

点分树

算法简介

点分治的动态版本，可以动态维护树上路径信息，一般会与线段树或平衡树等数据结构配合使用。

特别适用于一些需要维护路径长度相关信息的题目。

一般空间复杂度为 $O(n \log n)$ 单次查询或修改时间复杂度为 $O(\log^2 n)$

算法思想

把点分治过程中得到的重心连接，得到一棵虚树，该树有如下性质：

- 1、深度不超过 $\log n$
- 2、所有结点的子树大小和不超过 $n \log n$

由于点分治最多递归 $\log n$ 层，所以性质一成立。

对性质二，考虑每个结点对子树大小和的贡献。

每个结点对每个祖先结点(包括它自己)产生一个贡献，而由性质一，每个节点最多有 $\log n$ 个祖先结点，所以每个结点贡献最多为 $\log n$

所以所有结点的子树大小和不超过 $n \log n$ 性质二证毕。

有了这两个性质的保障，便可以用点分树暴力地解一些题目。

一般思路为对每个结点，维护经过该结点且在虚树上该结点是路径中深度最小的点的路径的信息

对每个结点，若使用线段树或平衡树等数据结构维护路径信息，由于路径数等于子树大小，根据性质二，空间复杂度仅为 $n \log n$

修改和查询都暴力跳 fa 若每次查询线段树或平衡树数据结构维护路径信息，根据性质一，有单次查询或修改时间复杂度为 $O(\log^2 n)$

算法习题

习题一

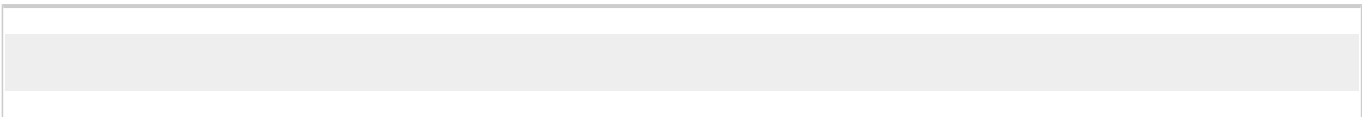
[洛谷p6329](#)

给出一棵 n 个结点的点权树 m 个操作。


操作0：输出所有到结点 x 距离不超过 k

操作1：将结点 x 点权修改为 y

Last update: 2020-2021:teams:legal_string:jxm2001: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:%E7%82%B9%E5%88%86%E6%A0%91&rev=1591790711
2020/06/10 点分树
20:05



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

Permanent link: https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:%E7%82%B9%E5%88%86%E6%A0%91&rev=1591790711 

Last update: 2020/06/10 20:05