初级数据结构 2

cdsfcesf

Tsinghua University

2021 年 8 月 2 日

树状数组

线段树

线段树的可持久化

综合练习

树状数组

树状数组

可用于维护前缀信息。

树状数组

可用于维护前缀信息。
优点：代码短，速度快。

树状数组

可用于维护前缀信息。
优点：代码短，速度快。

Example

给定长为 $n(n \le 10^5)$ 的数组 $a_1, a_2, \ldots, a_n$，支持操作：

1. 将第 i 个数 a_i 加 v
2. 询问区间 $[l, r]$ 中数的和

树状数组

树状数组

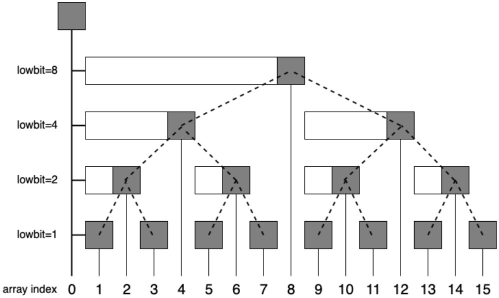
lowbit(i)：只保留 i 的二进制表示中最低位的 1。

树状数组

$\text{lowbit}(i)$：只保留 i 的二进制表示中最低位的 1。
树状数组中，节点 i 表示的区间为 $[i-\text{lowbit}(i)+1, i]$。

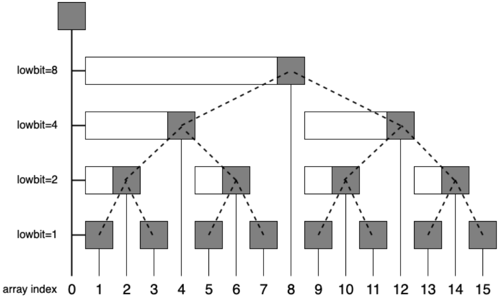
树状数组

lowbit(i)：只保留 i 的二进制表示中最低位的 1。
树状数组中，节点 i 表示的区间为 $[i - \text{lowbit}(i) + 1, i]$。



树状数组

lowbit(i)：只保留 i 的二进制表示中最低位的 1。
树状数组中，节点 i 表示的区间为 $[i-\text{lowbit}(i)+1, i]$。



记录 b_i 表示 i 对应的区间和。

单点修改

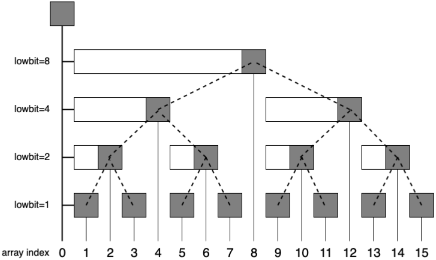
修改 i 时，寻找包含 i 的所有区间：

单点修改

修改 i 时，寻找包含 i 的所有区间：
第一个包含 i 且 lowbit 比 i 大的节点是 $i + \text{lowbit}(i)$。

单点修改

修改 i 时，寻找包含 i 的所有区间：
第一个包含 i 且 lowbit 比 i 大的节点是 $i+\text{lowbit}(i)$。



前缀查询

前缀查询

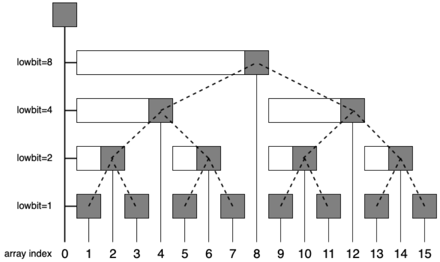
查询区间 $[1, i]$ 的和时，由 i 可直接得到 $[i - \text{lowbit}(i) + 1, i]$ 的和。

前缀查询

查询区间 $[1, i]$ 的和时，由 i 可直接得到 $[i - \text{lowbit}(i) + 1, i]$ 的和。接下来需要查询 $[1, i - \text{lowbit}(i)]$ 的区间和，重复上面的步骤即可。

前缀查询

查询区间 $[1, i]$ 的和时，由 i 可直接得到 $[i - \text{lowbit}(i) + 1, i]$ 的和。接下来需要查询 $[1, i - \text{lowbit}(i)]$ 的区间和，重复上面的步骤即可。



树状数组

```
int lowbit(int x) { return x & -x; }

void add(int i, int v) {
    while(i <= n) {
        b[i] += v;
        i += lowbit(i);
    }
}

int query(int i) {
    int ret = 0;
    while(i > 0) {
        ret += b[i];
        i -= lowbit(i);
    }
    return ret;
}
```

差分

Example

长为 $n (n \leq 10^5)$ 的数组 $a_1, a_2, \ldots, a_n$，每次操作将区间 $[l, r]$ 内的数全部加 v，或是查询 a_i 的值。

差分

Example

长为 $n(n \leq 10^5)$ 的数组 $a_1, a_2, \ldots, a_n$，每次操作将区间 $[l, r]$ 内的数全部加 v，或是查询 a_i 的值。

- 差分序列 $d_i = a_i - a_{i-1}$ $(a_0 = 0)$

差分

Example

长为 $n(n \le 10^5)$ 的数组 $a_1, a_2, \ldots, a_n$，每次操作将区间 $[l, r]$ 内的数全部加 v，或是查询 a_i 的值。

- 差分序列 $d_i = a_i - a_{i-1}$ $(a_0 = 0)$
- 区间 $[l, r]$ 加 v 时，将 d_l 加 v，d_{r+1} 减 v。

差分

Example

长为 $n(n \le 10^5)$ 的数组 $a_1, a_2, \ldots, a_n$，每次操作将区间 $[l, r]$ 内的数全部加 v，或是查询 a_i 的值。

- 差分序列 $d_i = a_i - a_{i-1}$ $(a_0 = 0)$
- 区间 $[l, r]$ 加 v 时，将 d_l 加 v，d_{r+1} 减 v。
- 树状数组维护 $\{d_n\}$ 的前缀和。

区间修改

Description

长为 $n(n \le 10^5)$ 的数组 $a_1, a_2, \dots, a_n$，每次操作将区间 $[l, r]$ 内的数全部加 v，或是查询区间 $[l, r]$ 内数的和。

区间修改

区间修改

考虑用差分数组 $d_i = a_i - a_{i-1}$ 表示出 $[1, R]$ 的区间和 ($a_0 = 0$):

区间修改

考虑用差分数组 $d_i = a_i - a_{i-1}$ 表示出 $[1, R]$ 的区间和 $(a_0 = 0)$:

$$\begin{aligned}
\sum_{k=1}^{R} a_k &= \sum_{k=1}^{R} \sum_{i=1}^{k} d_i \\
&= \sum_{i=1}^{R} \sum_{k=i}^{R} d_i \\
&= \sum_{i=1}^{R} (R - i + 1) d_i = (R+1) \sum_{i=1}^{R} d_i - \sum_{i=1}^{R} i d_i.
\end{aligned}$$

区间修改

考虑用差分数组 $d_i = a_i - a_{i-1}$ 表示出 $[1, R]$ 的区间和 $(a_0 = 0)$:

$$\begin{aligned}\sum_{k=1}^{R} a_k &= \sum_{k=1}^{R}\sum_{i=1}^{k} d_i \\ &= \sum_{i=1}^{R}\sum_{k=i}^{R} d_i \\ &= \sum_{i=1}^{R}(R-i+1)d_i = (R+1)\sum_{i=1}^{R} d_i - \sum_{i=1}^{R} id_i.\end{aligned}$$

维护 $\{d_i\}$ 和 $\{id_i\}$ 两个序列的前缀和。

逆序对

Description

给定序列 $a_1, \ldots, a_n$ $(n \leq 10^5)$，求有多少对 $i < j$ 满足 $a_i > a_j$。

逆序对

逆序对

- 对每个 i，它的前面有多少个数比它大；

逆序对

- 对每个 i，它的前面有多少个数比它大；
- 先离散化，对值域建树状数组；

逆序对

- 对每个 i，它的前面有多少个数比它大；
- 先离散化，对值域建树状数组；
- 从左往右扫描一次，在每个位置需要查询权值的前缀和，然后把这个数加入树状数组；

逆序对

- 对每个 i，它的前面有多少个数比它大；
- 先离散化，对值域建树状数组；
- 从左往右扫描一次，在每个位置需要查询权值的前缀和，然后把这个数加入树状数组；
- 时间复杂度 $O(n \log n)$。

二维树状数组

Description

给定 $n \times m$ 的二维数组 a_{ij}，有 Q 次操作，每次操作修改单个元素或查询某子矩阵的和。

Constraints

$n, m \leq 1000, Q \leq 10^5$

二维树状数组

二维树状数组

- 记 b_{ij} 表示从 $(i - \text{lowbit}(i) + 1, j - \text{lowbit}(j) + 1)$ 到 (i, j) 的子矩阵的和。

二维树状数组

- 记 b_{ij} 表示从 $(i - \text{lowbit}(i) + 1, j - \text{lowbit}(j) + 1)$ 到 (i, j) 的子矩阵的和。
- 查询和修改时先枚举第一维，再枚举第二维。

二维树状数组

- 记 b_{ij} 表示从 $(i-\text{lowbit}(i)+1, j-\text{lowbit}(j)+1)$ 到 (i,j) 的子矩阵的和。
- 查询和修改时先枚举第一维，再枚举第二维。
- 时间复杂度 $O(Q\log n\log m)$。

树状数组

线段树

线段树的可持久化

综合练习

线段树

可用于维护区间信息。

线段树

可用于维护区间信息。

Example

给定长为 $n(n \leq 10^5)$ 的数组 $a_1, a_2, \ldots, a_n$，支持操作：

1. 将第 i 个数 a_i 加 v (单点操作)
2. 将区间 $[l, r]$ 中的数加 v (区间操作)
3. 询问区间 $[l, r]$ 的和/最小值/最大值

线段树

一棵二叉树，在每个节点维护一段区间的信息（如最值、和）：

线段树

一棵二叉树，在每个节点维护一段区间的信息（如最值、和）：

- 根节点区间为 $[1, n]$

线段树

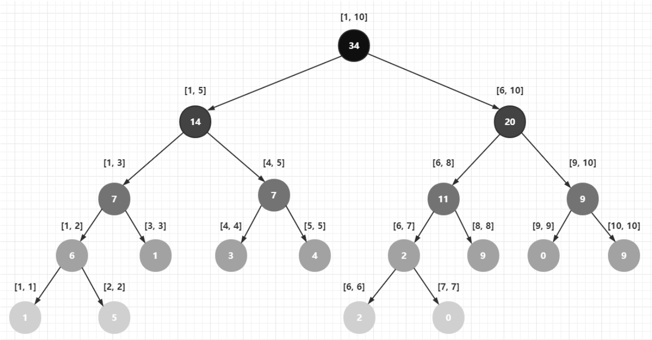
一棵二叉树，在每个节点维护一段区间的信息（如最值、和）：

- 根节点区间为 $[1, n]$
- 对节点 $[L, R]$，设中点 $mid = \lfloor \frac{L+R}{2} \rfloor$。把该节点从 mid 处分开，左子节点为 $[L, mid]$，右子节点为 $[mid + 1, R]$。

线段树

一棵二叉树，在每个节点维护一段区间的信息（如最值、和）：

- 根节点区间为 $[1, n]$
- 对节点 $[L, R]$，设中点 $mid = \lfloor \frac{L+R}{2} \rfloor$。把该节点从 mid 处分开，左子节点为 $[L, mid]$，右子节点为 $[mid+1, R]$。



线段树

- 树高 $O(n\log n)$，总节点数 $2n-1$
- 堆式编号：根节点 1；点 x 的左儿子为 $2x$，右儿子为 $2x+1$

线段树

- 树高 $O(n \log n)$，总节点数 $2n - 1$
- 堆式编号：根节点 1；点 x 的左儿子为 $2x$，右儿子为 $2x + 1$

建树

根节点为 $[1, n]$。考虑根节点为 $[L, R]$ 时：

线段树

- 树高 $O(n\log n)$，总节点数 $2n-1$
- 堆式编号：根节点 1；点 x 的左儿子为 $2x$，右儿子为 $2x+1$

建树

根节点为 $[1,n]$。考虑根节点为 $[L,R]$ 时：

- 设 $m=\lfloor\frac{L+R}{2}\rfloor$，对区间 $[L,m]$ 和 $[m+1,R]$ 递归建树

线段树

- 树高 $O(n\log n)$，总节点数 $2n-1$
- 堆式编号：根节点 1；点 x 的左儿子为 $2x$，右儿子为 $2x+1$

建树

根节点为 $[1,n]$。考虑根节点为 $[L,R]$ 时：

- 设 $m=\lfloor\frac{L+R}{2}\rfloor$，对区间 $[L,m]$ 和 $[m+1,R]$ 递归建树
- 合并子区间 $[L,m]$ 和 $[m+1,R]$ 的信息

单点修改/查询

单点修改/查询

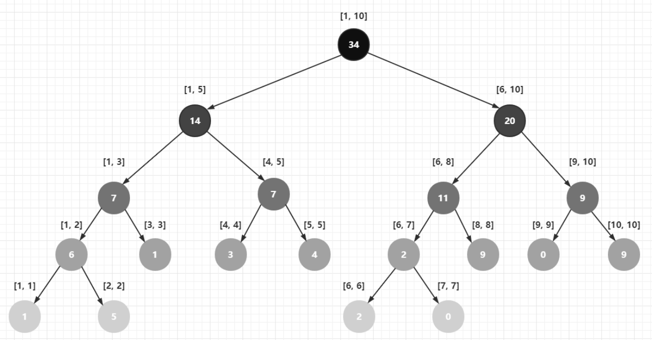
单点查询：从上往下找到对应叶子节点。

单点修改/查询

单点查询：从上往下找到对应叶子节点。
单点修改：先从上往下找到对应叶子，修改该节点，再从下往上更新。

单点修改/查询

单点查询：从上往下找到对应叶子节点。
单点修改：先从上往下找到对应叶子，修改该节点，再从下往上更新。



区间查询

区间查询

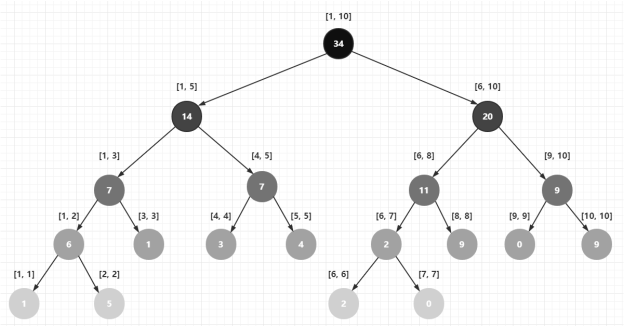
查询区间 $[l, r]$ 时，可将其拆成最多 $2\log n$ 个不相交区间。合并这些区间可得到答案。

区间查询

查询区间 $[l, r]$ 时，可将其拆成最多 $2\log n$ 个不相交区间。合并这些区间可得到答案。
从线段树上从上往下寻找。

区间查询

查询区间 $[l, r]$ 时，可将其拆成最多 $2\log n$ 个不相交区间。合并这些区间可得到答案。
从线段树上从上往下寻找。



区间修改

对区间 $[l,r]$ 加 v：同样拆成最多 $2\log n$ 个不相交区间，并修改这些区间。

区间修改

对区间 $[l, r]$ 加 v：同样拆成最多 $2\log n$ 个不相交区间，并修改这些区间。

- 在一个节点做区间修改时，不立刻修改它的子节点

区间修改

对区间 $[l, r]$ 加 v：同样拆成最多 $2\log n$ 个不相交区间，并修改这些区间。

- 在一个节点做区间修改时，不立刻修改它的子节点
- 记录懒标记：每个节点 x 记录还没有加到子树中的值的总和 tag_x

区间修改

对区间 $[l,r]$ 加 v：同样拆成最多 $2\log n$ 个不相交区间，并修改这些区间。

- 在一个节点做区间修改时，不立刻修改它的子节点
- 记录懒标记：每个节点 x 记录还没有加到子树中的值的总和 tag_x
- 查询经过某节点时，需要先把这个点的标记下放到左右儿子节点

标记永久化

- 不下放标记，而是在查询的时候合并所有经过的标记。

P2023

Description

一个长度为 n 的序列 a_n。有 m 个操作：

1. 将区间 $[l, r]$ 中的数加 v
2. 将区间 $[l, r]$ 中的数乘以 v
3. 询问区间 $[l, r]$ 的和对 P 取模的值

Constraints

$n, m \leq 10^5, 0 \leq p, v, a_i \leq 10^9$

P2023

P2023

- 两种标记：乘法标记和加法标记

P2023

- 两种标记：乘法标记和加法标记
- 规定：计算实际区间和时，先计算乘法标记，再计算加法标记

P2023

- 两种标记：乘法标记和加法标记
- 规定：计算实际区间和时，先计算乘法标记，再计算加法标记
- 下放标记时，乘法标记会影响加法标记，但加法标记不影响乘法标记

P4513

Description

一个长度为 n 的序列 a_n。有 m 个操作，每次修改序列中的一个数，或者给定区间 $[l,r]$，求该区间内的最大子段和。

Constraints

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

P4513

在每个节点处维护：

P4513

在每个节点处维护：

- 区间内最大子段和
- 区间最大前缀和
- 区间最大后缀和

P4513

在每个节点处维护：

- 区间内最大子段和
- 区间最大前缀和
- 区间最大后缀和

合并时分类讨论即可。

P2894 Hotel

Description

有 n 个房间，编号为 $0 \sim n-1$。有 m 个请求：

1. Check in：找到最靠前的一段连续长度为 x 的房间，然后这些房间被占用
2. Check out：一段连续的房间状态变为空闲

Constraints

$n, m \leq 5 \times 10^4$

P2894

- 类似上一题，在每个节点处维护：区间内最长全 0 序列长度、最长全 0 前缀长度、最长全 0 后缀长度。

P2894

- 类似上一题，在每个节点处维护：区间内最长全 0 序列长度、最长全 0 前缀长度、最长全 0 后缀长度。
- 查询时，在线段树上二分找到最靠前的答案。

P5490 扫描线 (模板)

Description

求平面上的 $n(n \leq 10^5)$ 个矩形的面积并。坐标范围 $[0, 10^9]$。

扫描线

扫描线

离散化坐标，并按照 x 从小到大的顺序扫描。
每一步加入或删除一个矩形（一根线段），线段树维护被覆盖部分。

扫描线

离散化坐标，并按照 x 从小到大的顺序扫描。
每一步加入或删除一个矩形（一根线段），线段树维护被覆盖部分。
画图：

扫描线

扫描线

为维护被覆盖部分的面积，节点需记录：
区间实际长度、被覆盖部分长度、最小值、最小值部分长度。

扫描线

为维护被覆盖部分的面积，节点需记录：
区间实际长度、被覆盖部分长度、最小值、最小值部分长度。
画图：

树状数组

线段树

线段树的可持久化

综合练习

可持久化

- 可持久化：支持保存和访问数据结构的历史版本。

Example

可持久化

- 可持久化：支持保存和访问数据结构的历史版本。

Example

维护一个序列，支持操作：

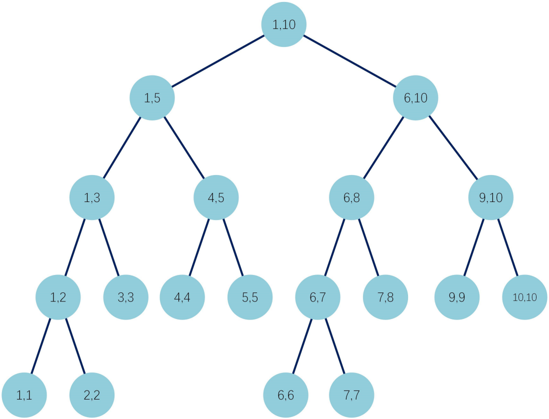
- 单点/区间修改/查询
- 回退到某一次操作之前的状态

可持久化线段树

- 在单次修改后，产生变化的节点只有 $O(\log n)$ 个。
- 只需为这些节点开辟新的存储空间。

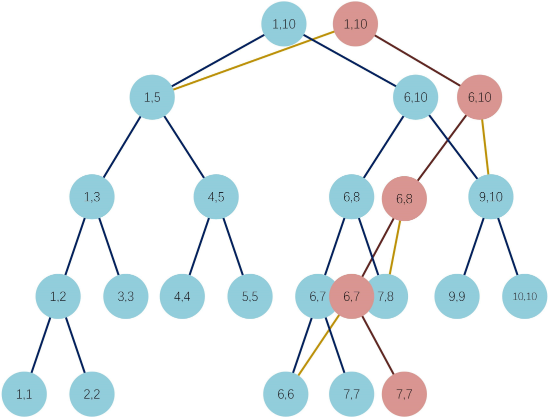
可持久化线段树

- 在单次修改后，产生变化的节点只有 $O(\log n)$ 个。
- 只需为这些节点开辟新的存储空间。



可持久化线段树

- 在单次修改后，产生变化的节点只有 $O(\log n)$ 个。
- 只需为这些节点开辟新的空间。



可持久化线段树

- 不能用堆式编号，需要记录子树节点编号，并动态开点。

可持久化线段树

- 不能用堆式编号，需要记录子树节点编号，并动态开点。
- 区间操作需要标记永久化。

可持久化线段树

- 不能用堆式编号，需要记录子树节点编号，并动态开点。
- 区间操作需要标记永久化。
- 单次操作的时间和空间复杂度为 $O(\log n)$。

可持久化并查集

- 用可持久化线段树可实现可持久化数组：单点修改、单点查询；

可持久化并查集

- 用可持久化线段树可实现可持久化数组：单点修改、单点查询；
- 将并查集的 `fa[]` 数组可持久化即可。

可持久化并查集

- 用可持久化线段树可实现可持久化数组：单点修改、单点查询；
- 将并查集的 `fa[]` 数组可持久化即可。
- 期望复杂度 $O(n\log n\alpha(n))$。

P3834 可持久化线段树 (模板)

Description

给定 n 个整数构成的序列 $\{a_n\}$，将对于指定的闭区间 $[l, r]$ 查询其区间内的第 k 小值。

Constraints

$1 \leq n, m \leq 2 \times 10^5, |a_i| \leq 10^9$

P3834

考虑查询区间是 $[1, n]$ 的情况：

P3834

考虑查询区间是 $[1, n]$ 的情况：

- 离散化后建立值域线段树，区间 $[L, R]$ 保存落在 $[L, R]$ 内的数的个数。

P3834

考虑查询区间是 $[1, n]$ 的情况：

- 离散化后建立值域线段树，区间 $[L, R]$ 保存落在 $[L, R]$ 内的数的个数。
- 在线段树上二分找到第 k 大的值。

P3834

考虑查询区间是 $[1, n]$ 的情况：

- 离散化后建立值域线段树，区间 $[L, R]$ 保存落在 $[L, R]$ 内的数的个数。
- 在线段树上二分找到第 k 大的值。
- 如果左子树的总和 $\leq k$，则向左子树走，否则向右子树走。

主席树

主席树

区间 $[L, R]$ 的值域线段树，可以用 $[1, R]$ 和 $[1, L-1]$ 的两棵线段树相减得到。

主席树

区间 $[L, R]$ 的值域线段树，可以用 $[1, R]$ 和 $[1, L-1]$ 的两棵线段树相减得到。

暴力：开 n 棵线段树，时间复杂度 $O(n^2)$。

主席树

区间 $[L, R]$ 的值域线段树，可以用 $[1, R]$ 和 $[1, L-1]$ 的两棵线段树相减得到。

暴力：开 n 棵线段树，时间复杂度 $O(n^2)$。

- 第 i 棵线段树在第 $i-1$ 棵线段树的基础上修改了一个值。

主席树

区间 $[L, R]$ 的值域线段树，可以用 $[1, R]$ 和 $[1, L-1]$ 的两棵线段树相减得到。

暴力：开 n 棵线段树，时间复杂度 $O(n^2)$。

- 第 i 棵线段树在第 $i-1$ 棵线段树的基础上修改了一个值。
- 将线段树可持久化，共 n 个历史版本。查询时同时访问两个历史版本，并把它们相减。

P4216

Description

`https://www.luogu.com.cn/problem/P4216`

Constraints

$n, Q \le 2 \times 10^5$

P4216

P4216

- 首先离线操作，得到每个点开始收集情报的时间 t。

P4216

- 首先离线操作，得到每个点开始收集情报的时间 t。
- 对一次 T 时刻的查询，相当于询问路径 (X, Y) 上有多少个点的权值 $\leq T - C$。

P4216

- 首先离线操作，得到每个点开始收集情报的时间 t。
- 对一次 T 时刻的查询，相当于询问路径 (X, Y) 上有多少个点的权值 $\leq T - C$。
- 在树上建立主席树，一个点的前驱是它的父亲。

P4216

- 首先离线操作，得到每个点开始收集情报的时间 t。
- 对一次 T 时刻的查询，相当于询问路径 (X, Y) 上有多少个点的权值 $\leq T - C$。
- 在树上建立主席树，一个点的前驱是它的父亲。
- 路径 (X, Y) 的树为 $X + Y - \text{LCA}(X, Y) - \text{fa}(\text{LCA}(X, Y))$。

树状数组

线段树

线段树的可持久化

综合练习

P4145

Description

给定长为 n 的序列 $a_1, a_2, \ldots, a_n$，支持两种操作：

1. 将区间 $[l, r]$ 中的数开平方根 (下取整)
2. 询问区间 $[l, r]$ 中数的和

Constraints

$1 \leq n, m \leq 10^5, 0 < a_i \leq 10^{12}$

P4145

P4145

注意到每个数被开根最多 6 次后就会变成 1。

P4145

注意到每个数被开根最多 6 次后就会变成 1。

- 法 1：在线段树上暴力开根；把全 1 区间打上标记。

P4145

注意到每个数被开根最多 6 次后就会变成 1。

- 法 1：在线段树上暴力开根；把全 1 区间打上标记。
- 法 2：树状数组暴力单点修改。在修改时跳过所有的 1，可以用并查集实现。

P4145

注意到每个数被开根最多 6 次后就会变成 1。

- 法 1：在线段树上暴力开根；把全 1 区间打上标记。
- 法 2：树状数组暴力单点修改。在修改时跳过所有的 1，可以用并查集实现。

时间复杂度 $O(n \log n)$。

P1972

Description

有一个长为 n 的序列 $a_1, \dots, a_n$。有 m 次询问，每次询问给定区间 $[l, r]$，问 $[l, r]$ 中出现过多少种不同的数。

Constraints

$1 \le n, m, a_i \le 10^6$

P1972

P1972

- 离线询问，从左往右考虑每个下标 i，并处理右端点在 i 的询问。

P1972

- 离线询问，从左往右考虑每个下标 i，并处理右端点在 i 的询问。
- 一个数只有最后一次出现的位置对区间的答案有贡献。

P1972

- 离线询问，从左往右考虑每个下标 i，并处理右端点在 i 的询问。
- 一个数只有最后一次出现的位置对区间的答案有贡献。
- 对每个数处理出它上一次出现的位置 last_i。

P1972

- 离线询问，从左往右考虑每个下标 i，并处理右端点在 i 的询问。
- 一个数只有最后一次出现的位置对区间的答案有贡献。
- 对每个数处理出它上一次出现的位置 last_i。
- 树状数组支持单点加减，查询前缀和。

P3178 树上操作

Description

一棵 N 个点的有根树，树点有点权。有 M 个操作，分为三种：

- 把某个节点 x 的点权增加 v
- 把某个节点 x 为根的子树中所有点的点权都增加 v
- 询问某个节点 x 到根的路径中所有点的点权和

Constraints

$n, m \leq 10^5, |v| \leq 10^6$

DFS 序

DFS 序

- 先跑一次 DFS，子树对应 DFS 序上一段区间。

DFS 序

- 先跑一次 DFS，子树对应 DFS 序上一段区间。
- 在 DFS 序上，用线段树维护每个点到根的点权和。

DFS 序

- 先跑一次 DFS，子树对应 DFS 序上一段区间。
- 在 DFS 序上，用线段树维护每个点到根的点权和。
- 对操作 1，把 x 的子树内的点加 v。

DFS 序

- 先跑一次 DFS，子树对应 DFS 序上一段区间。
- 在 DFS 序上，用线段树维护每个点到根的点权和。
- 对操作 1，把 x 的子树内的点加 v。
- 对操作 3，单点查询 x 即可。

DFS 序

DFS 序

- 对操作 2，x 的子树内的点 i 需要加上 $(\mathrm{dep}(i) - \mathrm{dep}(x) + 1)v$。

DFS 序

- 对操作 2，x 的子树内的点 i 需要加上 $(\text{dep}(i) - \text{dep}(x) + 1)v$。
- 可分解为 $(1 - \text{dep}(x))v$ 和 $\text{dep}(i)v$。

DFS 序

- 对操作 2，x 的子树内的点 i 需要加上 $(\mathrm{dep}(i)-\mathrm{dep}(x)+1)v$。
- 可分解为 $(1-\mathrm{dep}(x))v$ 和 $\mathrm{dep}(i)v$。
- 第一部分直接区间加。

DFS 序

- 对操作 2，x 的子树内的点 i 需要加上 $(\text{dep}(i) - \text{dep}(x) + 1)v$。
- 可分解为 $(1 - \text{dep}(x))v$ 和 $\text{dep}(i)v$。
- 第一部分直接区间加。
- 第二部分只需要额外预处理每个节点区间内 dep_i 的和。

P4587

Description

一个可重数字集合 S 的神秘数定义为最小的不能被 S 的子集的和表示的正整数。例如 $S=\{1,1,1,4,13\}$，则 S 的神秘数为 8。给定 n 个正整数构成的序列 $\{a_n\}$。m 次询问，每次询问给定闭区间 $[l,r]$，求由 $a_l,a_{l+1},\dots,a_r$ 组成的集合的神秘数。

Constraints

$1\le n,m\le 10^5,\sum a_i\le 10^9$

P4587

如果没有区间查询：

P4587

如果没有区间查询：

- 从小到大考虑每一个数，设当前能表示出 $[1, x]$ 的数；

P4587

如果没有区间查询：

- 从小到大考虑每一个数，设当前能表示出 $[1, x]$ 的数；
- 如果下一个数 $a_i \geq x + 2$，则答案是 $x + 1$。

P4587

如果没有区间查询：

- 从小到大考虑每一个数，设当前能表示出 $[1, x]$ 的数；
- 如果下一个数 $a_i \geq x + 2$，则答案是 $x + 1$。
- 否则，能表示的数的范围扩充到 $[1, x + a_i]$。

P4587

P4587

- 优化扩充过程：设一个 last，表示用 $\leq$ last 的数能凑出的范围是 $[1, x]$。

P4587

- 优化扩充过程：设一个 last，表示用 $\leq$ last 的数能凑出的范围是 $[1, x]$。
- 若下一个数 $> x+1$，则答案就是 $x+1$。

P4587

- 优化扩充过程：设一个 last，表示用 $\leq$ last 的数能凑出的范围是 $[1, x]$。
- 若下一个数 $> x+1$，则答案就是 $x+1$。
- 否则，求出所有在 $[last+1, x]$ 之间的数的和 sum。

P4587

- 优化扩充过程：设一个 last，表示用 $\leq$ last 的数能凑出的范围是 $[1, x]$。
- 若下一个数 $> x + 1$，则答案就是 $x + 1$。
- 否则，求出所有在 $[last + 1, x]$ 之间的数的和 sum。
- 则 x 增加到 $x + sum$，last 增加到原来的 x。

P4587

- 优化扩充过程：设一个 last，表示用 $\leq$ last 的数能凑出的范围是 $[1, x]$。
- 若下一个数 $> x + 1$，则答案就是 $x + 1$。
- 否则，求出所有在 $[last + 1, x]$ 之间的数的和 sum。
- 则 x 增加到 $x + sum$，last 增加到原来的 x。
- 扩充不超过 $O(\log(\sum a_i))$ 步（斐波那契）。

P4587

- 优化扩充过程：设一个 last，表示用 $\leq$ last 的数能凑出的范围是 $[1, x]$。
- 若下一个数 $> x+1$，则答案就是 $x+1$。
- 否则，求出所有在 $[last+1, x]$ 之间的数的和 sum。
- 则 x 增加到 $x+sum$，last 增加到原来的 x。
- 扩充不超过 $O(\log(\sum a_i))$ 步（斐波那契）。
- 主席树实现查询。

P4587

- 优化扩充过程：设一个 last，表示用 $\leq$ last 的数能凑出的范围是 $[1, x]$。
- 若下一个数 $> x+1$，则答案就是 $x+1$。
- 否则，求出所有在 $[last+1, x]$ 之间的数的和 sum。
- 则 x 增加到 $x+sum$，last 增加到原来的 x。
- 扩充不超过 $O(\log(\sum a_i))$ 步（斐波那契）。
- 主席树实现查询。
- 时间复杂度 $O(m\log n\log(\sum a_i))$。

P1712

Description

给定 n 个闭区间 $[l_i, r_i]$。需要从中选出 m 个区间，使得它们的交集不为空。一个选择方案的花费定义为：被选中的最长区间长度减去被选中的最短区间长度。求最小花费。

Constraints

$n \le 5 \times 10^5, m \le 2 \times 10^5, 0 \le l_i, r_i \le 10^9$

P1712

P1712

- 按区间长度排序，双指针扫一遍。

P1712

- 按区间长度排序，双指针扫一遍。
- 线段树区间加，维护区间最大值。