

# FHQ Treap

前置知识

讲解044-前缀树 整个专题的要求

有序表专题安排

专题1: AVL树, 讲解148

专题2: 跳表, 讲解149

专题3: 替罪羊树, 讲解150

专题4: 笛卡尔树、Treap树, 讲解151

专题5: FHQ Treap树, 讲解152, 本节

专题6: Splay树, 讲解153

大厂笔试、算法竞赛掌握以上有序表结构足够, 其他有序表结构不再讲述, 面试遇到只是聊, 可以自行学习  
算法竞赛的同学, 有序表必带模版: 替罪羊树、Treap树、FHQ Treap树、Splay树  
Splay树是实现Link-Cut-Tree的关键, 这个结构的讲述, 会在【挺难】阶段的课程里安排

# FHQ Treap

本节课的前置知识

讲解110 - 线段树，本节课题目4、题目6需要

讲解151 - Treap树，理解FHQ-Treap需要

本节课讲述

FHQ-Treap的原理，重点解析split过程、merge过程

FHQ-Treap实现普通有序表，使用词频压缩，题目1

FHQ-Treap实现普通有序表，不用词频压缩，FHQ-Treap最常规的实现，题目2

FHQ-Treap实现区间移动，题目3

FHQ-Treap实现范围翻转，题目4

FHQ-Treap可持久化的设计，题目5、题目6

注意，FHQ-Treap可持久化和线段树可持久化，设计非常相似

可持久化线段树，也叫主席树，会在【挺难】课程阶段，进行重点讲述

# FHQ Treap

为了所有语言背景的同学，都能理解FHQ-Treap，并实现出自己语言的版本

本节课不采用传递地址的编程设计！效率也完全不会变差！因为传递地址这个特性支持的语言较少！

本节课所有题目，都提供了java和C++的实现，也适合各种语言背景的同学进行学习

FHQ-Treap的分裂，split方法，课上重点图解

可以根据key分裂树， $\leq \text{key}$ 的节点变成一棵搜索二叉树， $> \text{key}$ 的节点变成另一棵搜索二叉树

也可根据排名分裂树， $\leq$ 排名的节点变成一棵搜索二叉树， $>$ 排名的节点变成另一棵搜索二叉树

最终分别得到两棵树的头节点编号， $\leq$ 部分的头记录在`right[0]`， $>$ 部分的头记录在`left[0]`

FHQ-Treap的合并，merge方法，课上重点图解

给定两棵搜索二叉树的头`l`和`r`，要求，`l`整棵树的key值或排名值 总体小于 `r`整棵树的key值或排名值

每一步谁优先级大谁做头，合并成一棵树，最终返回整棵树的头节点编号

FHQ-Treap是最常用的有序表结构、可持久化有序表结构，原理简单、实现简单、扩展性好、常数时间好

# FHQ Treap

## 题目1

FHQ-Treap, 使用词频压缩

实现一种结构, 支持如下操作, 要求单次调用的时间复杂度 $O(\log n)$

1, 增加 $x$ , 重复加入算多个词频

2, 删除 $x$ , 如果有多个, 只删掉一个

3, 查询 $x$ 的排名,  $x$ 的排名为, 比 $x$ 小的数的个数+1

4, 查询数据中排名为 $x$ 的数

5, 查询 $x$ 的前驱,  $x$ 的前驱为, 小于 $x$ 的数中最大的数, 不存在返回整数最小值

6, 查询 $x$ 的后继,  $x$ 的后继为, 大于 $x$ 的数中最小的数, 不存在返回整数最大值

所有操作的次数  $\leq 10^5$

$-10^7 \leq x \leq +10^7$

测试链接 : <https://www.luogu.com.cn/problem/P3369>

# FHQ Treap

## 题目2

FHQ-Treap, 不用词频压缩, FHQ-Treap最常规的实现

实现一种结构, 支持如下操作, 要求单次调用的时间复杂度 $O(\log n)$

1, 增加 $x$ , 重复加入算多个词频

2, 删除 $x$ , 如果有多个, 只删掉一个

3, 查询 $x$ 的排名,  $x$ 的排名为, 比 $x$ 小的数的个数+1

4, 查询数据中排名为 $x$ 的数

5, 查询 $x$ 的前驱,  $x$ 的前驱为, 小于 $x$ 的数中最大的数, 不存在返回整数最小值

6, 查询 $x$ 的后继,  $x$ 的后继为, 大于 $x$ 的数中最小的数, 不存在返回整数最大值

所有操作的次数  $\leq 10^5$

$-10^7 \leq x \leq +10^7$

测试链接 : <https://www.luogu.com.cn/problem/P3369>

左树上的key  $\leq$  头节点的key  $\leq$  右树上的key, 其实这种设定是FHQ-Treap最常规的设定

# FHQ Treap

## 题目3

文本编辑器，FHQ-Treap实现区间移动

一开始文本为空，光标在文本开头，也就是1位置，请实现如下6种操作

Move k : 将光标移动到第k个字符之后，操作保证光标不会到非法位置

Insert n s : 在光标处插入长度为n的字符串s，光标位置不变

Delete n : 删除光标后的n个字符，光标位置不变，操作保证有足够字符

Get n : 输出光标后的n个字符，光标位置不变，操作保证有足够字符

Prev : 光标前移一个字符，操作保证光标不会到非法位置

Next : 光标后移一个字符，操作保证光标不会到非法位置

Insert操作时，字符串s中ASCII码在[32,126]范围上的字符一定有n个，其他字符请过滤掉

测试链接 : <https://www.luogu.com.cn/problem/P4008>

FHQ-Treap根据排名分裂、合并

# FHQ Treap

## 题目4

文艺平衡树，FHQ-Treap实现范围翻转

长度为 $n$ 的序列，下标从1开始，一开始序列为 $1, 2, \dots, n$

接下来会有 $k$ 个操作，每个操作给定 $l, r$ ，表示从 $l$ 到 $r$ 范围上的所有数字翻转

做完 $k$ 次操作后，从左到右打印所有数字

$1 \leq n, k \leq 10^5$

测试链接：<https://www.luogu.com.cn/problem/P3391>

需要理解线段树的懒更新机制

建立懒更新机制 + FHQ-Treap根据排名分裂、合并



# FHQ Treap

## 可持久化FHQ-Treap

所谓的可持久化，就是把每一个版本的树都记录下来

为了节省时间和空间，需要良好的可持久化设计

每次生成出一棵新版本的树，时间和空间的代价都为 $O(\log n)$

1, 可持久化的重点是理解底层节点会有复用的情况

2, 可持久化FHQ-Treap的分裂过程图解

3, 可持久化FHQ-Treap的合并过程图解

4, 执行一个具体功能的图解

5,  $i$ 号版本树的头，记录在 $head[i]$ 中，每个版本的头都记录

特别注意1, 保证0版本永远是空树，所以执行完split过程后:  $left[0] = right[0] = 0$

特别注意2, merge最好加入可持久化设计，为了支持直接合并两个版本的树，等其他操作

课上重点图解



# FHQ Treap

## 题目5

可持久化平衡树，FHQ-Treap实现，不用词频压缩

认为一开始是0版本的树，为空树，实现如下操作，操作一共发生n次

v 1 x : 基于v版本的树，增加一个x，生成新版本的树

v 2 x : 基于v版本的树，删除一个x，生成新版本的树

v 3 x : 基于v版本的树，查询x的排名，生成新版本的树状况=v版本状况

v 4 x : 基于v版本的树，查询数据中排名为x的数，生成新版本的树状况=v版本状况

v 5 x : 基于v版本的树，查询x的前驱，生成新版本的树状况=v版本状况

v 6 x : 基于v版本的树，查询x的后继，生成新版本的树状况=v版本状况

不管什么操作，都基于某个v版本，操作完成后得到新版本的树，但v版本不会变化

如果x的前驱不存在，返回 $-2^{31} + 1$ ，如果x的后继不存在，返回 $+2^{31} - 1$

$1 \leq n \leq 5 * 10^5$        $-10^9 \leq x \leq +10^9$

测试链接 : <https://www.luogu.com.cn/problem/P3835>

# FHQ Treap

## 题目6

可持久化文艺平衡树，FHQ-Treap实现

一开始序列为空，实现如下操作，操作一共发生 $n$ 次

$v\ 1\ x\ y$  : 基于 $v$ 版本的序列，在第 $x$ 个数后插入 $y$ ，生成新版本的序列

$v\ 2\ x$  : 基于 $v$ 版本的序列，删除第 $x$ 个数，生成新版本的序列

$v\ 3\ x\ y$  : 基于 $v$ 版本的序列，范围 $[x, y]$ 所有数字翻转，生成新版本的序列

$v\ 4\ x\ y$  : 基于 $v$ 版本的序列，查询范围 $[x, y]$ 所有数字的和，生成新版本的序列状况= $v$ 版本状况

不管什么操作，都基于某个 $v$ 版本，操作完成后得到新版本的序列，但 $v$ 版本不会变化

每种操作给定的参数都是有效的，插入数字的范围 $[-10^6, +10^6]$

$1 \leq n \leq 2 * 10^5$

本题目要求强制在线，具体规则可以打开测试链接查看

测试链接 : <https://www.luogu.com.cn/problem/P5055>

建立懒更新机制 + FHQ-Treap根据排名分裂、合并