# 静态可持久化线段树：

区间第k小问题:

给定序列A[0]到A[n-1] 和m次询问 (1<n<1e5,abs(A[i]) <1e9 )

每次询问区间[l,r]内第k小的数是多少？

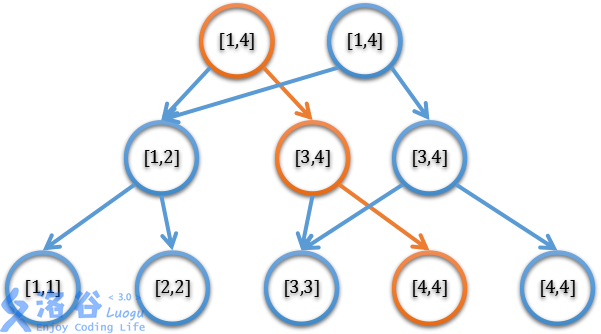
我们知道，求序列整体的第k小有2种常用方法，一个是平衡树，一个是权值线段树，具体原理参考数据结构的其他文章。在此权值线段树的基础上强化就诞生了可持久化权值线段树。

可持久化线段树也叫主席树是基于权值线段树的数据结构，权值线段树不说了，它是处理范围较大的离散数据需要离线做(大于1e6)，主席树也是一种离线数据结构。

先对所有数据排序 (指A[i])，用数组建立 <顺序,数值> 的索引。得到不同的数字个数up个,用up建树维护区间[0,up)。

## 属性：

主席树是一棵多根二叉树，直观的印象如下图 (有2个根时候的样子)：



需要注意的是，普通线段树有两种实现方式，数组实现和指针实现。

主席树不能像线段树那样，通过计算(now\*2和now\*2+1)得到左右儿子，只能像其他普通二叉树那样，维护左右儿子的指针了。再有就是主席树空间很大，线段树的左右端点信息如果不必要，尽量就省略不维护了。

树的节点属性有：ch[2]代表子树位置，sum代表区间和

此外还要维护root[]数组，root[i]代表第i个根的下标。

这种用数组维护可持久化权值线段树的根的数据结构，叫静态主席树。

实际上静态主席树的代码并不多，静态主席树不支持删除和修改操作

## **建树：**

利用线段树的建树方式，建立维护区间[0, up)的树，这颗树是空的sum=0，到这里还和权值线段树差不多。不要忘给root[0]赋值代表空树树根是root[0]

我们开始把每个A[i]插入刚才建的树里。

## 插入:

就是类似线段树单点更新，但是不去修改节点，而是建新的节点。

对于插入一个A[i],得到离散化后的数值x=ind[A[i]]，遍历上一棵树root[last],主席树里插入x,类似权值线段树插入x，看mid和x的大小关系，x<mid向左，反之向右，递归过程不修改节点，而是每层添加新节点，建立方式参考图中，相当于一次多出了logn个节点。

这里要注意从，每次插入操作，会建立新root，要记录下来。

## 查询区间第k小:

对于查询区间第k小问题，先把数据离散化，使得他们紧凑。

然后建立一个空的主席树，每个节点sum是0

对于插入操作，用主席树的插入操作去插入,插入的第i个数A[i]对应的那个根节点，就代表了了前i个数所建立的权值线段树。我们能利用这些权值线段树得到：前i个数的第k小(权值线段树查询总体第k小参考其他文章)。

对于查找前n个数的区间[x,y]第k小(x和y从1开始)

可以通过查询前y个数第k小和前x个数第k小得到。

方法：我们分别从root[x-1]和root[y]这两棵树的查找，对于各自每次走的节点，

设它们左儿子的sum属性的差值是d:

如果d>=k, 说明被找的数在左边，同时往左儿子找第k小

如果d<k, 说明被找的数在右边，同时往右儿子找第k-x小，

且我们的视线是同时从root[x-1]和root[y]下降的，要找到根节点一定同时到根节点，如果到了根节点，则返回这个节点的左端点值即可。

# 动态可持久化线段树：

区间第k小问题:

给定序列A[0]到A[n-1] 和m次询问 (1<n<1e5,abs(A[i]) <1e9 )

每次询问有两种:

* 区间[l,r]内第k小的数是多少？
* 修改A[i]的值

可见是在静态区间第k大问题里增加序列单个数字的修改操作

静态主席树没有任何修改操作，不能解决这个问题。

把静态主席树的root用普通数组维护改为用树状数组维护，用树状数组的一些性质来支持静态主席树单点修改，这种数据结构叫做动态主席树。

## 插入:

## 查询区间第k小:

前面为什么要一个个插入，保存历史节点。就是为了实现查询区间第k小，否则普通权值线段树只能查询总区间第k小，接下来是主席树最难理解的地方：

对于查找区间[x,y]第k小(x和y从1开始)

我们分别从root[x-1]和root[y]这两棵树的查找，对于每次走到2个节点，看它们左儿子的sum属性的差值x，如果x<=k,则同时往左儿子找第k大，否则同时往右儿子找第k-x大，且我们的视线是同时从root[x-1]和root[y]下降的，要找到根节点一定同时到根€€s节点，如果到了根节点，则返回这个节点的左端点值即可。