

左偏树

思想

我们希望两个堆能够合并，于是在普通二叉堆的基础上设计出了一种支持**高效合并**操作的数据结构——**左偏树（左式堆）**。

表示方式

因为涉及合并，所以一样用指针会好一点，但是在demo中，因为这是一道竞赛题，所以它用了数组表示。

定义

定义一棵树中一个节点到一个最多只有一个子节点的节点的最短距离为**零路径长**。对于左偏树而言，对于以每一个节点为根节点的子树而言，**左子节点零路径长都要大于等于右子节点的零路径长**。

合并操作

对于两个左偏树，将根节点键值大的树递归地合并到根节点键值小的树的右子树上。合并后有可能右子树的根节点的零路径长大于左子树，这时候将二者交换即可。递归的base case:其中一棵树为空树。

另：期望时间复杂度为 $O(\log n)$ 。

合并的思想：让短的子树生长！但是不能严格保证深度

demo luoguP4331题解

①先处理b序列要严格递增的问题：将a序列每个元素减去它的编号 ②首先考虑a序列如果单调递减，那么可以容易地说明 $b_1=b_2=\dots=b_n=x$ =a序列的中位数。然后将实际的a序列看作是由若干个递减序列构成的，求每个递减序列的中位数。若后一个递减序列的中位数小于前一个递减序列的中位数，那么将两个递减序列合并（**用到左偏树**），求它们的中位数（**具体方法：将合并得到的堆的根踢走，进行 $N/2$ 次， N 为树节点个数，剩下的堆的根就是合并后的区间的中位数**）。这个方法的正确性也容易证明。然后再把合并后的区间与上上个区间的中位数进行比较，直至后一个区间的中位数大于等于前一个区间的中位数。具体实现见

v3_finalversion.cpp

附

给定一个序列，求 $1 \sim i$ 的区间的中位数， $i=1, 2, \dots, n$ 。

方法：维护一个大根堆一个小根堆，确保大根堆的所有元素都小于小根堆所有的元素，同时维护两个堆的节点个数差距不超过1（方法：如果超过了，就从节点数多的堆上移几个节点到另一个堆上）。这样，中位数要么是大根堆的根，要么是小根堆的根，要么是两个根的平均值。