

## 简介

基本的线段树可以视作一种建立在数组上的分治手段，解决的问题大多是线性的（也存在树形结构上的线段树方法，另说）。一般问题会涉及到序列上的多种类型完全不同、给定下标范围的操作（比如要求序列可以修改和查询），线段树可以将每种操作都做到  $O(\log_2 n)$  的时间实现。

本部分涉及以下内容：入门线段树（区间求和等基本问题）、权值线段树的应用、区间合并线段树、线段树动态开点、扫描线。

进阶内容如线段树合并、线段树分裂以及一些大牛的线段树优化会在之后更新。

可持久化线段树收录在可持久化数据结构中。

下面是两篇初学线段树时写的博客：

[线段树（简单实现高效区间操作）](#)

[线段树求解各种问题的模板（单点修改、区间修改、扫描线思想）](#)

## 线段树基础模板1

**原题链接：** [P3372 【模板】线段树 1](#)

已知一个长度为  $n$  的序列  $a$  和  $m$  个操作，操作有两种：

1. 给定  $x, y, k$ ，将下标范围  $[x, y]$  内的每个数都加  $k$ ；
2. 给定  $x, y$ ，输出下标范围  $[x, y]$  中数值的总和。

约定  $1 \leq n, m \leq 10^5$ ，操作 2 保证结果在  $\text{int64}$  内。

Time limit : 1000ms / Memory limit : 125mb

**思路：** 带懒标记的线段树。

## 线段树基础模板2

**原题链接：** [P3373 【模板】线段树 2](#)

已知模数  $p$ ，一个长度为  $n$  的序列  $a$  和  $m$  个操作，操作有三种：

1. 给定  $x, y, k$ ，将下标范围  $[x, y]$  内的每个数都加  $k$ ；
2. 给定  $x, y, k$ ，将下标范围  $[x, y]$  内的每个数都乘  $k$ ；

3. 给定  $x, y$  , 输出下标范围  $[x, y]$  中数值的总和对  $p$  取模的结果。

约定  $1 \leq n, m \leq 10^5$ 。

Time limit : 1000ms / Memory limit : 125mb

**思路：** 带懒标记的线段树，维护两个懒标记，分别表示加和乘，同时需要注意两种标记往下推的顺序。

## 权值线段树求逆序对个数

**原题链接：** [P1908 逆序对](#)

已知一个长度为  $n$  的序列  $a$  , 求序列中逆序对的个数。

(逆序对：满足  $a_i > a_j$  且  $i < j$  的有序对)

约定  $n \leq 5 \times 10^5$  , 序列中的任意数不超过  $10^9$ 。

Time limit : 1000ms / Memory limit : 125mb

**思路：** 对于下标  $i$  , 只需要知道下标小于  $i$  且比  $a_i$  大的元素个数，然后对每个下标求出的结果累计即可得到逆序对的个数。首先对原序列排个序，知道每个数字的排名，原序列中的数字用排名取代，构成新序列  $b$  , 同时记最大排名为  $b_{max}$  。然后，用线段树维护每个排名出现的次数，每次查询区间  $[b_i + 1, b_{max}]$  中数字的个数，累计到答案中。最后，更新一次线段树，将排名  $b_i$  出现的次数加 1。

## 权值线段树求三元组个数

**原题链接：** [P1637 三元上升子序列](#)

已知一个长度为  $n$  的序列  $a$  , 求序列中三元上升子序列的个数。

(三元上升子序列：满足  $i < j < k$  且  $a_i < a_j < a_k$  的有序对)

约定  $n \leq 5 \times 10^5$  , 序列中的任意数不超过  $10^9$ 。

Time limit : 1000ms / Memory limit : 125mb

**思路：** 与计算逆序对的方法基本相同，对于下标  $i$  , 查询一次下标比它小同时数值比它小的元素个数  $x_1$  , 再查询一次下标比它大同时数值比它大的元素个数  $x_2$  , 答案累计  $x_1 x_2$ 。

## 区间合并求最大连续子段和

原题链接: [GSS1 - Can you answer these queries I](#)

已知一个长度为  $n$  的序列  $a$  和  $m$  个操作, 操作有一种:

1. 给定  $x, y$ , 输出下标范围  $[x, y]$  的最大连续子段和。

约定  $1 \leq n \leq 5 \times 10^4$ , 序列中  $a$  每个元素满足  $-15007 \leq a_i \leq 15007$ 。

Time limit : 1000ms / Memory limit : 1536mb

**思路:** 考虑如何用区间  $[L, mid]$  和  $[mid + 1, R]$  的最大连续子段和表示区间  $[L, R]$  的最大连续子段和。显然可以分三种情况, 区间  $[L, R]$  的最大连续子段和在左子区间中、右子区间中或横跨左右两个区间。前两种情况只需要分别知道各个子区间的最大连续子段和即可, 与求  $[L, R]$  的问题一致。对于第三种情况, 考虑左区间从  $mid$  开始往左取元素累加的最大值, 右区间从  $mid + 1$  开始往右取元素累加的最大值, 两者相加即为横跨  $[L, R]$  两个子区间的最大连续子段和。如此一来, 原问题转化成如何维护一个区间分别从左往右、从右往左累加的最大值, 分别记为  $lsum$ 、 $rsum$ 。还是从  $[L, R]$  和它的两个子区间开始考虑, 区间  $[L, R]$  的  $lsum$  有两种情况, 一种是等于左子区间的  $lsum$ , 另一种是左子区间的所有元素和加上右子区间的  $lsum$ , 区间  $[L, R]$  的  $rsum$  计算方法与之类似。

整理上面的思路, 总结出线段树需要维护共 4 个值:

1.  $sum$ , 区间内所有元素的和;
2.  $lsum$ , 区间元素从左开始累加的最大值;
3.  $rsum$ , 区间元素从右开始累加的最大值;
4.  $msum$ , 区间内的最大连续子段和。

如何利用左右子区间更新  $[L, R]$  的 4 个值:

(设树上代表区间  $[L, R]$  的节点为  $p$ , 左右子区间分别为  $lc$ 、 $rc$ )

$$\begin{aligned} p.sum &= lc.sum + rc.sum \\ p.lsum &= \max(lc.lsum, lc.sum + rc.lsum) \\ p.rsum &= \max(rc.rsum, rc.sum + lc.rsum) \\ p.msum &= \max(\max(lc.msum, rc.msum), lc.rsum + rc.lsum) \end{aligned}$$

## 线段树维护后缀和

原题链接: [GSS2 - Can you answer these queries II](#)

已知一个长度为  $n$  的序列  $a$  和  $m$  个操作, 操作有一种:

1. 给定  $x, y$ , 输出下标范围  $[x, y]$  **去掉重复元素后 (指保留最左边的)** 的最大连续子段和。

约定  $1 \leq n, m \leq 10^5$ , 序列中  $a$  每个元素满足  $-10^5 \leq a_i \leq 10^5$ 。

Time limit : 1000ms / Memory limit : 1536mb

**思路：** 当一个问题与先前解决过的问题类似时，可以大胆地往类似的方法上去探索。但是显然，在这里区间合并的方法是难以实施的，考虑其他线段树的构造方法。回归问题的本质，从最大连续子段和的定义入手。设

$$sum(i, j) = \sum_{k=i}^j a_k, \quad 1 \leq i \leq j \leq n$$

定义  $msum(i, j)$  表示从  $i$  开始累加的最大值，用公式表示为

$$msum(i, j) = \max(sum(i, j), sum(i+1, j), \dots, sum(j, j)), \quad 1 \leq i \leq j \leq n$$

那么求解区间  $[L, R]$  上的答案  $ans$  可以表示为

$$ans = \max(msum(i, R), msum(i+1, R), \dots, msum(R, R)), \quad L \leq i \leq R$$

此时，缩小  $sum(i, j)$  的定义，用  $sum(i)$  来表示从下标  $i$  累加到下标  $R$  的所有元素的和。那么  $msum(i, R)$  相当于求区间  $[i, R]$  上  $sum(i)$  的最大值，显然这是可以用线段树维护的。而所要求的答案就变成了求区间  $[L, R]$  上， $msum(i, R)$  的历史最大值。考虑到  $sum(i)$  的定义，如果在求解  $[L, R]$  询问之前就已经将  $a_{R+1}$  更新到线段树中，则破坏了答案的构造。因此，需要对所有询问根据右端点的位置重新排序，且保持线段树的更新和询问同步。

这样解区间最大连续子段和可以化解元素去重的问题。假如元素  $a_R$  恰好和元素  $a_k$  重复，则对于  $sum(1)$  到  $sum(k)$ ，它们已经加过数值和  $a_R$  相等的元素，因此只需用  $a_R$  更新区间  $[k+1, R]$  上的和。

## 区间合并求最大连续子段和带单点修改

**原题链接：** [GSS3 - Can you answer these queries III](#)

已知一个长度为  $n$  的序列  $a$  和  $m$  个操作，操作有两种：

1. 给定  $0, x, y$ ，将下标为  $x$  元素的值修改为  $y$ ；
2. 给定  $1, x, y$ ，输出下标范围  $[x, y]$  的最大连续子段和。

约定  $1 \leq n, m \leq 5 \times 10^4$ ，序列中  $a$  每个元素满足  $-10^4 \leq a_i \leq 10^4$ 。

Time limit : 1000ms / Memory limit : 1536mb

**思路：** 带单点修改的区间合并。

## 线段树区间开根求和

**原题链接：** [GSS4 - Can you answer these queries IV](#)

已知一个长度为  $n$  的序列  $a$  和  $m$  个操作，操作有两种：

1. 给定  $0, x, y$ ，下标范围  $[x, y]$  的所有元素分别开平方根，且向下取整；
2. 给定  $1, x, y$ ，输出下标范围  $[x, y]$  中数值的总和。

约定  $1 \leq n, m \leq 10^5$ ，序列中  $a$  每个元素都为正数，且  $\sum_{i=1}^n a_i < 10^{18}$ 。

Time limit : 1000ms / Memory limit : 1536mb

**思路：**按常规思路来想，对一个区间内的每个数字开根号无法做到同时进行。如果用一个懒标记维护区间开根的次数而不是做多次单点修改，则难以统计区间所有元素的和。从开平方根运算的性质考虑，一个  $10^{18}$  的数字经过 6 次开平方根并向下取整就变成了 1，1 无论如何开平方根也保持不变。因此，序列  $a$  中的每个元素最多进行 6 次开平方根操作，而当我们知道当前操作区间的总和正好等于其长度时，这个区间就没有向下更新的必要。因此，该题无需懒标记，对于总和不等于长度的区间向下更新，对于总和等于长度的区间直接忽视。由于每个元素仅需极少开根次数即可变成 1，因此可以轻易推出，这种更新方法的常数极小。

## 线段树求区间端点不固定的最大连续子段和

**原题链接：**[GSS5 - Can you answer these queries V](#)

已知一个长度为  $n$  的序列  $a$  和  $m$  个操作，操作有一种：

1. 给定  $x_1, y_1, x_2, y_2$ ，设区间左端点  $L$  的范围是  $[x_1, y_1]$ ，区间右端点  $R$  的范围是  $[x_2, y_2]$  输出下标范围  $[L, R]$  的最大连续子段和（保证  $x_1 \leq x_2, y_1 \leq y_2$ ）。

约定  $1 \leq n, m \leq 10^4$ ，序列中  $a$  每个元素满足  $-10^4 \leq a_i \leq 10^4$ 。

Time limit : 1000ms / Memory limit : 1536mb

**思路：**根据  $x_1, y_1, x_2, y_2$  有两种排布方式： $x_1 \leq y_1 \leq x_2 \leq y_2$  或  $x_1 \leq x_2 \leq y_1 \leq y_2$ 。

对于第一种，显然最大连续子段和由三部分组成，区间  $[x_1, y_1]$  从右端点往左累加的最大值、区间  $(y_1, x_2)$  中所有元素的和、区间  $[x_2, y_2]$  从左端点往右累加的最大值，三者加起来即为答案。

对于第二种排布方式，答案所在的区间  $[l, r]$  分三种情况：

1.  $x_2 \leq l \leq r \leq y_1$ ，只需求区间  $[x_2, y_1]$  的最大连续子段和。
2.  $x_1 \leq l \leq x_2$  且  $x_2 \leq r \leq y_2$ ，分别求区间  $[x_1, x_2]$  从右端点往左累加的最大值、区间  $[x_2, y_2]$  从左端点往右累加的最大值，两项相加并减去公共的  $a_{x_2}$ （该点被重复计算了一次）
3.  $x_1 \leq l \leq y_1$  且  $y_1 \leq r \leq y_2$ ，分别求区间  $[x_1, y_1]$  从右端点往左累加的最大值、区间  $[y_1, y_2]$  从左端点往右累加的最大值，两项相加并减去公共的  $a_{y_1}$ （该点被重复计算了一次）

再取三种情况下的最大值即为第二种排布方式的答案。

## 动态开点1

原题链接: [Physical Education Lessons](#)

已知一个长度为  $n$  的序列  $a$  和  $m$  个操作, 操作有两种:

1. 给定  $1, x, y$ , 将下标范围  $[x, y]$  内的所有数变成 1;
2. 给定  $2, x, y$ , 将下标范围  $[x, y]$  内的所有数变成 0。

每一次操作完成后, 输出序列中 0 的个数。

约定  $1 \leq n \leq 10^9$ ,  $1 \leq m \leq 3 \times 10^5$

Time limit : 1000ms / Memory limit : 256mb

**思路:** 本题中序列的长度达到  $10^9$ , 使用一般的线段树无法开辟如此大的内存。但是, 注意到一开始所有元素的值都是 0, 这就提示我们有些区间只需更新或查询的时候再去分配内存给它即可, 而懒标记恰好可以推迟子区间的更新。因此, 更新时, 每到一个节点先检查该节点是否存在, 如果不存在即给它安排一个编号, 并将编号赋值给它父节点对应的孩子。

## 动态开点2

原题链接: [Periodic RMQ Problem](#)

已知一个序列  $a$  和  $m$  个操作, 操作有两种:

1. 给定  $l, r, x$ , 将下标范围  $[l, r]$  内的所有数变成  $x$ ;
2. 给定  $l, r$ , 输出下标范围  $[l, r]$  中所有数的最小值。

已知一个长度为  $n$  的序列  $b$ , 序列  $a$  的长度是  $b$  的  $k$  倍, 且序列  $a$  一开始满足任意的  $1 \leq i \leq n$ ,  $0 \leq j < k$ , 有  $a_{i+j \times n} = b_i$ 。

约定  $1 \leq n, m \leq 10^5$ ,  $1 \leq k \leq 10^4$ ,  $1 \leq b_i, x \leq 10^9$ 。

Time limit : 4000ms / Memory limit : 512mb

**思路:** 序列  $a$  最大长度可达  $10^9$ , 依然采用动态开点的方法。对于每个新开的点, 需要计算它的初始值, 即找到对应原序列且求出对应原序列的最小值。分两种情况: 区间长度大于等于  $n$ , 显然最小值即为序列  $b$  的最小值; 区间长度小于  $n$ ,  $l$  和  $r$  需要分别先对  $n$  取模, 判断出对应于序列  $b$  的哪段区间, 然后求出相应区间的最小值即可。其它操作与一般的线段树基本没有区别。

## 线段树扫描线求多个矩形的面积和

原题链接: P5490 【模板】扫描线

已知  $n$  个矩形左下角和右上角的直角坐标, 分别为  $(x_1, y_1)$ 、 $(x_2, y_2)$ , 求所有这些矩形组成图案的面积和。

约定  $1 \leq n \leq 10^5$ ,  $0 \leq x_1 < x_2 \leq 10^9$ ,  $0 \leq y_1 < y_2 \leq 10^9$ 。

Time limit : 1000ms / Memory limit : 125mb

**思路:** 想象众多矩形组成的图案被平行于  $x$  轴的直线划分成多个区块, 每个区块的面积等于相邻两个水平线的距离乘上图案与水平线重合的总长度。利用这个思想, 将每个矩形分解成两条水平线, 线段树维护相邻水平线之间图案和水平线相交的总长度。将所有的矩形的下边界标记为1, 上边界标记为-1, 从下往上计算。如果遇到某个矩形的下边界,  $(x_{1i}, x_{2i}]$  范围内所有区间段加 1 并得到如此操作后标记为正数的区间总长度, 乘上当前水平线与上方一条水平线的高度差, 计算得该区块的面积; 如果遇到某个矩形的上边界, 则  $(x_{1i}, x_{2i}]$  范围内所有区间段加 -1, 余下操作前一种相同。累计每次求出的面积, 即为图案的总面积。

## 线段树扫描线求多个矩形的周长并

原题链接: P1856 [USACO5.5]矩形周长Picture

已知  $n$  个矩形左下角和右上角的直角坐标, 分别为  $(x_1, y_1)$ 、 $(x_2, y_2)$ , 求所有这些矩形组成图案的周长和。

约定  $1 \leq n \leq 5 \times 10^3$ ,  $-10^4 \leq x_1 < x_2 \leq 10^4$ ,  $-10^4 \leq y_1 < y_2 \leq 10^4$ 。

Time limit : 1000ms / Memory limit : 125mb

**思路:** 方法与上题类似, 分别考虑水平方向和竖直方向上长度的计算。

水平方向: 水平方向上, 利用上一题的方法, 我们可以得到每根水平线的有效长度。相邻水平线的有效长度相减即为周长在水平方向上的部分。

竖直方向: 竖直方向上需要考虑相邻两个水平线之间的图案有几段。如果是连续的仅有一段, 即周长加上两倍高度差即可。如果存在多段, 则需要另外考虑。此处, 可以用线段树多维护三个值, 分别是当前查询区间左端点是否包含在有效区间内  $lflag$ 、右端点是否包含在有效区间内  $rflag$ 、当前查询区间包含的不连续有效区间段数  $num$ 。显然这三个值的维护可以用区间合并的思想维护。(设树上代表区间  $[L, R]$  的节点为  $p$ , 左右子区间分别为  $lc$ 、 $rc$ )

$$p.lc = lc.lflag$$

$$p.rc = rc.rflag$$

对于  $num$  的维护，需要考虑左右两个子区间合并后是否会合并掉端点处的有效区间。显然当且仅当左子区间的右端点包含在有效区间内且右子区间的左端点包含在有效区间内时，才会合并掉。因此  $num$  的更新如下：

$$p.num = lc.num + rc.num - \begin{cases} 1, & lc.rflag = 1 \wedge rc.lflag = 1 \\ 0, & lc.rflag = 1 \wedge rc.lflag = 0 \end{cases}$$