towns 解题报告

宁波镇海蛟川书院 卢啸尘

1 试题来源

IOI2015 Contest Day 2

2 试题大意

现有一棵边上带权的树。已知其叶子的数目。所有非叶子均至少有3个邻居。

每次可以查询一对叶子之间的距离。

要求求出：

1. 离最远叶子最近的非叶子，离最远叶子的距离。

2. 所有离最远叶子距离为这个值的非叶子中，是否存在一个非叶子，使得去掉这个点后，树的每个连通块包含的叶子数不超过总叶子数的一半。

3 数据规模

叶子数N在[6,110]。

允许询问ceil(3.5N)次。

4 算法介绍

4.1 算法框架

由于中心必定在直径上，下面的文章主要是针对直径来做。

从而这个算法可以被划分成两个相对独立的部分。

1. 求直径，求每个叶子投影到直径上后的位置和到直径的距离。

2. 对有疑问的非叶子进行判定。

我们注意到直径上的所有非叶子对应了至少一个叶子（度数条件），从而第一问的值可以在第一部分完成后直接求出。

考虑有两个非叶子满足条件的情况：断开这两个非叶子之间的边，如果两边的叶子数相同，那么这两个非叶子都是满足第二问条件的非叶子。否则，所在连通块中叶子数少的那个显然是不满足要求的。从而，最多只需要对一个非叶子进行判定。

这就是说，最后所需的询问次数是两部分直接相加。容易写出一个第一部分3N步，第二部分2N步的做法，下面将将其优化到2N+1.5N步。（非正常数已略去）

4.2 第一部分

4.2.1 3N

从任意点S，求其他点到它的距离。设最远点为A。从A点求其他点的距离，设最远点为B。则AB即为一条直径。

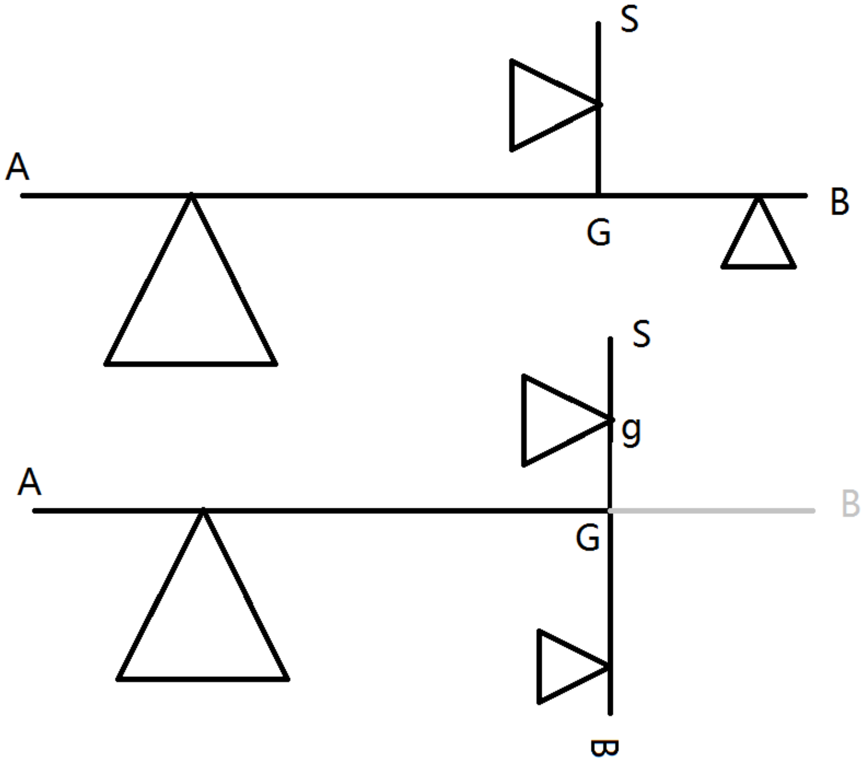
从B点求其他点的距离，则可以根据点p到A，p到B和A到B三个距离来算出所需的两个值。

4.2.2 2N

当S等于B的时候上面这个算法就是2N的。

当S不等于B的时候，S的投影G一定落在AB中靠近B的那一半。（考虑到SA不小于SB）

从而，如果我们需要验证一个落在靠近B端的非叶子，那么这个叶子不会落在GB(不含)上。因此可以把投影落在GB的那些叶子等效成落在G上。这就是说，只利用p到A和p到S的距离来计算，然后如果算出的投影g使得gA≥GA，则将gA修改成GA。GA可以通过SB 和SA求出，不需要额外的查询。



4.3 第二部分

4.3.1 2N

求一系列数的众数出现频率是否超过一半有一个经典算法：记录两个值back, cnt。back记录数的种类，cnt记录数的个数。将数x一个一个拿来处理。如果x=back或cnt=0，那么$x \leftarrow back, cnt \leftarrow cnt + 1$，否则$cnt \leftarrow cnt - 1$。我们可以看到，如果有一个数出现频率超过一半，最后的back一定是这个数。从而当cnt=0时没有这样的数，否则back是唯一备选。

那么这个算法就是一个N扫一遍，一个N验证。

4.3.2 1.5N

对2N做法进行以下优化：当cnt=0，不进行比较直接$x \leftarrow back, cnt \leftarrow cnt + 1$；记录下扫描时每个点的情况，验证时跳过那些cnt≠0并且x=back的点，他们的贡献加到back上。

考虑每个$x \leftarrow back, cnt \leftarrow cnt + 1$，要么是前一种情况，在扫描过程中节省了一次查询，要么是后一种情况，在验证过程中节省了一次查询。而我们知道cnt不是负数，所以这两种情况的总和至少为0.5N，也就是说至少能节省下0.5N，从而总查询次数的上界是1.5N。

5 关于比赛时的情况

我在比赛时先后有25,35,48,61,100的提交。

25分的提交是只做第一问3N。

35分的是特判了度数非1即3的情况，也是不需要实现第二部分的。

48分的是3N+暴力平方。

61分的是3N+2N。

100分的是2N+2N，但由于数据弱通过了。使用一个含有奇数个点的星型树就可以卡掉我的程序。