数据结构

九条可怜

杭州天水幼儿园

基础框架

- 左偏树

基础框架

- 左偏树
- 线段树

基础框架

- 左偏树
- 线段树
- 平衡树

基础框架

- 左偏树
- 线段树
- 平衡树
- 树套树

基础框架

- 左偏树
- 线段树
- 平衡树
- 树套树
- 树链剖分

基础框架

- 左偏树
- 线段树
- 平衡树
- 树套树
- 树链剖分
- LCT

基础框架

- 左偏树
- 线段树
- 平衡树
- 树套树
- 树链剖分
- LCT
- 点分树

基础框架

- 左偏树
- 线段树
- 平衡树
- 树套树
- 树链剖分
- LCT
- 点分树
- kd-tree

左偏树

- $w_i \le w_{l_i}, w_i \le w_{r_i}$

左偏树

- $w_i \le w_{l_i}, w_i \le w_{r_i}$
- 左子树深度不小于右子树深度

左偏树

- $w_i \le w_{l_i}, w_i \le w_{r_i}$
- 左子树深度不小于右子树深度
- 左子树和右子树都是左偏树

左偏树

- $w_i \le w_{l_i}, w_i \le w_{r_i}$
- 左子树深度不小于右子树深度
- 左子树和右子树都是左偏树
- **一直向右走O(log n)步就能到达叶子节点**

左偏树

```
struct tree{
    int l,r,w,d;
}t[N];
int merge(int k1,int k2){
    if (k1==0||k2==0) return k1+k2;
    if (t[k1].w>t[k2].w) swap(k1,k2);
    t[k1].r=merge(t[k1].r,k2);
    if (t[t[k1].l].d<t[t[k1].r].d) swap(t[k1].l,t[k1].r);
    t[k1].d=t[t[k1].r].d+1;
    return k1;
}
```

左偏树

- 插入：合并原树和新节点。

左偏树

- 插入：合并原树和新节点。
- 删除最小值：合并左右子树。

JLOI2015 城池攻占

一棵带点权 f_i 的树，有 m 个人，最开始第 i 个人在 a_i，能力值为 w_i。

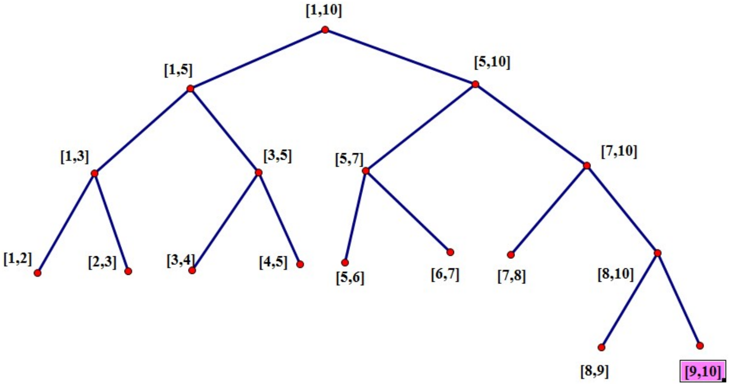
每个人开始往上走，如果能力值小于所在点点权，这个人就在这儿停下了。否则这个人的能力值会乘以 b_i 再加上 c_i 然后这个人走到当前节点的父节点。

问每个人向上走了多少步以及结束时每一个节点中有多少个人。

保证每一个人的能力值不会超过 10^18

$n, m \leq 10^5$

线段树



线段树

- 动态开节点的线段树

线段树

- 动态开节点的线段树
- 主席树

线段树

- 动态开节点的线段树
- 主席树
- 离散化之前，在 $[1, 10^9]$ 范围内建立线段树。

线段树

- 动态开节点的线段树
- 主席树
- 离散化之前，在 $[1, 10^9]$ 范围内建立线段树。
- 进行了若干次操作，但是要保留每一次的版本。

一个可持久化问题

区间加，询问第 i 次操作后的区间和，强制在线。

$n, m \leq 10^5$

主席树

- 线段树每一次操作只会有 $O(\log n)$ 个位置发生了变化。

主席树

- 线段树每一次操作只会有 $O(\log n)$ 个位置发生了变化。
- 对这 $O(\log n)$ 个位置新建节点，其余则沿用已有的节点。

主席树

- 线段树每一次操作只会有 $O(\log n)$ 个位置发生了变化。
- 对这 $O(\log n)$ 个位置新建节点，其余则沿用已有的节点。
- 时间空间都是 $O(n\log n)$。

例题

给出一棵带点权的树，询问路径 k 小值。

$n, m \leq 10^5$

BZOJ2635 middle

给出一个数列，询问左端点在 $[a,b]$，右端点在 $[c,d]$ 的区间中中位数的最大值。

对于一个长度为 K 的序列，中位数定义为第 $\lceil \frac{n}{2} \rceil$ 大的数。

$n, m \leq 10^5$

树上 LIS

给出一棵点权 $[1, n]$ 的树，求树上的 LIS。

$n \le 10^5$

BZOJ2635 middle

- 对于序列 A，对于数 k，构造数列 B，其中如果 $A_i > k$，那么 $B_i = -1$ 否则 $B_i = 1$

BZOJ2635 middle

- 对于序列 A，对于数 k，构造数列 B，其中如果 $A_i > k$，那么 $B_i = -1$ 否则 $B_i = 1$
- k 大于等于中位数当且仅当 $\sum B_i \geq 0$

BZOJ2635 middle

- 对于序列 A，对于数 k，构造数列 B，其中如果 $A_i > k$，那么 $B_i = -1$ 否则 $B_i = 1$
- k 大于等于中位数当且仅当 $\sum B_i \geq 0$
- 对于每一个 k，建立线段树维护 B，这个可以用主席树做到。

BZOJ2635 middle

- 对于序列 A，对于数 k，构造数列 B，其中如果 $A_i > k$，那么 $B_i = -1$ 否则 $B_i = 1$
- k 大于等于中位数当且仅当 $\sum B_i \geq 0$
- 对于每一个 k，建立线段树维护 B，这个可以用主席树做到。
- 二分答案，求第 i 棵线段树中，左端点在 $[a, b]$，右端点在 $[c, d]$ 的区间的和的最大值。

BZOJ2635 middle

- 对于序列 A，对于数 k，构造数列 B，其中如果 $A_i > k$，那么 $B_i = -1$ 否则 $B_i = 1$
- k 大于等于中位数当且仅当 $\sum B_i \geq 0$
- 对于每一个 k，建立线段树维护 B，这个可以用主席树做到。
- 二分答案，求第 i 棵线段树中，左端点在 $[a, b]$，右端点在 $[c, d]$ 的区间的和的最大值。
- 维护最大前缀和最大后缀和以及区间和就可以了。

BZOJ2635 middle

- 对于序列 A，对于数 k，构造数列 B，其中如果 $A_i > k$，那么 $B_i = -1$ 否则 $B_i = 1$
- k 大于等于中位数当且仅当 $\sum B_i \geq 0$
- 对于每一个 k，建立线段树维护 B，这个可以用主席树做到。
- 二分答案，求第 i 棵线段树中，左端点在 $[a,b]$，右端点在 $[c,d]$ 的区间的和的最大值。
- 维护最大前缀和最大后缀和以及区间和就可以了。
- $O(n\log^2 n)$

三代平衡树

- 旋转

三代平衡树

- 旋转
- 暴力重构

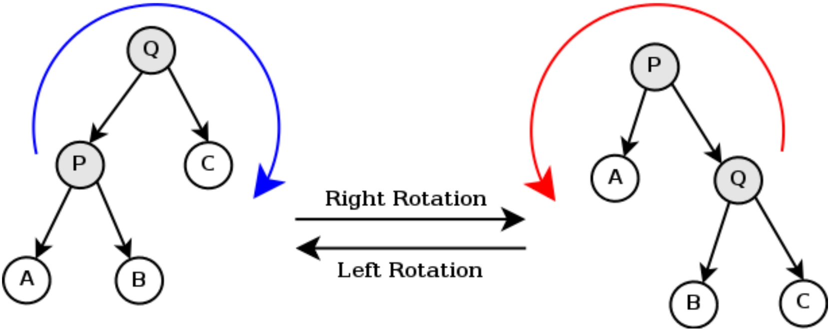
三代平衡树

- 旋转
- 暴力重构
- 合并与分裂

Splay

- 又慢又难写还不能可持久化，但是必须要学啊..

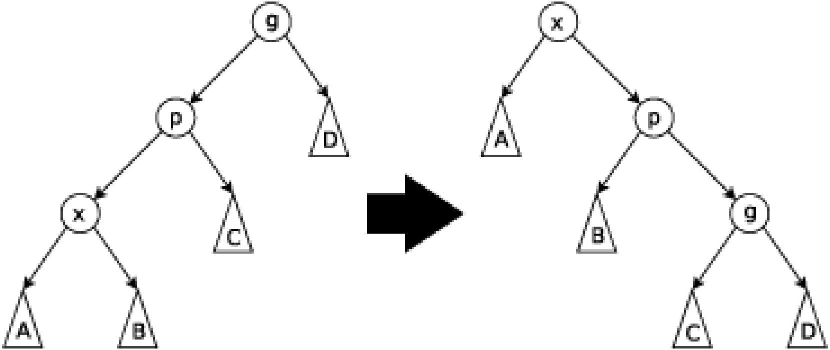
Splay

- 又慢又难写还不能可持久化，但是必须要学啊..
- 

双旋

- 特殊处理连续的左旋和连续的右旋。

双旋

- 特殊处理连续的左旋和连续的右旋。
- 

BZOJ3224 普通平衡树

您需要写一种数据结构（可参考题目标题），来维护一些数，其中需要提供以下操作：

1) 插入 x 数
2) 删除 x 数 (若有多个相同的数，因只删除一个)
3) 查询 x 数的排名 (若有多个相同的数，因输出最小的排名)
4) 查询排名为 x 的数
5) 求 x 的前驱 (前驱定义为小于 x，且最大的数)
6) 求 x 的后继 (后继定义为大于 x，且最小的数)

笛卡尔树

- 笛卡尔树的每一个点有两个权值 f 和 w。

笛卡尔树

- 笛卡尔树的每一个点有两个权值 f 和 w。
- w 满足排序二叉树的性质。

笛卡尔树

- 笛卡尔树的每一个点有两个权值 f 和 w。
- w 满足排序二叉树的性质。
- f 满足堆的性质。

笛卡尔树

- 笛卡尔树的每一个点有两个权值 f 和 w。
- w 满足排序二叉树的性质。
- f 满足堆的性质。
- 给定 f 和 w 笛卡尔树是唯一的。

笛卡尔树

- 笛卡尔树的每一个点有两个权值 f 和 w。
- w 满足排序二叉树的性质。
- f 满足堆的性质。
- 给定 f 和 w 笛卡尔树是唯一的。
- Treap 就是 w 随机的笛卡尔树，深度期望 $O(\log n)$。

笛卡尔树

- 给定 n 个二元组 (f_i, w_i)，建立笛卡尔树。

- 给定 n 个二元组 (f_i, w_i)，建立笛卡尔树。
- 按照 w 排序，依次插入。

笛卡尔树

- 给定 n 个二元组 (f_i, w_i)，建立笛卡尔树。
- 按照 w 排序，依次插入。
- 每一次插入后新的点一定是根节点一直向右走走到的点。

笛卡尔树

- 给定 n 个二元组 (f_i, w_i)，建立笛卡尔树。
- 按照 w 排序，依次插入。
- 每一次插入后新的点一定是根节点一直向右走走到的点。
- 用栈维护。

FHQ Treap

- 相比传统 Treap 好写好调。

FHQ Treap

- 相比传统 Treap 好写好调。
- 核心操作：按照阈值 K 分裂，合并两棵值域不相交的 Treap。

FHQ Treap

```
int merge(int k1,int k2){
    if (k1==0||k2==0) return k1+k2;
    if (t[k1].f>t[k2].f){
        t[k1].r=merge(t[k1].r,k2); change(k1); return k1;
    } else {t[k2].l=merge(k1,t[k2].l); change(k2); return k2;}
}
void splite(int k1,int k2,int &k3,int &k4){
    if (k1==0){k3=0; k4=0; return;}
    if (t[k1].w<k2){
        k3=k1; splite(t[k1].r,k2,t[k1].r,k4);
    } else {k4=k1; splite(t[k1].l,k2,k3,t[k1].l);}
    change(k1);
}
```

可持久化

- FHQ Treap 每一次操作都是严格 $O(\log n)$ 的。

可持久化

- FHQ Treap 每一次操作都是严格 $O(\log n)$ 的。
- 因此可以和主席树一样的保留历史版本。

可持久化

- FHQ Treap 每一次操作都是严格 $O(\log n)$ 的。
- 因此可以和主席树一样的保留历史版本。
- 时间空间都是 $O(n \log n)$ 的。

没有人的数论

定义好的数：

1) 0 是好的数
2) 如果 x, y 是好的数，那么 (x, y) 是好的数。
3) 0 小于任何非 0 数。
4) $(a, b) < (c, d)$ 当且仅当 $a < c$ 或 $a = c, b < d$。

给出一个数组 A，初始所有数都是 0。

两种操作，询问区间最小的位置或者给出 a, b, c 将 A_c 变成 (A_a, A_b)

$n, m \leq 10^5$

树套树

1) 线段树套线段树
2) 线段树套平衡树
3) 树状数组套主席树
4) 替罪羊树套主席树

您需要写一种数据结构（可参考题目标题），来维护一些数，其中需要提供以下操作：

1) 查询 k 在区间内的排名
2) 查询区间内排名为 k 的值
3) 修改某一位值上的数值
4) 查询 k 在区间内的前驱 (前驱定义为小于 x，且最大的数)
5) 查询 k 在区间内的后继 (后继定义为大于 x，且最小的数)

$n, m \leq 5 \times 10^4$

例题

给出一个排列，每次交换两个数，询问逆序对个数。

$n, m \leq 5 \times 10^4$

有 n 个位置，m 个操作。

操作有两种，如果是 $1\ a\ b\ c$ 的形式表示在第 a 个位置到第 b 个位置，每个位置加入一个数 c。如果是 $2\ a\ b\ c$ 形式，表示询问从第 a 个位置到第 b 个位置，第 C 大的数是多少。

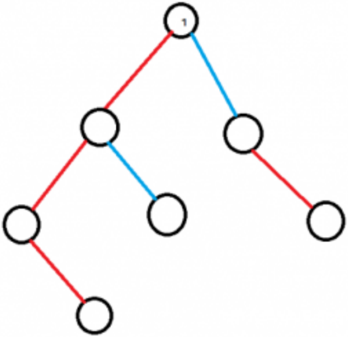
$n, m \leq 5 \times 10^4$

BZOJ3065 带插入区间 K 小值

在第 i 个位置后插入一个数，单点修改，询问区间 K 小值。

$n, m \leq 10^5$

树链剖分



SDOI2011 染色

一棵带点权的树，链覆盖，询问链上有几个颜色段。

$n, m \leq 10^5$

BZOJ3038 遥远的国度

给定一棵有根树，每个点有一个权值，提供三种操作：

1) 将 x 节点变为根节点
2) 将 x 到 y 路径上的点的权值全部改为 v
3) 询问 x 的子树中点权的最小值

$n, m \leq 10^5$

- 动态树链剖分。

LCT

- 动态树链剖分。
- 用 splay 维护每一条重链的相对顺序。

LCT

- 动态树链剖分。
- 用 splay 维护每一条重链的相对顺序。
- 基本操作：access, makeroot, link, cut

给定一张图，每一条边有两个权值 a_i 和 b_i。

你可以带两种精灵，只有当第一种精灵数量不小于 a_i，第二种不小于 b_i 的时候你才能通过那条边。

问最少带多少数量的精灵才能从 1 走到 n。

$n, m \leq 10^5$

给出一棵树，点权为 $[1,n]$，询问树上的 LIS。

$n, m \leq 10^5$

点分树

给出一棵带点权的单位边权的树，询问距离 u 不超过 d 的所有点的点权最大值。

$n, m \leq 10^5$

给出一棵带点权的树，单点修改点权，询问带权重心到所有点的带权距离之和。

每一个点的度数不超过 20。

$n \leq 10^5$

广告

- http://www.genshuixue.com/i/zjzroi

广告

- http://www.genshuixue.com/i/zjzroi
- 有关课程方面可以咨询 QQ 81569188

谢谢大家