Leftist Tree（Leftist Heap）

左偏树（左倾堆）

描述：

左偏树是一种接近于堆的二叉树，它的根节点总是树中的最小值或最大值。两个小根堆（或大根堆）无法快速合并，而左偏树可以支持快速合并。本文只考虑根节点为最小值的左偏树。

左偏树的主要操作包括（1）合并两个左偏树；（2）在左偏树上插入新节点；（3）查找最小值（左偏树中根节点即为最小值）；（4）删除最小值（根节点）。其中（2）-（4）操作依赖于（1），因此合并操作是左偏树的核心操作。

定义左偏树中一个节点的距离，其值为该节点递归的向右下一直到叶子节点的边的数量。左偏树具有以下性质：

1. 父节点键值小于等于其左右孩子节点的键值，且；
2. 父节点的左孩子节点的距离大于等于右孩子节点的距离，；
3. 父节点的距离等于其右孩子节点的距离加1，；
4. 具有N个节点的左偏树的根节点的距离小于等于，；

如下图：



每个节点中上面的数字代表节点的下标号，下面的数字代表该节点的距离。

合并下图中的两个左偏树：



1. 比较两树根节点的值，节点7沿着节点6向右下寻找第1个满足的节点，替换作为节点6的新右孩子节点。该节点为节点8（），节点6的原右孩子节点8暂时脱离；



1. 节点8沿着节点7向右下寻找第1个满足的节点，替换作为节点7的新右孩子节点。该节点为节点12（），节点7的原右孩子节点12暂时脱离；



1. 节点12沿着节点8向右下寻找第1个满足的节点，替换作为节点8的新右孩子节点。该节点为节点13（），节点8的原右孩子节点13暂时脱离；



1. 节点13沿着节点12向右下寻找第1个满足的节点，替换作为节点12的新右孩子节点。该节点为节点26（），节点12的原右孩子节点26暂时脱离；



1. 节点26沿着节点13向右下寻找第1个满足的节点，节点13没有右孩子节点，因此节点26直接成为节点13的右孩子节点，不再需要替换，合并操作结束；



左偏树插入新节点的操作，可以看作左偏树与一个只有根节点的左偏树的合并操作；删除根节点的操作，可以看作删除根节点后，左右子树的合并操作。

左偏树的合并操作、插入节点操作、删除根节点操作的时间复杂度都为。