

[推酷](#)

- [文章](#)
- [站点](#)
- [主题](#)
- [公开课](#)
- [活动](#)
- [客户端](#) [荐](#)
- [周刊](#)
 - [编程狂人](#)
 - [设计匠艺](#)
 - [创业周刊](#)
 - [科技周刊](#)
 - [Guru Weekly](#)
 - [一周拾遗](#)

关于主席树的读书笔记

• [登录](#)时间 2014-05-12 20:23:01 [CSDN博客](#)原文 <http://blog.csdn.net/xiaofengcanyuexj/article/details/25553521>主题 [线段树 Nosql](#)

最近开始了自己高级数据结构之旅，在这次旅行中，我将持续把一些高级的数据结构从理论到编码都过一遍，同时通过博客形式分享出来，希望大家指出不足之处！

本次所要学习的数据结构-主席树并不是什么新东西，不过是种线段树的变种，是一种可持久化的线段树；关于可持久化数据结构，一时找不到好的解释。下面是摘自大牛博客的解释：

可持久化数据结构

可持久化数据结构 (Persistent data structure) 就是利用函数式编程的思想使其支持询问历史版本、同时充分利用它们之间的共同数据来减少时间和空间消耗。

前面介绍过的伸展树就是一种典型的可持久化数据结构；本篇文章所要讲的主席树也是一种比较有特色的可持久化数据结构。关于可持久化数据结构，可以参考大牛博客：[可持久化数据结构介绍](#)。

主席树

主席树 就是利用函数式编程的思想来使线段树支持询问历史版本、同时充分利用它们之间的共同数据来减少时间和空间消耗的增强版的线段树。

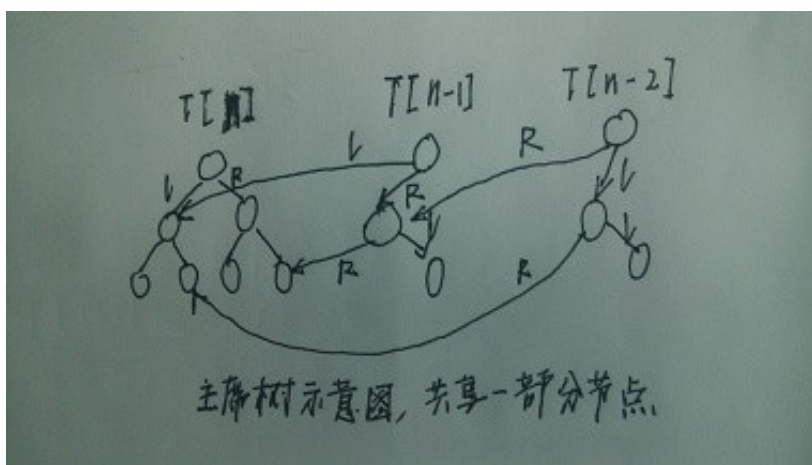
很多问题如果用线段树处理的话需要采用离线思想，若用主席树则可直接在线处理。故很多时候离线线段树求解可以转化为在线主席树求解。注意，主席树本质就是线段树，变化就在其实现可持久化，后一刻可以参考前一刻的副本，二者共同部分很多。一颗线段树的节点维护的是当前节点对应区间的信息，倘若每次区间都不一样，就会给处理带来一些困难。有时可以直接细分区间然后合并，此种情况线段树可以直接敲定；但有时无法通过直接划分区间来求解，如频繁询问区间第k小元素，当然，此问题有比较特殊的数据结构-划分树。关于划分树，可以参考我的博客：[划分树介绍](#)。此时，除了划分树之外，主席树也是可以解决的。

主席树的每个节点对应一颗线段树，此处有点抽象。在我们的印象中，每个线段树的节点维护的树左右子树下标以及当前节点对应区间的信息（信息视具体问题定）。对于一个待

处理的序列 $a[1]$ 、 $a[2]\cdots a[n]$ ，有 n 个前缀。每个前缀可以看做一棵线段树，共有 n 棵线段树；若不采用可持久化结构，带来的严重后果就是会MLE，即对内存来说很难承受。根据可持久化数据结构的定义，由于相邻线段树即前缀的公共部分很多，可以充分利用，达到优化目的，同时每棵线段树还是保留所有的叶节点只是较之前共用了很多共用节点。主席树很重要的操作就是如何寻找公用的节点信息，这些可能可能出现在根节点也可能出现在叶节点。

下面摘自 [生人理解](#)：所谓主席树呢，就是对原来的数列 $[1..n]$ 的每一个前缀 $[1..i]$ ($1 \leq i \leq n$) 建立一棵线段树，线段树的每一个节点存某个前缀 $[1..i]$ 中属于区间 $[L..R]$ 的数一共有多少个（比如根节点是 $[1..n]$ ，一共 i 个数， $\text{sum}[\text{root}] = i$ ；根节点的左儿子是 $[1..(L+R)/2]$ ，若不大于 $(L+R)/2$ 的数有 x 个，那么 $\text{sum}[\text{root.left}] = x$ ）。若要查找 $[i..j]$ 中第 k 大数时，设某结点 x ，那么 $x.\text{sum}[j] - x.\text{sum}[i-1]$ 就是 $[i..j]$ 中在结点 x 内的数字总数。而对每一个前缀都建一棵树，会MLE，观察到每个 $[1..i]$ 和 $[1..i-1]$ 只有一条路是不一样的，那么其他的结点只要用回前一棵树的结点即可，时空复杂度为 $O(n \log n)$ 。

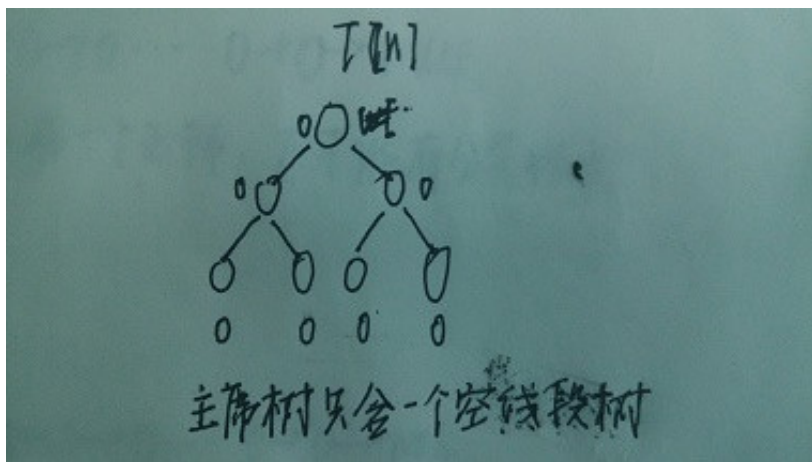
下面是我理解的一般形式上的主席树，如图：



主席树的很多很好的性质取决于操作的方便高效。下面依次介绍主席树的更新、查询等操作，给出效率分析。

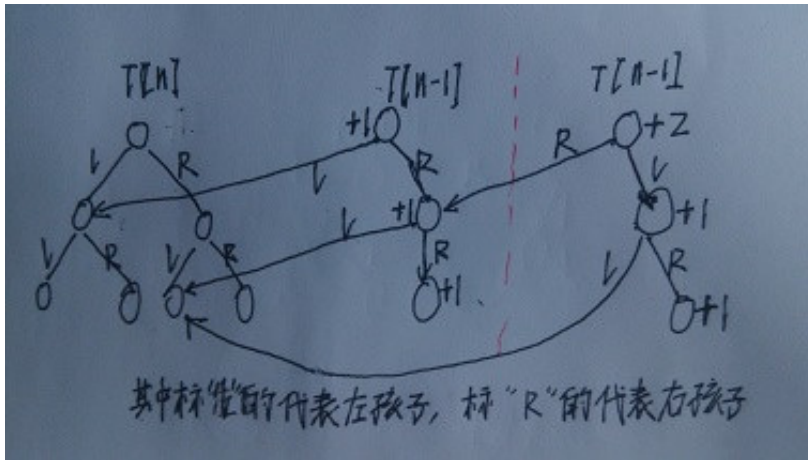
1、建立

首先建立一棵空的线段树，也是最原始的主席树，此时主席树只含一个空节点，然后依次对原序列按某种顺序更新，即将原序列加入到对应位置。此过程时间复杂度为 $O(M)$ ，空间复杂度为 $O(M \cdot \log(M))$ 。如图：



2、更新

更新时按照自己的离散值寻找对应位置，信息域 $data+1$ 。我们知道，更新一个叶节点只会影响根节点到该叶节点的一条路径，故只需修改该路径上的信息域 $data$ 。每个主席树的节点即每棵线段树的结构完全相同，只是对应信息域 $data$ 不同（可以理解为线段树的结构完全一样，只是对应叶子节点取值不同，从而有些节点的信息域 $data$ 不同，本质是节点节点不同），此时可以利用历史版本，即利用相邻的上一棵线段树的信息。相邻两颗线段树只有当前待处理的元素不同，其余位置完全一样。因此，如果待处理的元素进入线段树的左子树的话，右子树是完全一样的，可以共用，即直接让当前线段树节点的右子树指针指向相邻的上一棵线段树的右子树；若进入右子树，情况可以类比。此过程容易推出时间复杂度为 $O(\log(M))$ ，空间复杂度为 $O(\log(M))$ 。如图：



3. 查询

由于主席树每个节点是棵线段树，信息域、结构相同，可以相减。这是主席树查找的关键所在。例如查找第 k 小的元素，若左子树信息域 $data$ 之差大于等于 k ，则直接到左子树查找；否则调整 k 值即减去左子树的信息域 $data$ 之差，然后到相应的右子树查找。由于是线段树属于二叉树结构，故整个过程的时间复杂度为 $O(\log(M))$ ， M 往往是原问题离散化后的数据数量级。对于任意主席树的节点即某棵线段树，其含义再次说明一下，存储的是原序列的某个前缀： $a[1]$ 、 $a[2]$... $a[k]$ ，其中 k 小于等于 M ，所以主席树节点 i 、 j 信息域 $data$ 相减得到的即为原序列在区间 $[i, j]$ 上的信息域 $data$ 。此过程时间复杂度为 $O(\log(M))$ ，。

总结：

由以上可知，主席树是一种特殊的线段树集合，属于可持久化线段树，具备线段树的几乎所有优势。此处可持久化我的理解就是可以充分利用历史版本，从而达到优化目的，故主席树更新、查找的时间复杂度为 $O(\log(M))$ ，且总的空间复杂度为 $O(M \cdot \log(M))$ ，这些对于复杂问题都是可以接受的。关于主席树的代码，结合具体题目会在后面的有所展示，欢迎关注！

由于写博客时间有限，难免有错误或不足的地方，欢迎斧正！



分享

收藏

纠错



c++游戏编程



导轨式升降机



笔记本电脑排名



tsc条码打印机

推广

推荐文章

- 1. [HBase Rest客户端使用](#)
- 2. [环形缓冲区的设计及其在生产者消费者模式下的使用（并发有锁环形队..](#)
- 3. [MongoDB 初见指南](#)
- 4. [聊聊并发（四）深入分析ConcurrentHashMap](#)
- 5. [ElasticSearch 运维](#)
- 6. [AWS系列：Dynamodb深度体验（1）](#)

我来评几句

请输入评论内容...

登录后评论

已发表评论数(0)

相关站点



[CSDN博客](#)

+ 订阅





热门文章

- 1. [给jdk写注释系列之_jdk1.6容器\(2\)-LinkedList源码解析](#)
- 2. [Redis](#)
- 3. [HBase Rest客户端使用](#)
- 4. [环形缓冲区的设计及其在生产者消费者模式下的使用（并发有锁环形队列）](#)
- 5. [MongoDB 初见指南](#)
- 6. [聊聊并发（四）深入分析ConcurrentHashMap](#)

分享本文



收藏到推刊

[创建推刊](#)

收藏

取消

已收藏到推刊！

推刊名(必填)

请填写推刊名

推刊描述

描述不能大于100个字符！

权限设置：☒ 公开 ☐ 仅自己可见

创建

取消



文章纠错

邮箱地址

错误类型

正文不准确 ▼

补充信息

提交

网站相关

- [关于我们](#)
- [移动应用](#)
- [建议反馈](#)

关注我们

[推酷网](#)

tuicool2012

友情链接

- [人人都是产品经理](#)
- [魔部网](#)
- [PM256](#)
- [品途网](#)
- [移动信息化](#)
- [行晓网](#)
- [Code4App](#)
- [智城外包网](#)
- [LAMP人](#)
- [安卓航班网](#)
- [虎嗅](#)
- [缘创派](#)
- [IT耳朵](#)
- [艾瑞网](#)
- [创媒工场](#)
- [雷锋网](#)
- [经理人分享](#)
- [市场部网](#)
- [砍柴网](#)
- [CocoaChina](#)
- [北风网](#)
- [云智慧](#)
- [我赢职场](#)
- [大数据时代](#)
- [奇笛网](#)
- [咕噜网](#)
- [红联](#)
- [linux](#)
- [Win10之家](#)
- [鸟哥笔记](#)
- [爱游戏](#)
- [投资潮](#)
- [31会议网](#)
- [极光推送](#)
- [Teambition](#)
- [Cocos引擎](#)
- [中文官网](#)
- [硅谷网](#)
- [leangoo](#)
- [更多链接>>](#)