

Graph Challenge题解

姜晨耀

2015 年 11 月 9 日

1 题目大意

给你 n 个结点 m 条边，这 n 个结点的标号是利用DFS得到的DFS序生成的，即存在一种DFS方案使得标号和DFS序相同。定义 x 为 y 的supreme vertex即存在一条路径 $v_0 = x, v_1, v_2, \dots, v_k = y$ ，满足 $x < y$ 且对于所有的 $0 < i < k$ 都有 $v_i > y$ 。定义 x 为 y 的superior vertex即满足 x 是所有 y 的supreme vertex中结点编号最小的点。

询问每个节点有多少结点视其为superior vertex。

2 题解

我们可以求出来每个节点的superior vertex是谁，这样就可以统计出答案。求superior vertex参照了Editorial中给定参考论文，按照论文中的约定，我们记 $sdom(w)$ 表示 w 的superior vertex是谁。 $sdom(w)$ 的准确定义为，

$$sdom(w) = \min\{v \mid \text{there is a path } v = v_0, v_1, \dots, v_k = w \text{ such that } v_i > w \text{ for } 1 \leq i \leq k-1\} \quad (1)$$

稍后会证明 $sdom(w) < w$ ，所以上述式子和我们要求的目标是一样的。

2.1 $sdom(w)$ 的等价式

首先恢复原图，得到dfs的树边和非树边。

论文中给出了 $sdom(w)$ 的一个更方便计算的等价式，

$$sdom(w) = \min(\{v \mid (v, w) \in E \text{ and } u < w\} \cup \{sdom(u) \mid u > w \text{ and there is an edge } (v, w) \text{ such that } u \text{ is the ancestor of } v\}) \quad (2)$$

$u = v$ 也可以称 u 为 v 的祖先。我们记上式等号右边得到的结果为 x ，我们先证明 x 满足存在一个路径 $x = v_0, v_1, \dots, v_k = w$ 使得对于 $1 \leq i < k$ 满足 $v_i > w$ 。假如存在一条边 (x, w) ，则路径长度为1，一定满足条件， $sdom(w) \leq x$ 。另一种情况 $x = sdom(u)$ 存在一条边 (v, w) 且 u 为 v 的祖先。我们可以知道有一条路径 $x = v_0, v_1, \dots, v_j = u$ ，对于所有的 $1 \leq i \leq j - 1$ 都有 $v_i > u > w$ 。而树上的路径 $u = v_j, v_{j+1}, \dots, v_k - 1 = v$ 对于 $j \leq i \leq k - 1$ 满足 $v_i \geq u > w$ 。因此可以将这两条路径组成一条路径 $x = v_0, v_1, \dots, v_j, v_{j+1}, \dots, v_{k-1} = v, v_k = w$ ，对于 $1 \leq i \leq k - 1$ 均满足 $v_i > w$ ，所以这条路径合法， $sdom(w) \leq x$ 。

我们再证明 $sdom(w) \geq x$ ，这样就可以证明等式成立。 $v_0 = sdom(w), v_1, \dots, v_k = w$ 是一条简单路径。假如最优解中 $k = 1$ ，即存在一条 $(sdom(w), w)$ 的边，等价式考虑了所有的 v 当存在 (v, w) 时，所以 $sdom(w) \geq x$ 。假设最优解中 $k > 1$ ，那么我们找到一个最小的 j 使得满足 $j > 1$ 且 v_j 是 v_{k-1} 祖先。 j 可能等于 $k - 1$ ，所以这个 j 一定是存在的。

我们可以得到在 $1 \leq i \leq j - 1$ 中， $v_i > v_j$ 。可以利用反证法来证明，假如存在一个 i 使得 $v_i < v_j$ ，且 v_i 最小。此时 v_i 到 v_j 的路径中必定经过 v_i 与 v_j 的LCA，但是 v_i 是最小的，所以 v_i 与 v_j 的LCA为 v_i ， v_i 为 v_j 的祖先。但是这样就不符合我们选择 j 的条件了。

所以 $x \leq sdom(v_j) \leq sdom(w)$ 。联合起来这样我们就证明了 $sdom(w) = x$ 。

2.2 优化

得到等价式后我们就得到了一个暴力的计算方法，对于每个点寻找所有的相邻结点，再不断向根节点找最小的 $sdom(w)$ 。

假如我们从 N 到1计算 $sdom$ 。维护一个森林，当计算完 $sdom(i)$ 的值以后就把 i 加入森林中，并把在dfs树中的儿子的父亲设置为自己。这样每次查询相当于查询自己到自己所在树根的 $sdom$ 的最小值。整个维护只有link操作，所以可以用并查集直接做。总复杂度为 $O(n \log(n))$