# 静态可持久化线段树：

它解决了区间第k小问题，给定序列A[0]到A[n-1] m次询问(1<n<1e5,abs( A[i])<1e9)

每次询问 l r k 询问区间[l,r]内第k小的数是多少

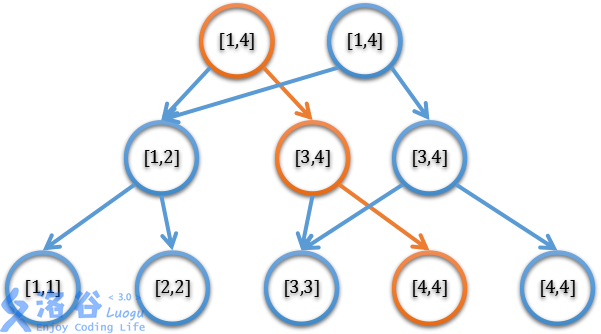
我们知道，求序列第k小有2种常用方法，一个是平衡树，一个是权值线段树。

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 顺序到数值 的索引。得到不同的数字个数up,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)：



主席树不能像线段树那样，通过计算得到左右儿子，我们要维护左右儿子的位置。再有就是主席树空间很大，线段树的左右端点信息就省略不维护了。

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。

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:

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。对于查找区间[x,y]第k大(x和y从1开始)，我们分别从root[x-1]和root[y]这两棵树查找，看两棵树差值x，如果x<=k,同时往左找，否则同时往右找。这是主席树难理解的地方。

# 动态可持久化线段树：

区间第k大问题里增加序列单个数字的修改操作，静态主席树只有插入操作，没有更新操作，不能解决这个问题。所以需要动态主席树

动态主席树就是把静态的root数组换成了树状数组，用树状数组的一些性质来支持主席树单点修改。