

lct

我说这是我本命数据结构有人信吗(

前置知识

请务必理解：

[树剖](#)
[splay](#)

参考论文 [QTREE](#) 解法的一些研究

原理

树剖维护重链，而lct维护实边。

我们令一个子树的树根到子树中最后访问的点的路径是实边
则最后一次访问的点，到树根的路径，全部为实边
每个点到它的儿子中的边最多有一条实边

splay有很好的性质：能够低代价的合并 / 拆分树

我们对每一条实边用splay维护[]splay内部以(原树)深度为key (并不用显式的维护) fa[i] 维护 i 到 father[i] (原树上的) 的虚边 (实边的fa[i]随便)

access

关键操作是 access(i) 代表访问点 i，同时维护实边

1. splay(i)，把 i 转到根
2. 分离 i (splay 中的)右子树，即：将 i 指向儿子的实边置为虚边
3. get_leftmost，找出 splay 中最高的点 u
4. 将 fa[u] 的右子树分离 (同 1, 2) (将原有的实边转为虚边)
5. 将当前 splay 与 fa[u] 所在 splay 合并 (将边 (fa[u], u) 置为实边)
6. 重复以上操作，直到根

makeroott

换根操作 makeroott(i)

1. access(i)
2. 实际上，只有路径 (root, i) 上的父子关系发生了改变
3. 那么，只要区间翻转 splay 就可以了

cut

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



Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:no_morning_training:fayuanyu:lct

Last update: **2020/05/17 23:28**