

1 Cuvelia

考虑两个点的情况，合法的点是两点中点以及中点出发不往两个点的子树中走能到的所有点。

新加入点时，如果原中点对于新点不合法，我们只能把中点向新点的子树里挪动才能维持平衡，否则怎么挪都会不合法。所以可以用一个 *set* 维护当前所有点的 DFS 序来判断能不能走下去，然后倍增一下就行了。

处理完所有点以后把没有询问点的子树大小加起来就是答案。

2 树上四次求和

不妨把 $dis(i, j)$ 转化为 $dep(i) + dep(j) - 2 * dep(lca(i, j))$ 。

设 f_i 为询问 i 的答案，我们考虑增量， f_i 从 f_{i-1} 过来时多了右端点为 i 的贡献，而右端点为 i 的贡献又是右端点为 $i-1$ 的贡献加上点 i 和前面的距离乘上缺少的次数。我们发现缺少的次数是不变的，所以就是要求 $\sum_{j=1}^i dep(lca(i, j)) * j$ 。这是一个经典问题，可以使用树链剖分在 $O(n \log^2 n)$ 的时间内求解。如果使用 lct 可以做到一个 \log 。

3 Cubelia

一个子段的最大前缀和位置显然就是原序列最大前缀和数组中区间最大值的位置。那么题目可以转换为求所有连续子序列的最大值的和，减去多出来的前缀和。多出来的部分可以 $O(n)$ 预处理。

然后你会发现这就是 [HNOI2016] 序列 数据加强版 加强版。kcz orz

看这个 <https://loj.ac/article/489>。