# 虚树

## Bzoj2286

学习了一下虚树，讲一下自己的理解。

       虚树是这么一个东西，对于需要求答案的点p，只保留对答案有影响的节点，从而减少时间。

       现在对于这道题目进行特定的说明。

       考虑朴素的dp，显然，dp[i]=min(val[i],Σdp[j](j为i的儿子))，val[i]表示将i和根节点分离的代价。那么这样的时间复杂度为O(N)，总时间复杂度O(NM)。

       注意题目中有Σk<=500000，因此如果能将一次的时间复杂度减小到O(K)或者O(KlogK)，就能通过了。因此，关键是能构造出一颗节点<=O(K)级别的虚树，以及能在O(K)或者O(KlogK)的时间构造出虚树。

       定义某一次询问给出的岛屿为关键点。注意到对于某对关键点(x,y)，考虑x->lca(x,y)的路径中，没有那个点是某一对关键点的lca，那么显然x->lca(x,y)的路径上的点对答案不会产生任何影响，换句话说将lca(x,y)->...->x的路径直接压缩成lca(x,y)->x，对答案不会产生影响。因此我们只需要保留所有关键点，以及它们两两之间的lca，然后按照原数的祖先关系连边，在构造得到的虚树上面跑dp即可。注意到k个点两两之间不同的lca只有k-1个，因此产生的虚树是O(K)的。

       下面来构造，首先按照在原树上的dfs将岛屿进行排序。

       用一个栈维护从根节点到栈顶的路径上，需要加入虚树的点形成的一条链。注意不能直接将链中的点连边，因为有可能会有新的关键点的lca也在这条链上。考虑现在新加进来一个点x，以及栈顶的点p，t=lca(p,x)，有以下几种情况。

       1.t=p，那么把x加入栈就好了；

       2.t!=p，那么显然t是p的祖先，那么显然栈中t->p的路径上面的点都可以加入虚树了，因为不可能会有新的lca插入到t->p的路径中来。

       考虑2的具体实现，只要去除栈中的第二个元素，不断比较是否是t的祖先即可。