

李诗懿

数据结构选讲

维护一个数组

- † 给出长度为 n 的一个数列，完成 m 次操作
- † 单点修改、单点询问
- † 区间修改、区间询问
- † 删除插入、区间移动

维护一个堆 *

维护一棵树（或森林）

- † 给出大小为 n 的一棵树，完成 m 次操作
- † 查询点到根的一条路径
- † 修改一棵子树
- † 删除添加一棵子树

维护一棵仙人掌*

† 仙人掌：每条边和每个点都至多属于一个简单环的无向连通图

分块

- † 分块思想
- † 块状数据结构:
 - † 块状数组
 - † 块状链表
 - † 树块剖分

块状数组

- † 给出长度为 n ($\leq 10^5$) 的一个正整数数列，完成 m ($\leq 10^5$) 次操作，操作分为 2 种：
- † $M\ x\ y\ k$ 表示将 $[x,y]$ 的数字全部修改为 k
- † $Q\ x\ y\ k$ 表示询问 $[x,y]$ 上的最大值

块状链表

- † 给一个初始为空的数列，完成以下几种操作：
- † $I \ x \ n$ ，在第 x 个数前面插入 n 个数；
- † $D \ x \ n$ ，从第 x 个数开始删除连续的 n 个数；
- † $Q \ x \ n$ ，输出从 x 个数开始的连续 n 个数的和。

hnoi2010 弹飞绵羊

- † 一条直线上 n 个装置，每个装置设定初始弹力系数 k_i ，当绵羊达到第 i 个装置时，它会往后弹 k_i 步，达到第 $i+k_i$ 个装置，若不存在第 $i+k_i$ 个装置，则绵羊被弹飞。绵羊想知道当它从第 i 个装置起步时，被弹几次后会被弹飞。Lostmonkey 可以修改某个弹力装置的弹力系数，任何时候弹力系数均为正整数。
- † $n \leq 200000, m \leq 100000$

树块剖分

- † 把树剖分成若干个连通块
- † 维护每个点到连通块的根的信息和
- † 如何做到根号 n 并且实现简单:
 - † 点到根的路径上的树块数为 $O(\sqrt{N})$
 - † 每个树块大小 $\leq \sqrt{N}$
- † 实现: 尝试合并 dfs 入栈序相邻的两个结点

树块剖分

- ✦ 如何做到动态:
 - ✦ 查询时更新
- ✦ 好处: 修改的时候可以“奔放”地处理树块

分块

- ✦ 优点：适应各种数据结构问题，代码简单，常数小
- ✦ 缺点：理论复杂度根号 n

树状数组

- † 支持：满足区间加法区间减法
- † 修改：区间加减
- † 询问：
 - † 区间求和 $\log n$
 - † 区间求最值：暴力修改 $\log^2 n$
- † 优点：代码简单 空间小 常数低
- † 缺点：扩展性低 适用范围小

线段树

- † 支持：只需要满足区间加法
- † 堆式线段树 非堆式线段树
- † 自底向上线段树 自顶向下线段树
- † 自底向上堆式线段树：常数小（**zkw** 线段树）
- † 自顶向下非堆式线段树：可持久化（主席树）
- † KD 树 * 、四分树 *

bzoj 3585 mex

- † 有一个长度为 n 的数组 $\{a_1, a_2, \dots, a_n\}$ 。 m 次询问，每次询问一个区间内最小没有出现过的自然数。

ONTAK2010 Peaks

- † 在 Bytemountains 有 N 座山峰，每座山峰有他的高度 h_i 。有些山峰之间有双向道路相连，共 M 条路径，每条路径有一个困难值，这个值越大表示越难走，现在有 Q 组询问，每组询问询问从点 v 开始只经过困难值小于等于 x 的路径所能到达的山峰中第 k 高的山峰，如果无解输出 -1 。

不知名的例题

- ✦ 维护一个序列
- ✦ 操作：给区间加上一个等差数列
- ✦ 询问：区间最值

平衡树

- † splay 扩展：动态树
- † treap 扩展：可持久化平衡树
- † 替罪羊树 *

伸展树

- † 1.splay 伸展操作
- † 2.insert 插入
- † 3.delete 删除
- † 4.select(k) 查询
- † 5. 翻转
- † 6. 循环平移

Hnoi2002 营业额统计

- † 我们知道某公司成立以来每天的营业额。
- † 经济学上定义了一种最小波动值：该天的最小波动值。分析整个公司的从成立到现在营业情况是否稳定，只需要把每一天的最小波动值加起来。你的任务就是编写一个程序来计算这一个值。
- † 第一天的最小波动值为第一天的营业额。

Scoi2014 方伯伯的 oj

- † n 个用户，编号 $1 \sim n$ ，一开始按照编号排名。维护以下操作
- † $1\ x\ y$ 将 x 的编号改为 y ，输出该用户的排名。保证 x 在队列中，并且 y 不在队列中
- † $2\ x$ 输出编号 x 的排名，并将 x 的移到第一位
- † $3\ x$ 输出编号 x 的排名，并将 x 的移到最后一位
- † $4\ k$ 查询当前排名为 k 的用户的编号并输出
- † 强制在线