A - 一棵简单的线段树
B - 一棵普通的线段树
C - 一棵像样的线段树
D - 一棵复杂的线段树

2018 数据结构专题训练 - A,B,C,D

傅宣登

2018 年 5 月 11 日

A - 一棵简单的线段树
题意
思路
核心代码
B - 一棵普通的线段树
C - 一棵像样的线段树
D - 一棵复杂的线段树

A - 一棵简单的线段树

题面

人生已如此艰难，让我们活得轻松一点.
给你一个数组 $A[1..n]$，初始时每个元素都为零.
我会请你帮我对数组完成一些操作.
第一种可能，我会给你两个数 p 和 x $(1 \leq p \leq n)$，请你帮我把数组的第 p 个元素替换为 x，即 $A[p] \leftarrow x$.
第二种可能，我会给你两个数 L 和 R $(1 \leq L < R \leq n)$，请你告诉我 $A[L], A[L+1], \ldots, A[R]$ 这几个数中去掉一个最大值和一个最小值后剩下的数的和是多少.
好了，现在锅都丢给你了，我可以活得轻松一点了.

A - 一棵简单的线段树
题意
思路
核心代码
B - 一棵普通的线段树
C - 一棵像样的线段树
D - 一棵复杂的线段树

A - 一棵简单的线段树

题意

一个数组 $A[1..n]$，初始全为 0，m 个操作
涉及的操作有

- 单点修改 $A[p] = x$
- 区间查询 $sum([l, r]), max([l, r]), min([l, r])$

数据范围

- $2 \leq n \leq 10^6$，$1 \leq m \leq 10^6$
- 修改的数的绝对值在 10^9 的范围内
- 时限 2 秒.

A - 一棵简单的线段树
题意
思路
核心代码
B - 一棵普通的线段树
C - 一棵像样的线段树
D - 一棵复杂的线段树

问题分析

- 时限 2 秒，直接暴力会超时
- 数组中每个数的绝对值不超过 10^9，10^6 个数其和不会超过 10^{15}，可以用 `long long` 存下

解决方案

单点修改、区间查询的线段树可以在 $O(\log n)$ 的时间内维护区间的和、最大值、最小值.
此题用三个线段树维护各个区间的和、最大值、最小值，对每个查询输出对应区间的 $sum - max - min$ 即可.

时间复杂度

$O(m \log n)$

A - 一棵简单的线段树

单点修改

```
void update(int p, ll x, int o = 1, int l = 1, in
{
    if(r <= p && p <= l)
    {
        sum[o] = ma[o] = mi[o] = x;
    } else {
        int mid = l+r>>1;
        if(p <= mid) update(p, x, o<<1, l, mid);
        else update(p, x, o<<1|1, mid+1, r);
        maintain(o);
    }
}
```

B - 一棵普通的线段树

A - 一棵简单的线段树
B - 一棵普通的线段树
题意
思路
C - 一棵像样的线段树
D - 一棵复杂的线段树

题意

一个数组 $A[1..n]$，初始值全为 0，m 个操作：

- 使区间 $[L, R]$ 内每个数都加上 v
- 查询区间 $[L, R]$ 内所有数的和

数据范围

- $1 \le n \le 10^6$
- $1 \le m \le 10^6$
- $|v| \le 10^3$
- 时限 4 秒

B - 一棵普通的线段树

A - 一棵简单的线段树
B - 一棵普通的线段树
题意
思路
C - 一棵像样的线段树
D - 一棵复杂的线段树

问题分析

- 时限 4 秒，直接暴力涉及最坏大概 10^{12} 次运算，会 TLE
- v 的绝对值不超过 10^3，每个数不会超过 $10^3 \cdot 10^6 = 10^9$，和不会超过 10^{15}，long long 可以存下答案

解决方案

区间修改（带lazy 标记）、区间查询的线段树可以在 $O(\log n)$ 的时间内维护区间的和.
此题写一个裸的区间修改、区间查询的线段树即可.

时间复杂度

$O(m \log n)$

A - 一棵简单的线段树
B - 一棵普通的线段树
C - 一棵像样的线段树
题意
思路
D - 一棵复杂的线段树

题意

给定 n，$c_1, c_2, \dots, c_n$ 和 $b_0 = 1$，按下述公式计算并输出 $b_1, b_2, \dots, b_n$:

$$b_i = \operatorname*{xem}_{i-c_i \le j < i} \{b_j\}, \quad i = 1, 2, \dots, n$$

数据范围

- $1 \le n \le 10^6$
- $1 \le c_i \le i$
- 时限 1 秒

A - 一棵简单的线段树
B - 一棵普通的线段树
C - 一棵像样的线段树
题意
思路
D - 一棵复杂的线段树

问题分析

- 每个 b_i 的值即是它前面 c_i 个 $b_k(i-c_i \le k \le i-1)$ 中最小的没有出现的正整数
- 题目保证对每个 c_i 有 $1 \le c_i \le i$，也即是不会涉及 $b_k(k<0)$ 的值
- 如果每个 $c_i = i$，此时 b_n 有最大值 $n+1$. 所有的 b_i 都不会超过 $n+1$
- 如果暴力计算，最坏需要大概 10^{12} 次计算（TLE），不可

解决方案

设 $last[x]$ 表示使 $b_i = x$ 的最大的 i，即 b_i 中 x 最后出现的位置.

C - 一棵像样的线段树

解决方案

这样问题就转换为 $b_i = \min\limits_{last[x] < i - c_i} \{x\}$，也就是满足 $last[x] < i - c_i$ 的最小的 x，其中 $1 \leq x \leq n+1, x \in \mathbb{N}^*$.
可以用线段树维护 $last[1..(n+1)]$ 每个区间的最小值.
查询的时候给定一个值 $v = i - c_i$，从最大的区间往下查找，如果左子区间的最小值大于等于 v，则递归查找右子区间；否则递归查找左子区间. 当区间长度为 1 时，此区间的左 (右) 端点就是答案，返回之.
得到 b_i 后，更新 $last[b_i] = i$，再进行后面的计算.
注意初始时 $last[1] = 0$，其他 $last$ 值全为小于 0 的值.

时间复杂度

$O(n \log n)$

D - 一棵复杂的线段树

A - 一棵简单的线段树
B - 一棵普通的线段树
C - 一棵像样的线段树
D - 一棵复杂的线段树
题意
思路

题意

给定一个数组 $A[1..n]$，对数组施以 m 次排序操作，每次排序针对不同的区间，升序或降序.
最后输出 $A[k]$ 的值.

数据范围

- $1 \leq n \leq 10^5$，$1 \leq k \leq n$
- $a_1, a_2, ..., a_n$ 是 $1 \sim n$ 的一个排列
- $1 \leq m \leq 10^5$
- 时限 4 秒

D - 一棵复杂的线段树

问题分析

- $a_1, a_2, \dots, a_n$ 互不相同且取遍 $1 \sim n$ 中的每个值
- 考虑二分答案

解决方案

二分答案，设 mid 为当前考虑答案区间的中点.
构造数组 $B[1..n]$，$b_i = \mathrm{sgn}(a_i - mid)$. 数组 B 中只有 0 或 1.
用线段树维护数组 B 每个区间 1 的个数，可以实现在 $O(\log n)$ 的时间内对区间进行排序.
懒惰标记用来表示该区间全部赋成 0 或是 1.

D - 一棵复杂的线段树

A - 一棵简单的线段树
B - 一棵普通的线段树
C - 一棵像样的线段树
D - 一棵复杂的线段树
题意
思路

解决方案

对区间 $[L, R]$ 排序:

1. 查询出区间 $[L, R]$ 有多少个 1，设为 c
2. 将区间 $[L, R]$ 全赋为 0
3. 从小到大排序 区间 $[R-c+1, R]$ 赋 1
   从大到小排序 区间 $[L, L+c-1]$ 赋 1

如果考虑 mid 时，最终排序后 $A[k]=1$(即区间 $[k,k]$ 里 1 的个数为 1)，则表示答案大于等于 mid，此时答案所在区间更新为 $[mid, r]$. 如果 $A[k]=0$，表示答案小于 mid，此时答案所在区间更新为 $[l, mid-1]$.
当 $l=r$ 时得到答案，输出即可.

时间复杂度

$O(m\log^2 n)$