暑期集训 2022

主席树

主讲人: 刘焕宇

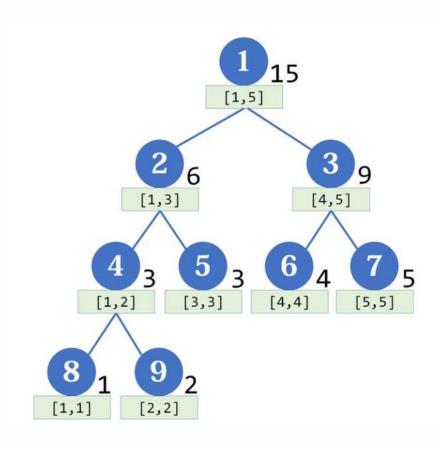
个人ID: gtxygyzb

(工藤新一宫野志保)



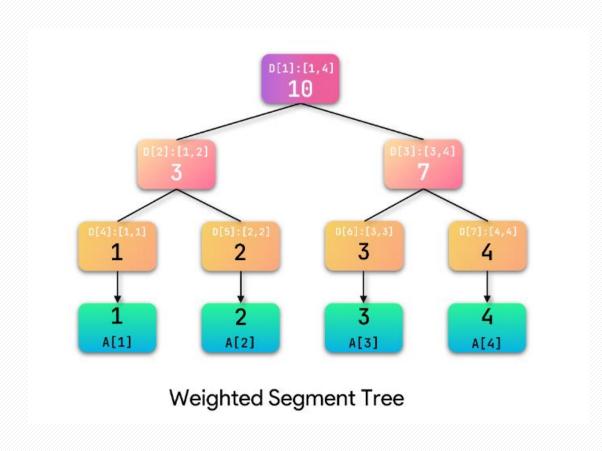
前置知识复习主席树算法基础题进阶题





- 区间[l,r]转化为对线段树上log(n)个区间的修改/询问,维护信息
- 运用懒人标记等方式,对整个大区间的信息下次访问到再往小区间更新
- n个叶子节点, n-1个非叶子节点, 并且防止数组访问越界, 总共有:

四倍空间!!!!!!!!!!!!!



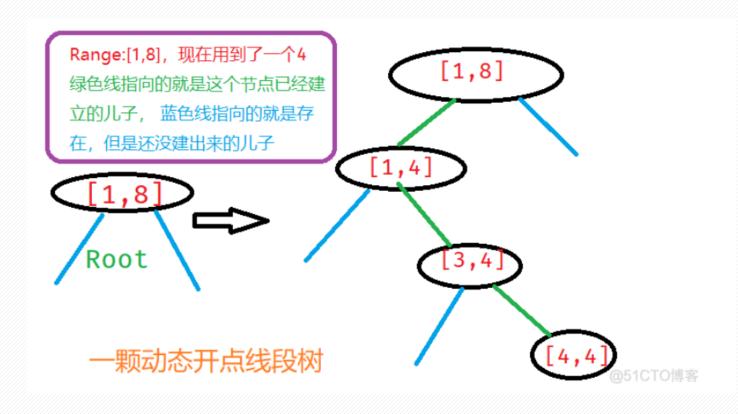
权值线段树一般用于维护一段区间的数出现的次数

给定序列{ 1, 2, 2, 3, 3, 3, 4, 4, 4, 4} 可以建出左图所示的权值线段树

可以求整体第k大

采用元素到下标数组映射的方式进行插入。 当空间不足时,常采用离散化的方式进行解 决。

权值线段树



- 链式储存法,即对一个节点 建立左右儿子指针,指向它的 左右儿子。
- 左右儿子从普通线段树的 $u \ll 1, u \ll 1 \mid 1$ 变为 lc[u], rc[u]
- 利用一个类似线段树懒标记的思想,如果一个节点需要使用,那么我们才建立这个节点, 反之指向空节点代表当前没有信息。

动态开点

为什么需要动态开点线段树?

信息的继承:建立新节点时,若想使用历史信息,直接把孩子连向已经建立的节点即可

于数据结构而言,**可持久化**总是可以保留每一个历史版本,并且支持操作的不可变特性。 而可持久化线段树(主席树)的本质就是多根线段树,每个<mark>根</mark>都代表一个历史版本。

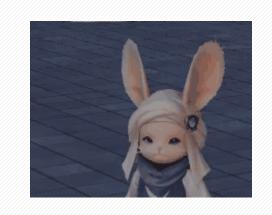
为什么是每个根而不是每棵树呢?

假如我们每次进行更新操作时都要先拷贝一份上次的版本用来回溯,再对原本的线段树进行修改,这样的话无论时间复杂度还是空间复杂度都是难以接受的。

可持久化

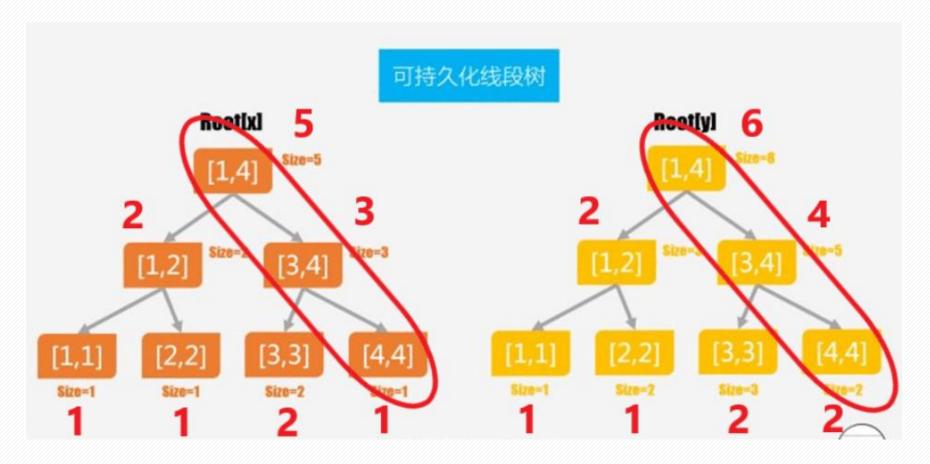
主席树又叫做可持久化(权值)线段树,一开始使用来解决区间第k大问题

据说因其发明者黄嘉泰的名字首字母hjt而得名主 席树

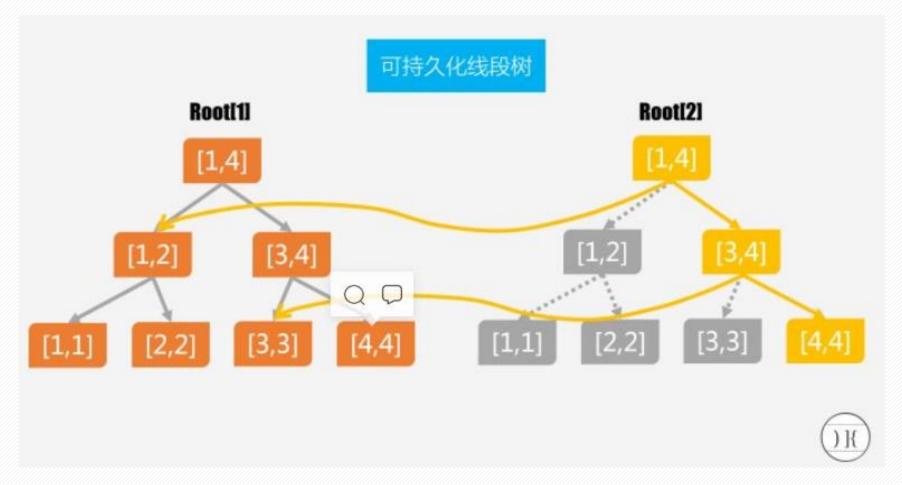


(名字短一点更方便,故后文都称为主席树)

主席树



当我们在修改关键值时,只有相关区间的节点内的值会受到影响,而 树本身结构并未发生变化,那么在每次修改时,不再是在原来的版本 上下功夫,而是创建一系列新的节点,从而达到可持久化的目的。



所以我们只需要多记录一条链就能代表历史的版本了

给定 n 个整数构成的序列 a_i , q组询问将对于指定的闭区间 [1,r] 查询其区间内的第 k 小值。

按顺序*i*从1到*n*建立可持久化权值线段树 每次查询第*r*个根*rt*[*r*],其余操作与普通权值线段树相 同: *k*比左子树权值个数大,则减去左子树大小后往右 子树走,否则往左子树走。

在权值离散化下,时空复杂度均为0 (nlogn)

例题一

```
void add(int &u, int pre, int l, int r, int pos)
{
    u = ++top; //新建结点
    lc[u] = lc[pre]; rc[u] = rc[pre]; //左右节点继承pre
    val[u] = val[pre] + 1; //权值加一
    if (l == r) return; //最后一层返回
    int mid = l + r >> 1;
    if (pos <= mid) add(lc[u], lc[pre], l, mid, pos); //往左
    else add(rc[u], rc[pre], mid + 1, r, pos); //往右
}
</pre>
```

Luogu3834 主席树模板题: 区间第k大 给定 n 个整数构成的序列 a_i , q组询问将对于指定的闭区间 [l,r] 查询其区间内的第 k 小值。

类似前缀和的方法,访问主席树时,同时从rt[r],rt[l-1]两个根一起向下遍历。设当前节点分别为u,pre,则该节点所代表的权值区间的权值数量即为val[u]-val[pre],其余同例题一。

在权值离散化下,时空复杂度均为O(nlogn)

例题二

```
int query(int u, int pre, int l, int r, int k)
{
    if (l == r) return l;
    int sum = val[lc[u]] - val[lc[pre]]; // 左节点权值数量
    int mid = l + r >> 1;
    if (k <= sum) return query(lc[u], lc[pre], l, mid, k); //往左递归
    else return query(rc[u], rc[pre], mid + 1, r, k - sum); //減掉左节点往右递归
}</pre>
```

Luogu 3919 可持久化数组模板

你需要维护这样的一个长度为 N 的数组,支持如下两种操作:

- 1.在某个历史版本上修改某一个位置上的值 2.访问某个历史版本上的某一位置的值 每一次操作1/2都记为一个历史版本,即使询 问没有改变数组
- 1.对于操作1,格式为 v_i 1 loc_i val_i ,即为在版本 v_i 的基础上,将 a_{loc_i} 修改为 val_i 2.对于操作2,格式为 v_i 2 loc_i ,即访问版本 v_i 中 a_{loc_i} 的值

输入#1

```
5 10

59 46 14 87 41

0 2 1

0 1 1 14

0 1 1 57

0 1 1 88

4 2 4

0 2 5

0 2 4

4 2 1

2 2 2

1 1 5 91
```

输出#1

例题三

Luogu 3919 可持久化数组模板

你需要维护这样的一个长度为 N 的数组,支持如下两种操作: 1.在某个历史版本上修改某一个位置上的值 2.访问某个历史版本上的某一位置的值 每一次操作1/2都记为一个历史版本,即使询问没有改变数组

- •建立可持久化<mark>位置</mark>线段树,只有最开始对数组建立完整的线段树,之后由于只修改一个位置,在线段树上只有log(n)个节点需要修改,所以第i个版本直接继承rt $[v_i]$ 节点,左右儿子没有修改的直接连向历史版本,修改的儿子建立新节点,继续递归即可。
- •时空复杂度O(nlogn)

例题三

暑期集训 2022

你已经学会了主席树接下来开始做题吧!

没什么事,请你继续刷w



我们要飞上去噜!!

風になってしまえよ僕ら~♪

没什么事,请你继续刷w



我们要飞上去噜!!

風になってしまえよ僕ら~』

早苗入手了最新的高级打字机。最新款自然有着与以往不同的功能,那就是它具备撤销功能,厉害吧。

请为这种高级打字机设计一个程序, 支持如下 3 种操作:

1. T x: 在文章末尾打下一个小写字母 x。 (type 操作)

2. □ x : 撤销最后的 x 次修改操作。 (Undo 操作)

(注意 Query 操作并不算修改操作)

3. Q x: 询问当前文章中第 x 个字母并输出。 (Query 操作)

文章一开始可以视为空串。

强制在线捏~~~ 注意Undo可以撤销Undo

Luogu **1383**

```
if (ch == 'T')
    ch = read();
    ++cnt; len[cnt] = len[cnt - 1] + 1;
    PST::add(rt[cnt], rt[cnt - 1], 1, n, len[cnt], ch);
if (ch == 'U')
    scanf("%d", &x);
    ++cnt;
    rt[cnt] = rt[cnt - x - 1];
    len[cnt] = len[cnt - x - 1];
```

•只需要再开一个数组*len*,Undo时同*rt*一起继承历史版本的字符串长度;修改时即可在*len*的位置上直接单点修改,其余细节同可持久化位置线段树模板

Luogu **1383**

给一个长度为 n 的正整数序列 a。共有 m 组询问,每次询问一个区间 [l,r] ,是否存在一个数在 [l,r] 中出现的次数严格大于一半。如果存在,输出这个数,否则输出 0。

 $1 \leq n, m \leq 5 imes 10^5$, $1 \leq a_i \leq n$,

输入#1

输出#1

1 0 3 0 4

POI2014 Luogu **3567**

给一个长度为 n 的正整数序列 a。共有 m 组询问,每次询问一个区间 [l,r] ,是否存在一个数在 [l,r] 中出现的次数严格大于一半。如果存在,输出这个数,否则输出 0。

$$1 \leq n, m \leq 5 imes 10^5$$
 , $1 \leq a_i \leq n$,

- •可持久化权值线段树(卡莫队捏!)
- •在树上搜索答案时,不断往 $val[u] val[pre] \ge k$ 的儿子走,直到叶子结点输出权值,否则无解。k = (r-l+1)/2+1
- •随机化

POI2014 Luogu **3567**

一家餐厅有 n 道菜,编号 1,2,...,n ,大家对第 i 道菜的评价值为 a_i 。有 m 位顾客,第 i 位顾客的期望值为 b_i ,而他的偏好值为 x_i 。因此,第 i 位顾客认为第 j 道菜的美味度为 b_i xor $(a_j + x_i)$,xor 表示异或运算。

第i位顾客希望从这些菜中挑出他认为最美味的菜,即美味值最大的菜,但由于价格等因素,他只能从第 l_i 道到第 r_i 道中选择。请你帮助他们找出最美味的菜。

第1行,两个整数, n, m,表示菜品数和顾客数。

对于 100% 的数据,满足 $1 \le n \le 2 \times 10^5$, $0 \le a_i, b_i, x_i < 10^5$, $1 \le l_i \le r_i \le n$ ($1 \le i \le m$), $1 \le m \le 10^5$ 。

第 2 行,n 个整数, a_1, a_2, \ldots, a_n ,表示每道菜的评价值。

第 $3 \subseteq m + 2$ 行,每行 4 个整数,b, x, l, r,表示该位顾客的期望值,偏好值,和可以选择菜品区间。

输入#1

输出#1

Luogu3293

一家餐厅有 n 道菜,编号 1,2,...,n ,大家对第 i 道菜的评价值为 a_i 。有 m 位顾客,第 i 位顾客的期望值为 b_i ,而他的偏好值为 x_i 。因此,第 i 位顾客认为第 j 道菜的美味度为 b_i xor (a_j+x_i) ,xor 表示异或运算。

第i位顾客希望从这些菜中挑出他认为最美味的菜,即美味值最大的菜,但由于价格等因素,他只能从第 l_i 道到第 r_i 道中选择。请你帮助他们找出最美味的菜。

- •从高到低位贪心,若在二进制下,b的第j位为 b_j ,则查找[l,r]存不存在使x + a的第j位为 $b_j \oplus 1$ 的数,即主席树里实际查询的区间范围要减去x
- •若存在,则 $ans += 2^{j}$,下一次在主席树搜索的权值区间也相应的要增加 2^{j} ;
- •若不存在,则ans不变,继续做下一位;
- •时间复杂度O(nlognloga), 空间复杂度O(nlogn)

Luogu3293

前几道题是较为明显的主席树,而进阶题和难题往往用主席树维护的东西很难一眼看出来

需要较深的思考或者其他数据结构的配合,比如dfs序上建主席树、SAM+主席树等等。

所以要求我们把基本的主席树模型掌握,才能在需要维护一个信息的时候,一眼发现,哦~这个可以用主席树!

权值和位置两种写法都需要反复理解、牢记在心!



小总结

休息一会儿!





题目大意

火山国家有 n 个城市,编号从 1 到 n ,城市 1 是首都,有一座大火山。城市 i 的温度是 t_i 。

n 个城市用 n-1 条无向边相连,第 i 条边连接城市 u_i 和 v_i 。如果 u_i 比 v_i 更接近首都,则 $t_{u_i} > t_{v_i}$ 。首都的温度是最高的。

如果在城市 x 爆发生存温度为 [l,r] 的病毒,求会感染多少城市。

若一个城市的温度位于 [l,r] 之内,且它与另一个受感染的城市相连,则它也会被感染。

多组询问,强制在线

Eyjafjalla

Eyjafjalla

题目大意

火山国家有 n 个城市,编号从 1 到 n ,城市 1 是首都,有一座大火山。城市 i 的温度是 t_i 。

n 个城市用 n-1 条无向边相连,第 i 条边连接城市 u_i 和 v_i 。如果 u_i 比 v_i 更接近首都,则 $t_{u_i} > t_{v_i}$ 。首都的温度是最高的。

如果在城市 x 爆发生存温度为 [l,r] 的病毒, 求会感染多少城市。

若一个城市的温度位于 [1,r] 之内,且它与另一个受感染的城市相连,则它也会被感染。

- •先倍增跳到深度最浅且温度 $t \in [l,r]$ 的节点u,答案为u的子树内,温度在l到r间的节点个数
- •在DFS序上建立主席树,记录入栈序 in_u 和出栈序 out_u ,访问入栈序和出栈序所对应的rt,在DFS序上做与上文前缀和相同的方式,相减即可
- •时空复杂度O(nlogn)

给你n个正整数 a_i ,有m个询问,每次询问用区间[l,r]内的数求和,最小不能表示的正整数是多少(只能从选定区间中选且每个数最多只能选一次),强制在线

如[1, 4, 2]最小不能表示的数为8 [2, 1, 6]最小不能表示的数为4

> 2020ICPC昆明 Stone Games

给你n个正整数 a_i ,有m个询问,每次询问用区间[l,r]内的数求和,最小不能表示的正整数是多少(只能从选定区间中选且每个数最多只能选一次),强制在线

- •如果当前区间没有1,那么答案肯定是1
- •假设当前有t个1,那能表示出的数为[1,t],之后最大的数为z。若z > t + 1,则t + 1即为答案,否则[1,t + z]也可以表示出来
- •所以,对于每次的区间[1,t],都需要找出在区间[l,r]内小于等于t的数之和记为 $sum_{l,r}^t$,并更新 $t = sum_{l,r}^t$,之后接着重复算法过程,当t不变时,t+1即为答案。
- •这个过程最多不超过 $\log a_i$ 次(斐波那契)
- •查询 $sum_{l,r}^t$ 可用主席树维护,即用val[rt[r]] val[rt[l-1]]之后根据t再接着访问rt[r]和rt[l-1]的左儿子或右儿子,单次时间复杂度O(logn)
- •总时间复杂度 $O(nlognloga_i)$,空间复杂度 $O(nloga_i)$

给一个长为n的串S,共Q次询问,每次询问给定x,y,求将S长为x的前缀和长为y的后缀拼接得到的新串在S中 出现了多少次。T组数据。

$$1 \leq T \leq 5, 1 \leq n, Q \leq 2 imes 10^5, 1 \leq x,y \leq n$$

Sample Input

6 3 ababaa

Sample Output

HDU7084

HDU7084

给一个长为n的串S,共Q次询问,每次询问给定x,y,求将S长为x的前缀和长为y的后缀拼接得到的新串在S中出现了多少次。T组数据。

$$1 \leq T \leq 5, 1 \leq n, Q \leq 2 \times 10^5, 1 \leq x, y \leq n$$

•考虑KMP的fail树,树上的每个节点代表一个前缀,节点的父亲为该前缀的最长公共前后缀。也就是说,该节点的所有祖先,都是该前缀的一个border,next数组是最长的border;对于后缀,反序KMP后也有相同的结论

|若
$$S_{1...x}=S_{|S|-x+1,|S|}x
eq |S|$$
,则 $S_{1...x}$ 为 S 的border。

- •如果一组前缀和后缀拼起来在原串中每次出现,必然分界点为一组前缀的 border与后缀border的组合
- •建出两颗fail树,将每个节点同前缀/后缀——对应,则问题转化为,在x的子树内,有多少个节点也在y子树内
- •将一棵树按dfs序重新映射,此时权值变为连续;在第二棵树上建权值主席树维护映射过后的点权即可,时空复杂度0 (nlogn)

Sample Input

```
1
6 3
ababaa
1 1
2 1
3 1
```

一个长度为 n 的序列 a ,设其排过序之后为 b ,其中位数定义为 $b_{n/2}$,其中 a,b 从 0 开始标号,除法取下整。

给你一个长度为 n 的序列 s。

回答 Q 个这样的询问: s 的左端点在 [a,b] 之间,右端点在 [c,d] 之间的子区间中,最大的中位数。

其中 a < b < c < d。

位置也从 0 开始标号。

我会使用一些方式强制你在线。

Luogu2839

Luogu2839

给你一个长度为 n 的序列 s。

回答 Q 个这样的询问: s 的左端点在 [a,b] 之间,右端点在 [c,d] 之间的子区间中,最大的中位数。 其中 a < b < c < d。

- •区间中位数可以考虑二分答案,大于mid的设为1,小于设为-1,查询区间内的和是否≥ 0 [b+1,c-1]是必选的,直接记录他们的和
- 接下来就是找[a,b]的最大后缀和[c,d]的最大前缀,所以对于每一个二分的mid值,建一棵线段树,记录区间和,区间最大后缀,最大前缀。
- 但是每次二分答案都需要建树,时空复杂度大大超限
- •发现对于数字每次增大,只有当前中位数一个值从+1变为-1
- •只需要一开始就把所有*mid*情况的线段树都建好→使用主席树,具体来说主席树上每一个节点同时维护区间和,区间最大后缀,最大前缀三个信息,开始建树的时候先按序列大小排序,再把+1到-1的修改信息,新开一个主席树的根就行了。

时间复杂度 $O(nlog^2n)$, 空间复杂度O(nlogn)

参考文献

算法学习笔记(2): 树状数组 https://zhuanlan.zhihu.com/p/93795692 算法学习笔记(14): 线段树 https://zhuanlan.zhihu.com/p/106118909 动态开点,线段树合并和可持久化线段树 https://blog.51cto.com/u_15060465/4167050 主席树详解——让你躺着学会主席树 https://blog.csdn.net/Foxreign/article/details/118770162 权值线段树详解 https://blog.csdn.net/yanweiqi1754989931/article/details/117380913 树套树 P2617 Dynamic Rankin https://www.luogu.com.cn/blog/zhoushuyu/solution-p2617 树状数组套主席树 https://blog.csdn.net/yanweiqi1754989931/article/details/119951353

补充资料

Thanks J(·ω·)/