树上GCD 解题报告

杭州学军中学 金策

1 试题来源

原创题，出给UOJ Round 2作为C题。

2 试题大意

有一棵n个结点的有根树T。树上每条边的长度为1。用$d(x,y)$表示结点x,y在树上的距离，

对于两个结点$u,v(u \neq v)$，令$a=\mathrm{LCA}(u,v)$，定义$f(u,v)=\gcd(d(u,a),d(v,a))$。对于所有$i \in \{1,2,\cdots,n\}$，求出有多少对$(u,v)$，满足$f(u,v)=i$。

3 数据规模

部分分数据请参阅原题面。

对于所有数据，$n \leq 200000$。

4 算法介绍

4.1 10%做法

枚举两个点u,v，用$O(\log n)$时间计算LCA和gcd，并更新答案。

复杂度$O(n^2\log n)$。

4.2 30%做法

随机生成的树的树高为$O(\log n)$。

枚举u，计算以u为LCA的所有点对的答案。计算时对u的子树DFS，统计出每个深度h上的结点数量$cnt[h]$。枚举深度h_1, h_2，并将$cnt[h_1] \cdot cnt[h_2]$累加到$ans[\gcd(h_1, h_2)]$中。注意还需把同一个子树内的多余计算给减掉。这样，统计答案的复杂度是$O(deg[u] \cdot \log^2 n)$，总和是$O(n \log^2 n)$。另外，每个点最多被DFS到$O(\log n)$次，所以DFS的总复杂度是$O(n \log n)$。

于是总复杂度$O(n \log^2 n)$。

4.3 40%做法

我们可以把问题转化为，对于每个d，求出$\gcd(h_1, h_2)$是d的倍数的点对的数量。然后用一个$O(n \log n)$的容斥算出最终答案。

第4个数据点中，从根结点挂下来若干条链，长度分别为$h_1, \cdots, h_k$。u, v属于同一条链的情况可以方便计算；u, v不在同一条链上时，LCA即为根结点。对于某个d，第i条链中高度为d的倍数的点有$\lfloor h_i/d \rfloor$个；我们要统计k条链中选出两条来的乘积总和，可以利用恒等式$(x_1 + \cdots + x_k)^2 = x_1^2 + \cdots + x_k^2 + 2\sum_{1 \le i < j \le k} x_i x_j$求得。

复杂度$O(n)$。

4.4 100%做法

考虑点分治，每次取出当前树的中心c。考虑所有$u \to \mathrm{LCA}(u, v) \to v$的路径经过$c$的点对$(u, v)$所产生的贡献，共有两种情况：

（1）u, v均在c的子树内，此时LCA即为c。和40%做法类似，只是第i条链中高度为d的倍数的点的数量需要在处理出cnt数组后花h_i/d的时间统计，所以复杂度是$O(n \log n)$。

（2）u在c的子树内，而v不在。我们把当前树的根节点记为$root$，$father[c]$到$root$的路径为$father[c] = a_1, a_2, \cdots, a_{k-1}, a_k = root$。记$c$下面的子树高度为$H$，$a_i$旁边伸出的子树（即不包含$a_{i-1}$的)的高度为$h_i$。枚举$a_i = \mathrm{LCA}(u, v)$，那么$v$位于$a_i$旁边的子树中，然后我们仍然枚举$d$，用$h_i/d$的时间求出$a_i$子树中高度为$d$的倍数的结点数量；但我们还需要知道$c$ 的子树中有多少点相对于a_i的高度是d的倍数，也就是说我们要在c子树的cnt数组中查询下标间隔为d的子序列中的元素之和。注意到间隔为d时，至多只有d种这样的子序列，我们对重复查询进行记

忆化。于是对于$d < \sqrt{H}$，查询的复杂度不超过$d \cdot H/d = H$，总共是$\sqrt{H} \cdot H$；对于$d > \sqrt{H}$，单次查询的复杂度为$H/d < \sqrt{H}$。

所以一层分治的复杂度是$O(n\sqrt{n})$的。根据主定理第(3)种情况，总的复杂度就是$O(n\sqrt{n})$。

5 总结

这个题目需要一定的代码量，不过比那些烂大街的码农数据结构题还是要高明得多的。首先它需要在在有根树上进行点分治，这样的题目并不是很多（印象中有NOI2014的第二天第三题）。因为有根，所以要对重心的子树以及不在子树内的结点分开处理。之后实际上是要做一个下标为等差数列的查询，这个东西感觉很难做到比$O(\sqrt{n})$更好的复杂度。不过比较好的一点是这样的分治是不会多log的。vfk当时提出了一个用启发式合并的做法，代码比较短，复杂度是$O(n\sqrt{n}\log n)$。