

左偏树

算法简介

一种可并堆，支持 $O(\log n)$ 的 push 、 pop 、 merge 操作。

算法思想

定义外节点为左儿子或右儿子为空的节点。 dist_i 表示节点 i 到其子树中最近外节点的距离，规定空节点的 dist 为 -1 。

左偏树的定义为对每个节点 u 均满足 $\text{dist}_{\text{lson}} \geq \text{dist}_{\text{rson}}$ 的堆。

左偏树有如下性质

- $\text{dist}_u = \text{dist}_{\text{rson}} + 1$
- $\text{dist}_u \sim O(\log \text{sz}_u)$

关于性质一，有

$\text{dist}_u = \max(\text{dist}_{\text{lson}}, \text{dist}_{\text{rson}}) + 1 = \text{dist}_{\text{rson}} + 1$ 证毕。

关于性质二，有 dist_u 表示节点 u 到其子树中最近外节点的距离，所以有节点 u 的子树的前 dist_u 层均为非外节点。

所以可将前 dist_u 层视为满二叉树，第 $\text{dist}_u + 1$ 层至少有一个外节点，所以有 $\text{sz}_u \geq 2^{\text{dist}_u}$ 证毕。

考虑左偏树的合并操作，类似 fhq treap 的 merge 操作，根据优先级(这里是点权)不断合并两棵树，遇到空结点返回。

不同之处在于合并两棵左偏树时对跳左节点还是右节点没有限制，所以强制每次跳右节点，使得 dist_u 单调递减。

这样，时间复杂度为 $O(\text{dist}_u) = O(\log n)$

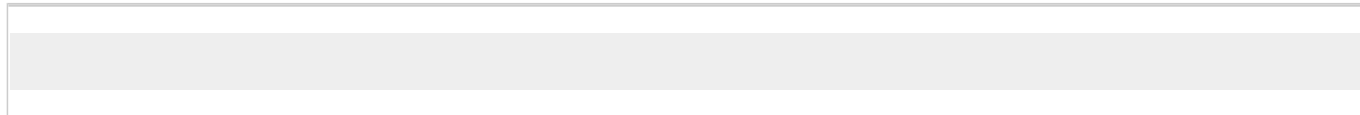
左偏树的另一个核心操作为寻根，事实上寻根操作可以利用路径压缩的并查集优化时间复杂度到 $O(\log n)$ 但需要注意一些细节，详细见代码。

代码模板

[洛谷p3377](#)

题意

题解



算法习题

习题一


[洛谷p1456](#)

题意

题解



From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:%E5%B7%A6%E5%81%8F%E6%A0%91&rev=1594270339 

Last update: 2020/07/09 12:52