

Synchronization 解题报告

绍兴市第一中学 任之洲

1 试题来源

JOI Open Contest 2012/2013

2 试题大意

给出一棵 N 个点的树，一开始每个点上都有一条不同的信息。

起初，所有 $N - 1$ 条边都处于关闭状态，现在有 M 个操作，每次操作会改变一条边的开闭状态。

在每次操作后，每个通过开启状态的边连通的连通块内的点会互相共享信息，也就是一个连通块内所有点的信息集合都会变成整个连通块的信息集合的并集。

有 Q 组询问，求在这 M 个操作全部进行完后，给定点上的信息集合大小。

2.1 数据规模与约定

$2 \leq N \leq 100000$, $1 \leq M \leq 200000$, $1 \leq Q \leq N$

- subtask1 [30points]: $Q = 1$
- subtask2 [30points]: 这棵树是一条链
- subtask3 [40points]: 没有额外的限制

3 算法介绍

3.1 算法一

先考虑 $Q = 1$ 的情况, 在这个subtask中, 只需要计算一个点上有多少信息。

这个问题比较别扭, 先考虑最终有哪些点上有这个点的信息, 并且求出该条信息到达每个点的时间。

可以从当前点开始DFS遍历, 需要计算通过每条边的时间。预处理出每条边在哪些时间区间是开启的, 在经过这条边时找到最早的一个能使用的。

将整棵树都遍历后, 也就计算出了哪些点最终得到了这个点的信息, 但是我们真正需要计算并不是这个问题, 不过可以类比一下。

同样从当前点开始遍历, 计算出每个点能到达目标点的最晚时间, 类似地, 需要找出的是最晚的一条能使用的边。

时间复杂度 $O(N + QM)$, 可以通过subtask1。

3.2 算法二

原问题是一棵树, 那么先来考虑相对于树的简化情况, 也就是树退化为一链, 一个点上的信息集合一定是一段区间。

对于每个点, 分别计算出左侧和右侧分别有多少点的信息可以传递到它, 下面介绍计算左侧点数的方法, 右侧可以相同得到。

参考算法一, 从左往右遍历, 维护到达这个点的时间为 t 的点数量, 经过一条边时需要计算对应的变动。

一条边的开启时间是一些不相交的区间, 容易发现, 经过一条边时可以计算出每个时间区间有多少点经过, 比如当前枚举到的时间区间为 $[l, r]$, 前一个时间区间为 $[l', r']$, 那么 $[r' + 1, r]$ 这段时间内的点是在这个区间通过的, 对应的变动只需要计算区间 $[r' + 1, l - 1]$ 区间内的点数, 将他们移动到 l 。

这些修改都可以用线段树来完成, 没有区间可以通过的点直接删除, 这样就计算出了每个点左侧有多少点可以传递到它。

时间复杂度 $O((N + M) \log N)$, 可以通过subtask2。

3.3 算法三

链的情况已经解决了, 树的情况只需要类比一下。

考虑进行点分治，每次处理经过重心的所有传递的贡献。

先计算出所有点到达重心的时间，经过边时的变动和算法二相同，合并不同子树的信息时可以用线段树合并¹，这一部分的复杂度为 $O((N + M) \log N)$ 。

所有点到达重心后，再用同样的方式从重心往下遍历，在遍历时涉及到了回溯操作，可以用可持久化来解决，复杂度也为 $O((N + M) \log N)$ 。

注意到这样直接做是会算重的，对于每棵子树需要再减去来自自己这棵子树的贡献，这一部分的问题规模和前一步相同。

时间复杂度 $O((N + M) \log^2 N)$ ，空间复杂度 $O((N + M) \log N)$ 。

3.4 算法四

还有另外一个截然不同的思路，考虑直接在连边和断边的同时在线维护每个点的集合大小。

同一个连通块中所有点的集合大小都是相同的，可以在选择每个连通块中深度最浅的点，定义为这个连通块的根，将每个连通块的答案存在对应的根上。

在断开一条边时，更新新分离出来的连通块的根上的信息，在断开的这条边上记录断边前这两个连通块的交集大小（即现在的答案），当下次连上这条边时，假如这两个连通块现在的集合大小为 A 和 B ，边上记录的交集大小为 C ，那么现在的并集大小为 $A + B - C$ 。

随便选择一个动态树数据结构就可以完成维护，比如Link-Cut Tree。

时间复杂度 $O((N + M) \log N)$ ，空间复杂度 $O(N)$ 。

¹线段树合并有 $O(N \log N)$ 的均摊保证，可以参考黄嘉泰WC2013的营员交流。