有 q 次询问, 每次给定 x, y, z , 你需要初始选定一个数 $v \in [0, z]$, 然后依次经过 $x \rightarrow y$ 的简单路径, 每经过一个节点, v 就变成 $v \text{ opt } x$ 。问 v 最后的最大值。

$0 \leq n, m \leq 10^5, 0 \leq k \leq 64$ 。

时间限制 250ms, 空间限制 128MB。

Solution

我们将 $u \rightarrow v$ 的路径抠出来, 问题相当于一棵链的情形。

考虑贪心, 我们从最高位往最低位贪, 如果当前位取 0, 最终能变成 1, 那么我们这一位一定取 0; 否则如果当前位取 1, 最终能变成 1, 我们这一位一定取 1; 否则这一维取 0。这些都很显然, 因此我们可以把这个事情丢给树剖, 在线段树上维护这些信息。

由于一共只有 64 位, 我们可以把这些位的信息全压在一个 `ull` 里, 因此线段树上信息的合并是 $\mathcal{O}(1)$ 的。时间复杂度 $\mathcal{O}(n \log^2 n)$ 。

当然, 我们可以用 LCT 去维护链上信息, 时间复杂度为 $\mathcal{O}(n \log n)$ 。

空间复杂度均为 $\mathcal{O}(n)$ 。

[Ynoi2017] 由乃打扑克

题目链接: [洛谷 P5356](#)。

给定一个长为 n 的序列 a , 实现两种操作:

- 查询区间 $[l, r]$ 的第 k 小值;
- 给区间 $[l, r]$ 加上 k 。

$1 \leq n, m \leq 10^5, -2 \cdot 10^4 \leq a_i, k \leq 2 \cdot 10^4$ 。

时间限制 2000ms, 空间限制 128MB。

Solution

我们对于每个块记录块内数排序后的结果, 设块长为 B :

- 对于修改: 整块复杂度 $\mathcal{O}(\frac{n}{B})$, 散块归并重构 $\mathcal{O}(B)$;
- 对于查询: 二分答案, 整块查询 $\mathcal{O}(\frac{n}{B} \log B)$, 散块暴力枚举 $\mathcal{O}(B)$ 。

整体复杂度 $\mathcal{O}(q(\frac{n}{B} \log B + B) \log n)$, 取 $B = \sqrt{n \log n}$ 得 $\mathcal{O}(q\sqrt{n \log n} \log n)$ 。

我们考虑每次二分答案后, 散块暴力枚举的代价很高。所以, 我们在二分前, 先将散块中在 $[l, r]$ 区间的数提取出来, 临时归并成一个假的整块, 这样二分答案后, 查询复杂度仅仅为 $\mathcal{O}(\frac{n}{B} \log B)$ 。

时间复杂度 $\mathcal{O}(q(\frac{n}{B} \log B \log n + B))$, 取 $B = \sqrt{n \log n}$ 得 $\mathcal{O}(q\sqrt{n \log n})$ 。

这样足以通过本题, 但我们可以通过分散层叠算法继续优化 (即多序列二分)。

时间复杂度 $\mathcal{O}(q\sqrt{n \log n})$ 。

空间复杂度 $\mathcal{O}(n)$ 。

[Ynoi2018] 五彩斑斓的世界

题目链接: [洛谷 P4117](#), [CF896E](#), BZOJ 5143。

给定一个长为 n 的序列 a , m 次询问, 支持两种操作:

- 将区间 $[l, r]$ 中 $> x$ 的数减去 x ;
- 查询区间 $[l, r]$ 中 x 的出现次数。

$1 \leq n \leq 10^6, 1 \leq m \leq 5 \times 10^5, 1 \leq a_i, x \leq 10^5 + 1$ 。

时间限制 8000ms, 空间限制 64MB。

Solution

突刺贯穿的第二分块。

假设值域和 n 同阶, 因为每次是减, 所以最大值**单调不增**。

设当前操作为减 x , 当前最大值为 $maxx$, 则

- 若 $maxx \geq 2 \times x$, 则可以将 $[1, x]$ 的数整体 $+x$, 然后**整体减标记** $+x$
- 若 $maxx < 2 \times x$, 则暴力将 $[x + 1, maxx]$ 的数 $-x$

可以发现, 定义势能函数 $\Phi = \max - \min$, 每次操作使得 $\Delta \Phi = x$, 暴力操作至多 $max - min$ 下就会出现 $\Phi = 0$, 因此暴力维护复杂度是对的。

关于整体平移, 可以用 dsu 来维护, 这里复杂度是 $O(\max\{a_i\} \alpha(\max\{a_i\}))$ 的。

本题空间卡的很紧, 可以考虑**逐块处理**的方式。对于整块直接平移 (对于同一块来说, 由势能函数知均摊 $O(n \alpha(n))$), 散块暴力重构 (因为对于一个查询最多 2 个散块, 所以均摊 $2m$ 次暴力重构, 这里复杂度 $O(m \sqrt{n})$)。

时间复杂度 $O(n \sqrt{m} \alpha(\max\{a_i\}))$ 。

空间复杂度 $O(n + m + \max\{a_i\})$ 。

[code](#)

[Ynoi2018] 末日时在做什么？有没有空？可以来拯救吗？

题目链接: [洛谷 P4118](#), [BZOJ 5144](#)。

给定一个长为 n 的序列 a , 实现两种操作:

- `1 l r x`: 给区间 $[l, r]$ 加上 x ;
- `2 l r`: 查询区间 $[l, r]$ 最大子段和, 可以不选。

$1 \leq n, m \leq 10^5, |a_i|, |x| \leq 10^9$, **保证任意时刻** $|a_i| \leq 2 \times 10^9$ 。

时间限制 1500ms, 空间限制 64MB。

Solution

深浅值藏的第六分块。

待填坑。

[Ynoi2018] 未来日记

题目链接: [洛谷 P4119](#), [hdu 6079 \(杭电多校\)](#), [BZOJ 5145](#)。

给定一个长为 n 的序列 a , 实现两种操作:

- 把区间 $[l, r]$ 中所有值为 x 的改为 y ;
- 查询区间 $[l, r]$ 内第 k 小值。

$1 \leq n, m, a_i \leq 10^5$ 。

时间限制 1000ms, 空间限制 512MB。

Solution

望月悲叹的最初分块。

待填坑。

[Ynoi2018] 天降之物

题目链接: [洛谷 P5397](#), [51nod 2046 \(SD省队集训2019 Day8之“有没有空”\)](#)。

给定一个长为 n 的序列 a , 实现两种操作:

- 将序列中所有值为 x 的数的值变为 y ;
- 找到最小的 $|i - j|$ 满足 $a_i = x, a_j = y$ 。

$1 \leq n, q \leq 10^5, 1 \leq a_i \leq 10^5$, 强制在线。

时间限制 500ms, 空间限制 256MB。

Solution

弑尽破净的第四分块。

先忽略修改操作, 我们设定阈值 B , 对每个数出现次数根号分治。我们称出现次数 $\leq B$ 的为轻数, $> B$ 为重数。

对于查询分类讨论:

- x 为轻数, y 为轻数

我们对于每个数开个 vector 存下标, 暴力归并即可, $O(B)$;

- x 或 y 为重数

重数只有 $O(\frac{n}{B})$ 个, 我们对每个重数预处理 `ans[x][y]` 表示 x 和 y 的答案, 直接查表即可, $O(\frac{n}{B})$ 。

现在带上修改操作, 我们发现:

- x 为轻数, y 为轻数

我们暴力归并, 如果加起来个数 $> B$, y 进化为重数, $O(B)$;

- x 为重数, y 为重数

这样的合并只会发生 $O(\frac{n}{B})$ 次, 我们同样暴力合并, 再重新计算 `ans`, $O(\frac{n}{B})$ 。

- x 为轻数, y 为重数 (反过来同理)

我们发现这种情况很棘手, 于是考虑再次根号分治: 对于每个重数, 类似懒标记的思想, 我们开一个“附属集合”, 表示合并到它的轻数下标集合。一旦“附属集合”的大小 $> B$, 我们重构 y 即可。重构的次数同样是 $O(\frac{n}{B})$ 次, 因此这里复杂度仍是 $O(\frac{n}{B})$ 的。

因为额外维护了“附属集合”, 我们得重新审视询问:

- x 为轻数, y 为重数

这部分仍旧暴力归并即可, $O(B)$;

- x 为重数, y 为轻数

显然答案是 $\min(x \text{ 到 } y \text{ 的附属集合的答案}, \text{ans}[y][x])$, 前者暴力归并, 后者查表, $O(B)$;

- x, y 均为重数

显然答案是 $\min(x \text{ 的附属集合到 } y \text{ 的附属集合的答案}, \text{ans}[x][y], \text{ans}[y][x])$, 处理方式同上。值得一提的是, 因为有“附属集合”这个懒标记的存在, 所以 $\text{ans}[x][y]$ 和 $\text{ans}[y][x]$ 中可能有一个信息没及时更新, 所以两个都要都取 \min , 答案才正确, $O(B)$ 。

最后, 在处理上有一个小 trick, 我们可以通过重标号的方式规避 x 为重数, y 为轻数的情况, 实现起来会简单许多。

时间复杂度 $O(\frac{n}{B}n + nB + mB)$, 取 $B = \sqrt{n}$ 得 $O((n + m)\sqrt{n})$ 。

空间复杂度 $O(n\sqrt{n})$ 。

[Ynoi2018] GOSICK

题目链接: [洛谷 P5398](#), [CometOJ 1072 \(Contest #3\)](#)。

给定一个长为 n 的序列 a , 每次询问区间 $[l, r]$ 中有多少对 (i, j) 满足 $a_j | a_i$ 。

$1 \leq n, m, a_i \leq 5 \times 10^5$

时间限制 3000ms, 空间限制 128MB。

Solution

第十四分块。

考虑离线跑莫队。

令 $f([1, x], y)$ 表示 $a_1 \sim a_x$ 中 a_y 倍数的个数, $g([1, x], y)$ 表示 $a_1 \sim a_x$ 中 a_y 约数的个数。

- $[l, r] \rightarrow [l, r + 1]$, $\Delta = 1 + f([l, r], r + 1) + g([l, r], r + 1)$, 得到:

$$\Delta = 1 + f([1, r], r + 1) + g([1, r], r + 1) - f([1, l - 1], r + 1) - g([1, l - 1], r + 1)$$

- $[l, r] \rightarrow [l, r - 1]$, $\Delta = -1 - f([l, r - 1], r) - g([l, r - 1], r)$, 得到:

$$\Delta = -1 - f([1, r - 1], r) - g([1, r - 1], r) + f([1, l - 1], r) + g([1, l - 1], r)$$

关于 l 的转移同理, 跑莫队二次离线即可。

在二次离线跑扫描线的时候, 给 a_i 倍数的 `cnt` 数组加 1 的 case, 可以继续根号分治维护: 若 $a_i > \sqrt{n}$, 直接贡献; 否则因为离线以后仅有 q 组 $[l, r]$ 查询, 所以我们一起计算 $x \leq \sqrt{n}$ 对 $[l, r]$ 这个整体的贡献即可。

时间复杂度 $O((n + m)\sqrt{n})$, 空间复杂度 $O(n)$, 注意常数因子。

[Ynoi2018] 駄作

题目链接: [洛谷 P5399](#), [LOJ6612 \(THUPC2019 不用找树\)](#)。

给定一棵 n 个节点的树, 有 m 个询问, 每次给定参数 p_0, d_0, p_1, d_1 , 求:

$$\sum_{dis(p_0,a) \leq d_0} \sum_{dis(p_1,b) \leq d_1} dis(a,b)$$

$$1 \leq n, m \leq 10^5。$$

时间限制 4000ms，空间限制 256MB。

Solution

第七分块。

别问我为啥鸽了。因为就是鸽了。

[Ynoi2019 模拟赛] Yuno loves sqrt technology I

题目链接：[洛谷 P5046](#)。

给定一个长为 n 的排列 a ， m 次询问，每次查询一个区间的逆序对数。

强制在线。

$$1 \leq n, m \leq 10^5。$$

时间限制 750ms，空间限制 512MB。

Solution

考虑序列分块，预处理块与块之间的贡献，以及每个块前后缀和某整块的贡献。

查询的时候，两个散块的贡献暴力归并计算即可。

时间复杂度 $\mathcal{O}(n\sqrt{n})$ 。

空间复杂度 $\mathcal{O}(n\sqrt{n})$ 。

Bonus

存在 $\mathcal{O}(1.41)$ 的做法。

[Ynoi2019 模拟赛] Yuno loves sqrt technology II

题目链接：[洛谷 P5047](#)。

给定一个长为 n 的序列 a ， m 次询问，每次查询一个区间的逆序对数。

$$1 \leq n, m \leq 10^5。$$

时间限制 250ms，空间限制 32MB。

Solution

考虑莫队二次离线。

先将 a 序列离散化。

记 $f([l, r], x)$ 表示 $a[l..r]$ 中 $> a[x]$ 的个数， $g([l, r], x)$ 表示 $a[l..r]$ 中 $< a[x]$ 的个数。

例如，从 $[l, r]$ 到 $[l, r+1]$ ， $\Delta = f([l, r], r+1) = f([1, r], r+1) - f([1, l-1], r+1)$ 。

$f([1, x], x+1), g([1, x], x+1)$ 均可用树状数组 $\mathcal{O}(n \log n)$ 预处理。

对于 $f([1, x], l) \dots f([1, x], r)$ 考虑 $O(\sqrt{n})$ 插入, $O(1)$ 查询的值域分块。用 $sum[x]$ 表示块与块间的贡献, $qwq[x][i]$ 表示块内前后缀贡献。简单维护即可。

时间复杂度 $O(n\sqrt{n})$ 。

空间复杂度 $O(n)$ 。

[code](#)

[Ynoi2019 模拟赛] Yuno loves sqrt technology III

题目链接: [洛谷 P5048](#)。

给定一个长为 n 的序列 a , m 次询问, 每次查询区间众数出现次数。

强制在线。

$1 \leq n, m \leq 5 \times 10^5, 0 \leq a_i \leq 10^9$ 。

时间限制 2000ms, 空间限制 64MB。

Solution

显然 $[l, r]$ 到 $[l, r+1]$ 众数出现次数至多增加 1。

因此预处理 $F[l][r]$ 表示从块 l 到块 r 的区间众数, 用 `vector` 存每个数出现的下标, 然后两边零散块内暴力跑即可。

时间复杂度 $O(n\sqrt{n})$ 。

空间复杂度 $O(n)$ 。

[code](#)

Bonus

存在 $O(n^{1.48541})$ 的做法, 依赖于**快速矩阵乘法**。

[Ynoi2020] 美好的每一天, 不连续的存在~

题目链接: [洛谷 P6580 \(洛谷月赛\)](#)。

给定一个数组 a 以及一棵 n 个节点的树, 每个点有一个颜色 $c \in [1, x]$ 。

有 m 次查询, 每次询问树上只保留 $[l, r]$ 内的点, 设一个极大连通块中出现奇数次数的颜色有 t 种, 则对答案的贡献为 a_t , 答案即为 $\sum_t a_t$, 询问间相互独立。

$1 \leq n, m \leq 10^5, 1 \leq x, A_i \leq 10^4$ 。

Solution

先考虑全局询问, 我们对每个联通块开一个大小为 10^4 的 `bitset`, 维护每种颜色出现次数奇偶性。总合并次数是 $O(n)$ 的, 因此复杂度 $O(\frac{nx}{\omega})$ 。我们发现其实可以把 `bitset` 换成压位 trie, 做到 $O(n \log_{\omega} x)$ 。

但是每个位置的势能不等, 所以我们需要按照势能进行分块。具体而言, 我们从后往前扫, 计算出每个点的合并势能。紧接着从前往后, 每次势能和 $\geq B$ 就分一块。

最后正常跑回滚莫队即可。由于“回滚”和“加入”操作本质上都是在做集合的对称差，所以是可以“回滚”的。

时间复杂度 $\mathcal{O}(\frac{n \log_{\omega} x}{B} n \log_{\omega} x + Bm)$ ，取 $B = \frac{n \log_{\omega} x}{\sqrt{m}}$ ，做到 $\mathcal{O}(n\sqrt{m} \log_{\omega} x)$ 。

注意在实现的时候，每个块 $[L, R]$ 我们应当把 R 这个位置扔进右指针单调移动， $[L, R)$ 区间暴力扩展和回滚。原因是我们每次在势能和 $\geq B$ 的时候才分成一块，对于 R 位置的势能大小是不可控的。