

线段树专题5-开点线段树、区间最值和历史最值

前置知识

讲解**110** - 线段树原理和代码详解 讲解**111** - 线段树特别修改操作的势能分析，题目**3**、题目**4**

线段树专题讲述顺序

专题**1**: 线段树原理和代码详解，讲解**110**

专题**2**: 线段树的离散化、二分搜索、特别修改，讲解**111**

专题**3**: 线段树维护更多类型的信息，讲解**112**

专题**4**: 线段树解决区间合并的问题，讲解**113**

专题**5**: 开点线段树、区间最值和历史最值，讲解**114**，本节

专题**6**: 线段树与扫描线结合的题目，讲解**115**

线段树与动态规划结合的内容，后续【扩展】标签下的课程里继续安排

树套树、可持久化线段树、树链剖分等内容，后续【挺难】标签下的课程里会安排

这个系列一定是全网有关线段树最好的教学视频，觉得好帮忙推荐给身边的人！

线段树专题5-开点线段树、区间最值和历史最值

开点线段树

可以支持很大的范围，一开始不为每个范围都分配空间

当真的需要开辟左侧、右侧的空间时，再临时申请

父范围的空间编号 i ，利用 cnt 自增给左右两侧申请的空间，记录在 $left[i]$ 和 $right[i]$ 里

除此之外和线段树再无区别

课上会重点图解

开点线段树适用的范围：需要支持的范围很大，但实际操作数量并不大

此时用开点线段树试试，俗称**莽一把**，使用空间约等于：**操作数量 * 树高 * 2**，适量调大即可

课上会重点图解

注意：测试通过的标准稍严，开点线段树很容易被卡，不过只有比赛才可能会卡，笔试一般不会卡

线段树专题5-开点线段树、区间最值和历史最值

开点线段树的劣势：常数时间不好，相比其他方法还是比较浪费空间

开点线段树的优势：分析难度低，往往理解了开点线段树就很容易想到并轻易使用

大多数情况下都可以被其他方法替代，比如离散化、有序表结构(平衡树)、堆结构等等
如果数据量允许，莽一把能节省很多思考的时间，但是往往不是最优解，只是时间复杂度不差而已

注意：

开点线段树不可被替代的用法是用于 **线段树的合并与分裂**

但是这一话题和**树链剖分**结合比较紧密，会在【**挺难**】标签下的视频里安排

有序表(平衡树)的实现会在后续【**扩展**】、【**挺难**】标签下的课程里安排，这是非常大的话题

线段树专题5-开点线段树、区间最值和历史最值

题目1

动态开点线段树

一共有 n 个位置，编号从 $1 \sim n$ ，一开始所有位置的值为0

实现如下两个操作，一共会调用 m 次

操作 1 $l\ r\ v$: 把 $l \sim r$ 范围的每个数增加 v

操作 2 $l\ r$: 返回 $l \sim r$ 范围的累加和

$1 \leq n \leq 10^9$

$1 \leq m \leq 10^3$

测试链接：<https://www.luogu.com.cn/problem/P2781>

看过代码就学会了开点线段树，相当简单

核心就是，除非不得不去申请空间的时候，才会申请

线段树专题5-开点线段树、区间最值和历史最值

题目2

统计区间中的整数数目

实现*CountIntervals*类的如下三个方法

1) *CountIntervals()* : 初始化

2) *void add(int l, int r)* : 把 $[l, r]$ 范围上的数字都设置成1

3) *int count()* : 返回整个区间有多少个1

*CountIntervals*类需要支持 $1 \sim 10^9$ 范围

调用*add*和*count*方法总共 10^5 次

测试链接 : <https://leetcode.cn/problems/count-integers-in-intervals/>

本题特殊之处在于，无需建立懒更新机制

线段树专题5-开点线段树、区间最值和历史最值

题目3

线段树的区间最值操作

给定一个长度为 n 的数组 arr ，实现支持以下三种操作的结构

操作 $0\ l\ r\ x$: 把 $arr[l..r]$ 范围的每个数 v ，更新成 $\min(v, x)$

操作 $1\ l\ r$: 查询 $arr[l..r]$ 范围上的最大值

操作 $2\ l\ r$: 查询 $arr[l..r]$ 范围上的累加和

学前建议：先理解讲解111 - 线段树的特别修改，这一节的题目3、题目4，势能分析入门题目

核心剪枝策略：最大值、最大值个数、严格次大值的配合，课上重点图解

时间复杂度分析：标签回收模型 + 势能分析，课上重点图解

发明者吉如一，也叫吉如一线段树，高中时的国家集训队论文，代码中附上了原论文，有需要可以查阅

线段树专题5-开点线段树、区间最值和历史最值

题目4

线段树范围增加操作 + 区间最值操作

给定一个长度为 n 的数组 arr ，实现支持以下四种操作的结构

操作 $0\ l\ r\ x$: 把 $arr[l..r]$ 范围的每个数 v ，增加 x

操作 $1\ l\ r\ x$: 把 $arr[l..r]$ 范围的每个数 v ，更新成 $\min(v, x)$

操作 $2\ l\ r$: 查询 $arr[l..r]$ 范围上的最大值

操作 $3\ l\ r$: 查询 $arr[l..r]$ 范围上的累加和

每个数字都增加的范围修改操作发生后， $setMin$ 暴力执行的势能增加多少？

增加幅度不超过 $(\log n)$ 平方规模，这个量很小，课上重点图解

$setMin$ 的懒更新只会削弱最大值，需要实现 $setMinLazy$ 函数

add 的懒更新会增加所有数字的值，需要实现 $addLazy$ 函数

代码层次的小优化：可以只实现一个 $lazy$ 函数，设计最大值增加幅度、其他值增加幅度两个入参即可

线段树专题5-开点线段树、区间最值和历史最值

题目5

区间最值和历史最值

给定两个长度都为 n 的数组 A 和 B ，一开始两个数组完全一样

任何操作做完，都更新 B 数组， $B[i] = \max(B[i], A[i])$

实现以下五种操作，一共会调用 m 次

操作 1 $l\ r\ v$: $A[l..r]$ 范围上每个数加上 v

操作 2 $l\ r\ v$: $A[l..r]$ 范围上每个数 $A[i]$ 变成 $\min(A[i], v)$

操作 3 $l\ r$: 返回 $A[l..r]$ 范围上的累加和

操作 4 $l\ r$: 返回 $A[l..r]$ 范围上的最大值

操作 5 $l\ r$: 返回 $B[l..r]$ 范围上的最大值

$1 \leq n, m \leq 5 * 10^5$

测试链接：<https://www.luogu.com.cn/problem/P6242>