Advanced Segment Tree Technology

An Introduction to Primary Data Structures

He Liao

3-9-2025

The High School Affiliated to Xi'an Jiaotong University

Table of contents

Review

Introduction

上节课我们简单了解了一下线段树, 解决了模板问题, 并且介绍了几道使用线段树维护较复杂结合律的数据结构问题 (最大区间子段和).

本节课, 我们将对上节课的内容做以回顾, 并介绍一些其他线段树的知识以及更复杂的线段树问题.

Review of Last Class

线段树是一个用以维护区间带结合律信息的数据结构, 每个线段树上的结点代表一个区间 (l, r), 并附之以维护的信息 u (e.g. 区间和).

Review of Last Class

线段树是一个用以维护区间带结合律信息的数据结构, 每个线段树上的结点代表一个区间 (l, r), 并附之以维护的信息 u (e.g. 区间和).

线段树通过遍历一条树上从根结点到叶结点的路径来完成单点修改, 在达到叶结点之前进行一次方向判断, 在回到根结点的路经上做 push up.

Review of Last Class

线段树是一个用以维护区间带结合律信息的数据结构, 每个线段树上的结点代表一个区间 (l, r), 并附之以维护的信息 u (e.g. 区间和).

线段树通过遍历一条树上从根结点到叶结点的路径来完成单点修改, 在达到叶结点之前进行一次方向判断, 在回到根结点的路经上做 push up.

线段树区间操作的核心是将所询问/操作区间分割成若干线段树上的区间, 故线段树上的一个区间作为询问答案的一部分当且仅当该区间被询问区间完全包含. 具体实现上, 从根结点逐渐向下搜索, 每次检查该结点的左右结点是否有和询问区间包含的部分.

Review of Last Class

线段树是一个用以维护区间带结合律信息的数据结构, 每个线段树上的结点代表一个区间 (l, r), 并附之以维护的信息 u (e.g. 区间和).

线段树通过遍历一条树上从根结点到叶结点的路径来完成单点修改, 在达到叶结点之前进行一次方向判断, 在回到根结点的路经上做 push up.

线段树区间操作的核心是将所询问/操作区间分割成若干线段树上的区间, 故线段树上的一个区间作为询问答案的一部分当且仅当该区间被询问区间完全包含. 具体实现上, 从根结点逐渐向下搜索, 每次检查该结点的左右结点是否有和询问区间包含的部分.

对区间修改操作, 为降低复杂度, 引入懒标记的概念. 即暂时保存修改, 当且仅当在另一个操作中遍历到该结点时进行 push down 操作.

线段树是一个用以维护区间带结合律信息的数据结构, 每个线段树上的结点代表一个区间 (l, r), 并附之以维护的信息 u (e.g. 区间和).

线段树通过遍历一条树上从根结点到叶结点的路径来完成单点修改, 在达到叶结点之前进行一次方向判断, 在回到根结点的路经上做 push up.

线段树区间操作的核心是将所询问/操作区间分割成若干线段树上的区间, 故线段树上的一个区间作为询问答案的一部分当且仅当该区间被询问区间完全包含. 具体实现上, 从根结点逐渐向下搜索, 每次检查该结点的左右结点是否有和询问区间包含的部分.

对区间修改操作, 为降低复杂度, 引入懒标记的概念. 即暂时保存修改, 当且仅当在另一个操作中遍历到该结点时进行 push down 操作.

线段树的复杂度 (以维护加法为例):

- 建树: $\mathcal{O}(n)$;
- 单点操作: $\mathcal{O}(\log n)$;
- 区间操作: $\mathcal{O}(\log n)$.

Advanced Segment Tree Techniques

动态开点线段树

传统意义上的线段树需要 $4n$ 级别的空间. 当 n 很大时 (m 一般要小一些数量级), 空间无法承担. 此时采用动态开点的方法, 即一个结点当且仅当其被访问时被创造.

由于一次操作最多访问 $O(\log n)$ 级别的结点, 所以这样做的空间复杂度是 $O(m \log n)$. 动态开点线段树在线段树合并与分裂中非常重要 (毕竟你总不能建 n 棵完整的线段树). 同时, 动态开点的思想也在我们未来学习其他数据结构的过程中发挥着重要的作用.

标记永久化

如果确保标记在累加的过程中不会爆数据, 则可以不进行 push down 而在查询时统一计算标记带来的影响, 这样可以减小常数. 在树套树中标记永久化是必须的 (因为当子结点代表一棵树的时候 pushdown 的复杂度会出问题).

权值线段树

相较于普通线段树以一个序列 a 作为建树之基础, 权值线段树则以值域作为建树的基础. 形式化地, 设有整数列 $\{a_n\}$. 其值域为 $\mathbb{V}$. 假设线段树之结点为 $\mathcal{S}(l, r, u)$. 则 u 表示

$$u = \sum_{j=1}^{n} [l \le a_j \le r]$$

其中 [] 为艾佛森括号, 当括号内表达式为真时其值为 1, 否则为 0. 该表达在数论中常用.

通俗地讲, 线段树上区间 $[l, r]$ 对应的结点表示 $\{a_n\}$ 中有多少个元素的值在 l, r 之间.

或者, 你可以将权值线段树理解为为桶建树.

权值线段树

相较于普通线段树以一个序列 a 作为建树之基础, 权值线段树则以值域作为建树的基础. 形式化地, 设有整数列 $\{a_n\}$. 其值域为 $\mathbb{V}$. 假设线段树之结点为 $\mathcal{S}(l, r, u)$. 则 u 表示

$$u = \sum_{j=1}^{n} [l \le a_j \le r]$$

其中 [] 为艾佛森括号, 当括号内表达式为真时其值为 1, 否则为 0. 该表达在数论中常用.

通俗地讲, 线段树上区间 $[l, r]$ 对应的结点表示 $\{a_n\}$ 中有多少个元素的值在 l, r 之间.

或者, 你可以将权值线段树理解为为桶建树.

由于是对值域建树, 而值域通常来说相较于序列长度级别要大, 这个时候我们有两种方法:

1. 离散化, 将值域为 $\mathbb{V}$ 的 n 个数按照其大小关系离散为 $1 \sim n$ 的值域, 适用于仅关注各数之间大小关系而对于数的大小本身并不在意的题目 (e.g. 逆序对).
2. 动态开点.

权值线段树

权值线段树有一个很经典的应用: 查询第 k 小 (大) 元素. 具体来说, 我们先建权值线段树, 对某一个线段树的结点 $\mathcal{S}(l, r, u)$, 假设其左结点为 $S_{\mathrm{lc}}(l, r, u)$, 右结点为 $S_{\mathrm{rc}}(l, r, u)$, 并且我们当前要查询第 k 小. 则判断 u_{lc} 与 k 的大小关系: $u_{\mathrm{lc}} \geq k$, 则说明答案在左子树, 否则在右子树, 递归寻找到叶结点即可.

权值线段树

权值线段树有一个很经典的应用: 查询第 k 小 (大) 元素. 具体来说, 我们先建权值线段树, 对某一个线段树的结点 $\mathcal{S}(l,r,u)$, 假设其左结点为 $S_{\text{lc}}(l,r,u)$, 右结点为 $S_{\text{rc}}(l,r,u)$, 并且我们当前要查询第 k 小. 则判断 u_{lc} 与 k 的大小关系: $u_{\text{lc}} \geq k$, 则说明答案在左子树, 否则在右子树, 递归寻找到叶结点即可.

另一方面, 注意到 (虽然你们可能注意不到) 权值线段树有类似前缀和的性质. 即 $a_l \sim a_r$ 所对应的权值线段树事实上等于 $a_1 \sim a_r$ 的权值线段树减去 $a_1 \sim a_{l-1}$ 的权值线段树. 如果我们能够找到高效的做两树相减的方法以及开 n 棵线段树的方法, 我们就能够解决经典的静态区间第 k 小问题.

权值线段树

权值线段树有一个很经典的应用: 查询第 k 小 (大) 元素. 具体来说, 我们先建权值线段树, 对某一个线段树的结点 $\mathcal{S}(l,r,u)$, 假设其左结点为 $S_{\text{lc}}(l,r,u)$, 右结点为 $S_{\text{rc}}(l,r,u)$, 并且我们当前要查询第 k 小. 则判断 u_{lc} 与 k 的大小关系: $u_{\text{lc}} \geq k$, 则说明答案在左子树, 否则在右子树, 递归寻找到叶结点即可.

另一方面, 注意到 (虽然你们可能注意不到) 权值线段树有类似前缀和的性质. 即 $a_l \sim a_r$ 所对应的权值线段树事实上等于 $a_1 \sim a_r$ 的权值线段树减去 $a_1 \sim a_{l-1}$ 的权值线段树. 如果我们能够找到高效的做两树相减的方法以及开 n 棵线段树的方法, 我们就能够解决经典的静态区间第 k 小问题.

这实际上就是可持久化线段树的思想, 我们将在下节课进行更为细致的学习.

zkw 线段树

zkw 线段树

接下来介绍一个常数比较小的线段树, 可以像线段树一样解决区间修改区间查询问题.

[1]注意, 此处的 M 和序列长度 n 不同

[2]根节点除外

接下来介绍一个常数比较小的线段树, 可以像线段树一样解决区间修改区间查询问题.

考虑满二叉树的线段树 (即 $M = 2^k$)[1]. 如下图所示:

我们发现这棵线段树有一些很好的性质:

- 线段树上的结点数为 $2n - 1$;
- 树一共有 $h = \log n + 1$ 层;
- 第 i 层有 2^{i-1} 个结点, 每个结点对应的线段长度为 2^{h-i}.
- 若某结点的编号为 m, 则父结点编号为 $\frac{m}{2}$, 子结点编号为 $2m$ 和 $2m + 1$.
- 设有结点 p, q, 则 p, q 是兄弟结点当且仅当 $p \oplus q = 1$.
- 所有编号是偶数的结点都是左结点, 所有编号为奇数的结点都是右结点[2].

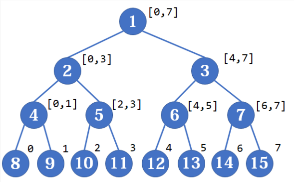

图 1: zkw segment tree

[1] 注意, 此处的 M 和序列长度 n 不同

[2] 根节点除外

建树及单点修改

我们发现, 在 zkw 线段树中我们很容易找到序列上 a_x 对应的线段树叶节点编号, 即 $x + N$.

于是我们只需要给线段树的 $S_{M+1} \sim S_{M+n}$ 赋值, 而对于非叶结点, 倒序 push up 即可.

同样, 对单点修改, 我们可以直接找到其所对应的叶结点, 再从下往上 push up.

单点修改下的区间查询

zkw 线段树的区间操作是相对比较难理解的一个部分. 假设查询的区间为 (l, r), 我们设立两个指针 p_l, p_r. 令其初始时分别指向 $\mathcal{S}(l-1, l-1, u)$ 和 $\mathcal{S}(r+1, r+1, u)$ (即查询区间转为开区间后对应的端点的叶节点).

同时向上跳指针, 直到跳到 p_l, p_r 为兄弟结点时停止. 对每一次跳跃, 若:

- p_l 指向的结点是左结点, 则累加右结点的答案;
- p_r 指向的结点是右结点, 则累加左结点的答案.

单点修改下的区间查询

zkw 线段树的区间操作是相对比较难理解的一个部分. 假设查询的区间为 (l, r), 我们设立两个指针 p_l, p_r. 令其初始时分别指向 $\mathcal{S}(l-1, l-1, u)$ 和 $\mathcal{S}(r+1, r+1, u)$ (即查询区间转为开区间后对应的端点的叶节点).

同时向上跳指针, 直到跳到 p_l, p_r 为兄弟结点时停止. 对每一次跳跃, 若:

- p_l 指向的结点是左结点, 则累加右结点的答案;
- p_r 指向的结点是右结点, 则累加左结点的答案.

至此, 我们就解决了使用 zkw 线段树进行单点修改及区间查询的问题, 不难发现其复杂度和普通线段树没有区别, 但是常数小很多.

区间修改及区间查询

问题在于, 尽管 zkw 线段树的常数相较于普通线段树小很多, 但是如果止步于单点修改区间查询, 那么树状数组有着更简洁的代码和更小的常数. 故而, 只有掌握了 zkw 线段树的区间修改和区间查询问题, 才算真正掌握了 zkw 线段树.

区间修改及区间查询

问题在于, 尽管 zkw 线段树的常数相较于普通线段树小很多, 但是如果止步于单点修改区间查询, 那么树状数组有着更简洁的代码和更小的常数. 故而, 只有掌握了 zkw 线段树的区间修改和区间查询问题, 才算真正掌握了 zkw 线段树.

依然以先前的模板为例 (区间加 区间求和), 由于我们舍弃了自上而下的递归结构, 故而懒标记的 push down 无法实现, 对于此类问题, 我们采用标记永久化.

区间修改

和区间查询类似, 我们在区间修改时, 如果指针 p_l 访问到的结点是左结点, 则我们修改右结点并打上标记, 这里的标记表示该结点的所有子树都要修改, p_r 同理.

但是并没有结束, 因为这个修改同样会对其祖先结点产生影响 (任何线段树上的修改都是对一条从根节点到叶结点的路径而言, 懒标记只解决了从当前结点到叶节点的问题).

故而我们需要动态记录 p_l 结点访问过程中所遍历到的区间长度, 并对所有访问到的结点实时计算修改带来的贡献.

区间修改下的区间查询

区间修改下的区间查询和单点修改下的区间查询唯一的区别就是要考虑到懒标记的影响.

即要实时计算该结点里包含的修改长度, 然后将懒标记的贡献计算进去. 其余的过程和单点修改时的区间查询是一样的.

区间修改下的区间查询

区间修改下的区间查询和单点修改下的区间查询唯一的区别就是要考虑到懒标记的影响.

即要实时计算该结点里包含的修改长度, 然后将懒标记的贡献计算进去. 其余的过程和单点修改时的区间查询是一样的.

至此, 我们就解决了用 zkw 线段树区间修改区间查询的问题, 同样, 比起普通线段树, 常数依然更小, 但是理解难度相对更大. 大家酌情选用.

Problems

P1438 无聊的数列

给定数列 a, 长度为 n, m 次操作, 分为如下两种:

1. 给定 l, r, k, d. 给 $a_l \sim a_r$ 加上一个首项为 k 公差为 d 的等差数列;
2. 给定 p, 查询 a_p.

Restrictions: $1 \leq n, m \leq 10^5$, $|a_i|, |k|, |d| \leq 200$.

P1438 无聊的数列

给定数列 a, 长度为 n, m 次操作, 分为如下两种:

1. 给定 l, r, k, d. 给 $a_l \sim a_r$ 加上一个首项为 k 公差为 d 的等差数列;
2. 给定 p, 查询 a_p.

Restrictions: $1 \le n, m \le 10^5$, $|a_i|, |k|, |d| \le 200$.

单点查询区间修改, 当区间修改不好做的时候就考虑差分. 令 $b_i = a_i - a_{i-1}$ (这里我们只对修改量建立差分数组).

P1438 无聊的数列

给定数列 a, 长度为 n, m 次操作, 分为如下两种:

1. 给定 l, r, k, d. 给 $a_l \sim a_r$ 加上一个首项为 k 公差为 d 的等差数列;
2. 给定 p, 查询 a_p.

Restrictions: $1 \le n, m \le 10^5$, $|a_i|, |k|, |d| \le 200$.

单点查询区间修改, 当区间修改不好做的时候就考虑差分. 令 $b_i = a_i - a_{i-1}$ (这里我们只对修改量建立差分数组).

考虑一次修改在差分数列上的影响, 实际上 b_l 要加上 k, $b_{l+1} \sim b_r$ 加上 d, 在 b_{r+1} 减去 $k + (r - l) \cdot d$ (即末项). 对于单点查询实际上是查询 $a_p + \sum_{i=1}^{x} b_i$.

转化为了单点修改, 区间修改, 区间查询问题. 线段树解决即可.

给定数列 a, 长度为 n, m 次操作, 分为如下两种:

1. 给定 l, r, k, d. 给 $a_l \sim a_r$ 加上一个首项为 k 公差为 d 的等差数列;
2. 给定 l, r, 查询 $\sum_{i=l}^{r} a_i$.

Restrictions: $1 \leq n, m \leq 10^5$, $|a_i|$, $|k|$, $|d| \leq 200$.

Advanced Problem

给定数列 a, 长度为 n, m 次操作, 分为如下两种:

1. 给定 l, r, k, d. 给 $a_l \sim a_r$ 加上一个首项为 k 公差为 d 的等差数列;
2. 给定 l, r, 查询 $\sum\limits_{i=l}^{r} a_i$.

Restrictions: $1 \le n, m \le 10^5$, $|a_i|, |k|, |d| \le 200$.

标记永久化, 考虑对每个结点维护一个二元组懒标记 (k, d), 表示一次修改. 则查询时计算标记的贡献即可. 实现时需要注意懒标记当且仅当区间与操作区间完全相等的时候才能够操作. 具体实现见作业上给的链接.

P5459 [BJOI2016] 回转寿司

给定数列 $\{a_n\}$, 以及两个整数 L, R. 查询有多少二元组 (l, r) 满足

$$L \leq \sum_{i=l}^{r} a_i \leq R$$

Restrictions: $1 \leq n \leq 10^5$, $|a_i| \leq 10^5$, $L, R \leq 10^9$.

P5459 [BJOI2016] 回转寿司

给定数列 $\{a_n\}$, 以及两个整数 L, R. 查询有多少二元组 (l, r) 满足

$$L \le \sum_{i=l}^{r} a_i \le R$$

Restrictions: $1 \le n \le 10^5$, $|a_i| \le 10^5$, $L, R \le 10^9$.

区间和转为前缀和, 令 $\mathrm{pre}_x = \sum_{i=1}^{x} a_i$. 则

$$\begin{aligned} L \le \sum_{i=l}^{r} \le R &\Longrightarrow L \le \mathrm{pre}_r - \mathrm{pre}_{l-1} \le R \\ &\Longrightarrow \mathrm{pre}_r - R \le \mathrm{pre}_{l-1} \le \mathrm{pre}_r - L \end{aligned}$$

故我们从 $r = 1, 2, \ldots, n$ 逐步处理, 每次查询有多少 pre_{l-1} 满足上述条件即可.

具体实现上使用动态开点权值线段树.

P4145 上帝造题的七分钟 2 / 花神游历各国

给定数列 $\{a_n\}$, m 次操作, 分为如下两种:

1. 给定 l, r, 对所有的 $l \le k \le r$, 将 a_k 赋值为 $\lfloor \sqrt{a_k} \rfloor$;
2. 给定 l, r, 查询 $\sum_{i=l}^{r} a_i$.

Restrictions: $1 \le n, m \le 10^5$, $0 \le a_i \le 10^{12}$.

P4145 上帝造题的七分钟 2 / 花神游历各国

给定数列 $\{a_n\}$, m 次操作, 分为如下两种:

1. 给定 l, r, 对所有的 $l \le k \le r$, 将 a_k 赋值为 $\lfloor \sqrt{a_k} \rfloor$;
2. 给定 l, r, 查询 $\sum_{i=l}^{r} a_i$.

Restrictions: $1 \le n, m \le 10^5$, $0 \le a_i \le 10^{12}$.

注意到一个数在被取根号若干次后就会变成 1, 而对 1 取根号是无效操作. 故而我们只需要记录区间的最大值, 对于最大值为 1 的区间略过, 否则递归暴力修改. 则暴力修改的次数不会超过 cn 次, 其中 c 为小常数. 故复杂度正确.

类似题目: CF438D The Child and Sequence

P3792 由乃与大母神原型和偶像崇拜

给定长度为 n 的数列 $\{a_n\}$, m 次操作, 分为如下两种:

1. 给定 x, y, 将 a_x 修改为 y;
2. 给定 l, r, 查询 $a_l \sim a_r$ 能否重排为值域上连续的一段 (例如 $4, 5, 3, 2$ 就可以重排为 $2, 3, 4, 5$, 而 $1, 2, 2, 3$ 就不行).

Restrictions: $1 \leq n, m \leq 5 \times 10^5$, $1 \leq a_i \leq 2.5 \times 10^7$.

P3792 由乃与大母神原型和偶像崇拜

给定长度为 n 的数列 $\{a_n\}$, m 次操作, 分为如下两种:

1. 给定 x, y, 将 a_x 修改为 y;
2. 给定 l, r, 查询 $a_l \sim a_r$ 能否重排为值域上连续的一段 (例如 $4, 5, 3, 2$ 就可以重排为 $2, 3, 4, 5$, 而 $1, 2, 2, 3$ 就不行).

Restrictions: $1 \le n, m \le 5 \times 10^5$, $1 \le a_i \le 2.5 \times 10^7$.

维护区间和, 区间 max, 区间 min 以及区间平方和, 计算一下如果是连续段的话, 根据 max, min 算一下区间和和区间平方和是多少, 与实际答案判断一下. 要取一个优秀的模数 (我取的是 998244353).

关于平方和的公式 (虽然你们应该都知道):

$$\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$$

另外, 此题实际上可以做区间推平修改, 但这个太超纲了, 或许等你们学习了 cdq 分治之后可以来挑战一下.

CF242E XOR on Segment

给定长度为 n 的数列 $\{a_n\}$, m 次操作, 分为如下两种:

1. 给定 l, r, x, 对所有的 $l \le k \le r$, 将 $a_k \oplus x$;
2. 给定 l, r, 查询 $\sum_{i=l}^{r} a_i$.

Restrictions: $1 \le n \le 10^5, 1 \le m \le 5 \times 10^4, 0 \le a_i, x \le 10^6$.

CF242E XOR on Segment

给定长度为 n 的数列 $\{a_n\}$, m 次操作, 分为如下两种:

1. 给定 l, r, x, 对所有的 $l \le k \le r$, 将 $a_k \oplus x$;
2. 给定 l, r, 查询 $\sum_{i=l}^{r} a_i$.

Restrictions: $1 \le n \le 10^5$, $1 \le m \le 5 \times 10^4$, $0 \le a_i, x \le 10^6$.

注意到位运算按位独立, 故考虑拆位, 对每一位建立一个线段树, 则每棵线段树转为区间翻转区间求和问题, 求解即可.

时间复杂度: $O(m \log a_i \log n)$.

P1972 [SDOI2009] HH的项链

给定长度为 n 的数列 $\{a_n\}$, m 次询问, 每次询问包含两个参数 l, r, 询问 $a_l \sim a_r$ 中有多少种不同的数.

Restrictions: $1 \le n, m, a_i \le 10^6$.

P1972 [SDOI2009] HH的项链

给定长度为 n 的数列 $\{a_n\}$, m 次询问, 每次询问包含两个参数 l, r, 询问 $a_l \sim a_r$ 中有多少种不同的数.

Restrictions: $1 \le n, m, a_i \le 10^6$.

十分经典的题, 有不少于三种解法. 我们这里考虑用树状数组/线段树/zkw 线段树解.

另外对于此题, 我们需要学习一种数据结构技巧: 离线处理询问. 具体来说, 在此题中, 如果询问区间右端点 r 固定, 那么一个颜色的贡献仅由其最右边的存在产生.

于是我们存储所有询问并按右端点排序, 记录每个颜色出现的最晚位置并给这个位置设为 1, 将这个颜色上一个位置出现的权值消去 (这需要我们记录每个颜色的前置位置). 对于一个询问 l, r, 我们只需要查询 l, r 的区间和即可. 这个操作可以通过很多方式实现.

P1972 [SDOI2009] HH的项链

给定长度为 n 的数列 $\{a_n\}$, m 次询问, 每次询问包含两个参数 l, r, 询问 $a_l \sim a_r$ 中有多少种不同的数.

Restrictions: $1 \le n, m, a_i \le 10^6$.

十分经典的题, 有不少于三种解法. 我们这里考虑用树状数组/线段树/zkw 线段树解.

另外对于此题, 我们需要学习一种数据结构技巧: 离线处理询问. 具体来说, 在此题中, 如果询问区间右端点 r 固定, 那么一个颜色的贡献仅由其最右边的存在产生.

于是我们存储所有询问并按右端点排序, 记录每个颜色出现的最晚位置并给这个位置设为 1, 将这个颜色上一个位置出现的权值消去 (这需要我们记录每个颜色的前置位置). 对于一个询问 l, r, 我们只需要查询 l, r 的区间和即可. 这个操作可以通过很多方式实现.

扩展: 区间推平区间查颜色: cdq 分治 + 树状数组; 强制在线: 可持久化线段树.

Luogu P5524 [Ynoi2012] NOIP2015 充满了希望

给定 n, m, q. 对一个初值为 0, 长度为 n 的序列 a 做 m 次操作, 分为如下 3 种:

1. 给定 x, y, 交换 a_x, a_y 的值;
2. 给定 l, r, k, 将 $a_l \sim a_r$ 赋值为 k;
3. 给定 x, 查询 a_x.

q 次询问, 每次询问给定 l, r, 查询按顺序做 $l \sim r$ 中的所有操作, 其中所有查询操作的和.

Restrictions: $1 \le n, m, q \le 10^6$.

给定 n, m, q. 对一个初值为 0, 长度为 n 的序列 a 做 m 次操作, 分为如下 3 种:

1. 给定 x, y, 交换 a_x, a_y 的值;
2. 给定 l, r, k, 将 $a_l \sim a_r$ 赋值为 k;
3. 给定 x, 查询 a_x.

q 次询问, 每次询问给定 l, r, 查询按顺序做 $l \sim r$ 中的所有操作, 其中所有查询操作的和.

Restrictions: $1 \le n, m, q \le 10^6$.

建立线段树, 维护 l, r 中最后一次推平操作的时间, 并维护时间 i 所对应的颜色.

对于交换操作, 无论之后是否有推平操作, 我们都可以直接交换而不会对结果有任何影响 (因为如果有推平操作, 对应的位置会直接推成一个颜色段).

单点查询操作直接查询点所对应的颜色段.

然后我们离线所有询问, 用树状数组维护询问前缀和, 查询询问区间即可.

时间复杂度: $O(n \log n)$.

结语

本次课程介绍了一些线段树的进阶技巧, 并介绍了线段树在实际 OI 问题中的应用, 希望大家下课之后好好做题. 之后的课程将很大一部分基于本节课讲的一些思想 (例如可持久化以及平衡树等), 线段树的两节课是数据结构基础中的基础, 务必掌握.

多谢大家.