树形图 题解

xiaolilsq

2024 年 7 月 20 日

题意简述

对于一个有向图，称一个点是根当且仅当它到其他所有点的简单路径唯一。
称一个点为一类点当且仅当它可以作为根。
称一个点为二类点当且仅当它不是一类点，且可以删除若干遍，在保证原来一类点仍然为根的前提下它也可以成为根。不同点是否为二类点的判断独立。
称一个点为三类点当且仅当它不是一类二类点。
给定一张 n 个点 m 条边的有向图，要求将其中点分类。
时间复杂度需要做到线性。

算法一

我会暴力枚举！
或者其余乱搞。
期望得分：5。

算法二

我会性质 A！
性质 A 保证没有一类点，所以判断二类点时可以任意删除边，注意删边只会让连通性变弱，所以一个点要是二类点必须要能够到其余所有点，然后我们直接保留它的 dfs 树上的树边即可让其成为根，由此这是充要的。
可以使用 Tarjan 算法缩点之后判断，但是更简单的办法是利用 Kosaraju 的思想，直接依次枚举每个点开始遍历图，如果有能够到达所有点的点，那么最后一个退栈的点一定可以到达所有点，然后在反图上 dfs 一遍即可找到整个可行的强连通分量。
结合算法一期望得分：15。

算法三

我会判断根！
考虑什么样的点是根，如果 r 是根首先要能够到达所有点，直接以 r 为根 dfs 出一棵树，考虑这棵 dfs 树上可能有返祖边、横向边、前向边三种非树边，注意到前向边和横向边只要存在立即出现矛盾，而非树边只有返祖边的时候归纳法证明 r 到其余任意点简单路径唯一，于是这是充要的。
由此我们直接枚举每个点 dfs 一遍即可判断它能否成为根，也就是找到所有一类点，性质 B 保证没有二类点，剩余的都是三类点。
于是可以过性质 B 的 $O(n^2)$。
结合算法二期望得分：25。

算法四

我会通过一类点找一类点！
在继续之前我们先证明一个小引理，如果 u 到 v 的简单路径唯一且经过了 w，那么必然有 u 到 w 简单路径唯一，w 到 v 简单路径唯一。
不妨假设 u 到 w 路径不唯一，设 $u \to w, w \to v$ 路径分别为 p_1, p_2，而 $u \to w$ 的另一条路径为 p_3，我们取 p_3, p_2 的交在 p_2 中最靠后的一个点，按照这个点把 p_3, p_2 截断得到一条新的 u 到 v 的路径。同理 w 到 v 路径唯一。

算法四

观察部分分发现特殊性质 C 给出了一个一类点，这启发我们用一个一类点去找到其它所有一类点。
以一个一类点 r 为根 dfs 得到 dfs 树，那么其中的非树边只有可能是返祖边。
一个点 x 要是一类点首先要能够到达 r，所以 x 子树内部必然有至少一条边可以走出整个子树，由于只有返祖边这条边必然是走到 x 的祖先处。
如果有多条边可以走出 x 子树，那么 x 到深度更深的那个点就至少有两条简单路径了，矛盾，由此 x 子树内部必然只有一条可以走出整个子树的边，下文简称为出子树边，出子树边连到的那个点简称为出子树点。

算法四

如果 x 子树内只有一条出子树边，且该边刚好连到 r，那么可以看出 x 也是一类点。
由此我们猜测：x 是一类点当且仅当 x 子树内部只有一条出子树边，且该边连向一个一类点。
必要性：x 有多条出子树边前面已经说过；x 有一条出子树边到了 y，如果 y 到 z 有多条路径，首先由一类点 dfs 树的性质知 z 不在 x 子树内部，由此 z 在子树外部，从而 x 到 z 的路径必然经过 y，和引理矛盾，如果 y 无法到 z 那么 x 也无法到 z。
充分性：我们直接把 x 到 r 的路径提取出来，然后观察树边和返祖边性质即可。
由此我们可以通过一个一类点找到所有一类点，可以过性质 B 与 C 的 $O(n)$。
结合算法三期望得分：35。

算法五

同样以一个一类点 r 为根建 dfs 树，找到所有一类点，由此所谓的判断二类点删边时需要保证一类点仍然为根就是要保留所有一类点为根 dfs 树的树边，对应在这棵 dfs 树上就是若干返祖边和所有树边，称这些边为必要边。
如果在只保留这些必要边的前提下一个非一类点 x 子树内部已经有多条出子树边那么 x 只能是三类点；如果只有恰好一条出子树边，那么必然连向一个一类点，由此 x 是二类点；否则我们看那些非必要边，那么 x 为二类点当且仅当 x 子树内部存在一条非必要边连出子树到达一个二类点或者一类点。
结合算法四，额外过性质 C 的 $O(n)$ 以及无特殊性质的 $O(n^2)$。
期望得分：55。

算法六

我会找一类点！
如果图中存在一类点，那么必然存在至少一个点能够到其它所有点，任取这样的一个点 r，以 r 为根建立 dfs 树 T_r。
T_r 中除了树边还可能有有返祖边，前向边，横叉边，其中横叉边和返祖边都是连到 dfs 序更小的点，由此出子树点 dfs 序必然更小。
我们考虑一个点 x 如何才能是根，首先它必须可以到达 r，找到 x 到 r 的唯一路径，如果在 x 到 r 的路径中走了非树边，那么我们称该非树边为关键边，且该边连接的下一个点为关键点，同时 x 也是关键点。
事实上这些非树边必须要是对于前一个关键点而言的出子树边，而关键点也是出子树点。
不妨将这些关键点按照经过顺序顺序排列为 $x = x_1, x_2, \dots, x_m = r$，那么 dfs 序在这条关键点列上面必然是递减的。
我们可以根据这些关键点将树分成 m 个区域，每个点被分配到它往上最近的关键点处，那么我们可以由此得到以 x 为根的一棵外向树 T_x，其中 T_x 的树边为所有关键边以及没有跨越不同区域的 T_r 中的树边。

算法六

首先对于每个关键点 x_i，要求其子树内部最多只有一条出子树边。
若有多条边，只有其中一条可以到达 r，不妨设另一条边为 $u \to v$，通过找到 x 到 v 的两条简单路径导出矛盾。
首先 $u \to v$ 在 T_x 上不是树边，只能是返祖边，从而有 T_x 上 v 是 u 的祖先，存在一条不经过 u 的 $x \to v$ 的路径。
然而出子树点的 dfs 序都是更小的，所以 v dfs 序比 x_i 小，v 不能出现在 $x \to x_i$ 中，从而 $x \to x_i \to u$ 必然不经过 v，从而 $x \to x_i \to u \to v$ 是另一条 $x \to v$ 的路径导出矛盾。

算法六

由 x_i 出子树边的唯一性知我们可以立即很好地唯一确定这 m 个关键点 $x_1, x_2, \ldots, x_m$：x_{i+1} 就是 x_i 唯一的出子树点。

接下来考虑 T_r 中的每条边对于 T_x 的影响（同样按照 m 个关键点将图分成 m 个区域）。

对于同一个区域内部：前向边或者横叉边不能存在，返祖边可以存在，树边显然合法。

对于跨越不同区域的非关键边：

- 如果是返祖边，由于其是非关键边，所以说明该区域关键点有至少两条出子树边矛盾；
- 如果是前向边或者横叉边，那么这些前向边和横叉边的终点必须落在 x 到 r 的路径上；
- 如果是树边显然合法。

上述所有条件事实上是充要的，接下来的难点是如何转化成比较好做的形式。

算法六

具体做法如下：

1. 任取一个可以到所有点的点 r，如果不存在显然没有答案，以 r 为根反向 dfs 建立 dfs 树 T_2，同时对于每个点规定其到父亲的边为特殊边（或者说，T_2 中的树边即为特殊边）。
2. 以 r 为根正向 dfs，其中对于每个点如果其有连出去的特殊边那么特殊边必须最后枚举，将这些点建立 dfs 树 T_1。
3. 首先要求 T_2 中的特殊边必须是 T_1 中的树边/横叉边/返祖边，如果不行该边在 T_2 中对应的整个子树都不合法。
4. 对于 T_1 中每条横叉边 $u \to v$，所有根必然在 T_2 中 v 的子树内。
5. 对于 T_1 中每条前向边 $u \to v$，所有根必然在 T_2 中 v 的子树内。
6. 若在 T_2 中某条边对应的是非树边，那么若这条边连过去的那个点在 T_1 子树内有多条出子树边，那么该边在 T_2 中对应的整个子树都不合法。同时所有根还需要保证其 T_1 子树内恰有一条出子树边。且上述路径必须和 T_2 中一致。

基本思路是条件 1235 共同保证区域内部没有前向边，且跨越不同区域的前向边终点落在 $x \to r$ 路径上；条件 4 保证区域内部没有横叉边，且跨越不同区域的横叉边终点落在 $x \to r$ 路径上；条件 6 保证关键点出子树边唯一。

算法六

若某个点 x 在以上条件中都判定为合法，则 x 必然可以顺着每次子树内部唯一出去的边一路走到 r，同样按照关键点的定义划分各个区域。

- 对于树边，若同一个区域合法；若不同区域按照关键点排列的规则合法。
- 对于前向边 $u \to v$，同一个区域不可能，具体而言 v 必然在 x 到 r 的路径上，而要 u, v 同一个区域必然有 u, v 属于同一个关键点，且该关键点通过 v 的某个儿子出去，而 $u \to v$ 要是前向边必须有该边是最后遍历的，这与条件 2 矛盾；不同区域则同样按照关键点排列规则合法。
- 对于横叉边 $u \to v$，必然有 $x \to v \to r$，则看 u 所在的关键点 w，若 w 是 v 的祖先，则为 $x \to v \to w \to r$，该横叉边合法，若 w 不是 v 的祖先，要么 $x \to w \to v \to r$，此时 w 内部唯一出子树边必然是 $u \to v$，该边合法，要么 $x \to v \to w \to r$，注意到 $u \to v$ 是 w 的出子树边，从而 w dfs 序大于 v dfs 序，而 w 是关键点，矛盾。
- 对于返祖边 $u \to v$，若 u, v 在同一个区域内部合法；若在不同区域考虑 v 上方的关键点，该边为出子树边，为关键边所以合法。

算法六

若某个点 x 在以上某个条件种被判定为非法。

- 若 1 中被砍掉，说明到不了 r 非法。
- 若 6 中被砍掉则不满足关键点出子树边唯一。
- 在 3 中被砍掉，说明它到 r 的路径上存在两个可能经过的点 u, v 间有多条路径非法。
- 若在 4 中被砍掉，那么说明 x 到 r 不需要经过 v，从而在不走 $u \to v$ 的时候可以找到一条 x 到 v 的路径，而同样可以找到一条 $x \to u$ 的路径并且不经过 v，从而再走以下 $u \to v$ 即找到两条 x 到 v 的路径。
- 若在 5 中被砍掉，则同样地说明 $x \to v$ 一条，还有 $x \to u$ 且不经过 v，再走一下 $u \to v$ 即找到两条 x 到 v 的路径。

算法六

综上结合算法五时间复杂度 $O(n)$。
期望得分：100。

算法七

我会随机化！
求求大家收了随机化这个神通吧！完全卡不掉啊！不会又有人随机化过了吧。。。
好，看起来没啥人写，随机化败！
期望得分：$0 \sim 100$。

吐槽环节