

题目比较散所以做一个题目汇总。

Code: https://github.com/RyanMuQ/Tree-partition_Path-on-Tree

POJ-1741[tag:树分治]

- Q :
 - $n(≤10000)$ 个点的带边权的树，求树上的距离 $≤k$ 的点对有多少个。
- A :
 - 点分治，找重心作为根。
 - 合并的时候先计算每个子树到其与大树重心相连的那个点的距离，大树答案即为子树答案，以及 LCA 为重心的答案之和。
 - 由于求后者的时候需要使用归并排序，所以 $f(n)=n\log n$ ，由主定理，总体复杂度 $n(\log n)^2$ 。

SPOJ-COT[tag:主席树，树上路径查询]

- Q :
 - $n(≤100000)$ 个点的带点权的树， $m(≤100000)$ 次询问 $u \rightarrow v$ 路径中的第 k 小。
- A :
 - 首先，若 $LCA(u,v)=u$ (或 v)，则想到可以用主席树造出每个点到根节点的点分布情况，相当于根节点到每个叶子的路径均可以对应“第 k 大的树”题目中的那个数组，而 dfs 的回溯过程很好处理，于是 $\log n$ 查询第 k 大。
 - 若 $LCA(u,v)=w$ ，则为路径 $w \rightarrow u$ 和 $w \rightarrow v$ 路径的合并。很快可以想到二分一个数，查询其在两个路径中的大小情况， $(\log n)^2$ ，被卡掉了。
 - 想到主席树查值过程其实就是二分过程，于是还可以查询四个主席树，变为 $\log n$ 。
 - 这个事情可以延伸为，一个数组，查询其任意个区间中所有数的第 k 大。

SPOJ-QTREE[tag:线段树，树链剖分]

- Q :
 - $n(≤10000)$ 个点的带边权的树，若干次操作，“Change”为改变一条边的权值，“Query”为询问从 u 到 v 路径上的边权最大值。
- A :
 - 树链剖分水题，用线段树记录重链上的情况，Query 的时候合并路径答案。

SPOJ-QTREE2[tag:LCA]

- Q :

- $n(\leq 10000)$ 个点的树，边的编号从 1 到 $N-1$ ，若干次询问，“DIST” 询问从 u 到 v 的距离，“KTH” 询问从 u 到 v 路过的第 k 条边。
- A :
 - 都用 LCA 就够了。

SPOJ-PT07J[tag:DFS 序, 主席树]

- Q :
 - $n(\leq 10^5)$ 个点的带点权的树， 10^4 次询问以 x 为根的子树第 k 大（总根结点为 1）
- A :
 - DFS 重调之后转为区间无修改第 k 大(小)问题

SPOJ-QTREE5[tag:树分治]

- Q :
 - $n(\leq 100000)$ 个点的树，点有两种颜色（黑/白）。若干次操作，“0 i ”为反转点 i 的颜色，“1 v ”为询问到 v 点最近的白点的距离。
- A :
 - 树分治，点 v 沿分治中心一个个上升的过程即为枚举经过哪个分治中心的过程。每个分治中心存一个 multiset 代表该分治树内到该点的距离的集合。修改颜色时依次将该点的距离拿进或拿出，查询时依次查询各个最小值。

SPOJ-QTREE6[tag:数链剖分]

- Q :
 - $n(\leq 100000)$ 个点的树，点有两种颜色（黑/白），初始为黑色。若干次操作，“0 u ”查询与到 u 路径上均为同色的点的个数，“1 u ”为反转 u 点的颜色。
- A :
 - 树分治可做。熟练剖分做法是，在每条链上开线段树，存 $No.i \sim No.j$ 的颜色数量和，以及 $No.i \sim No.j$ 上所有点沿轻链下行，相连的黑点总数和白点总数。query 时上行到相连块内 depth 最小点，再查询到链上相连块的 depth 最大点，区间和即为答案。两次查询均需要线段树二分。修改时要一路上行，修改到与根结点所在链。