题意大意

你有一棵n个点的无根树，节点编号从1开始，和一个初始为空的可重集合S。有m次向S中加入元素的操作，第i次操作给出原树中的k_i条边，保证互不相同，你需要将这k_i条边删去，得到k_i+1个联通块，设构成这些联通块的点的编号集合分别为$T_{i,1},T_{i,2},...,T_{i,k_i+1}$，你需要将这些集合加入$S$，注意每次操作是互相独立的。

在所有m次加入操作后，有q次询问，第i次询问给定树上一个点u_i和一个半径r_i，对于所有原树上到u_i简单路径边数小于等于r_i的点的编号构成的集合C_i，你需要求出S中有多少个元素是C_i的子集。

数据范围

Subtask	n	$m+\sum k,q\le$	**特殊性质**	**分值**	**子任务依赖**
1	500	500		10	
2	2000	2000		10	1
3	10^5	10^5	第i条边连接编号为$i,i+1$的两个点	15	
4	10^5	10^5	$\forall i\in[1,m],k_i=1$	25	
5	10^5	10^5		35	$2,3,4$
6	10^5	3×10^5		5	5

对于所有测试点，都满足 $1\le n\le 10^5,1\le m+\sum k,q\le 3\times 10^5$，$\forall i,0\le r_i\le n$。

时间限制：2s

空间限制：1G

解题过程

算法一

直接模拟题意，期望得分10。

算法二

在算法一的基础上使用bitset加速判断子集过程，期望得分20。

算法三

对于树形态为一条链的情况，联通块的包含关系等价于区间的包含关系，此时使用二维数点可以解决形态为链的子任务，期望得分15。

算法四

由于树上一个点到一个点集中的最远点一定是这个点集的直径两端点之一，那么可以将一个邻域包含一个点集等价为同时包含直径两端点，即包含点集的直径。这个结论较为普遍，证明可以考虑下图：

```
u---h---y---v
    |   |
    w   x
```

其中u,v,w属于同一个点集中，且点集直径端点为u,v两点。考虑反证，假设一点x到点集中的最远点为w，则可以得到w,h间距离大于u,h间距离，从而得到一条端点为w,y两点的更长的直径，矛盾。

邻域包含一条路径可以继续转换为关于到这条路径中点的距离的限制。使用换根法预处理出每个子树及每个子树的补集的直径后套用点分治求解答案即可。结合在数据规模较小时暴力求直径，期望得分45。

算法五

对于k大的情况，有若干种求解直径的方法。出题人的做法是考虑dfn，dfn在删去给定的k条边后被划分为了$O(k)$个连续段，对于每个连续段，由上文性质可以得出点集合并后的直径端点只可能在原有两个点集的四个直径端点中，那么可以使用ST表预处理回答询问。找出这些连续段只需要将给定的k条边按照dfs时第一次被访问的顺序排序，然后在dfn上模拟dfs并维护一个栈表示当前还未回退的被删去的边，那么每个点按照dfn序被遍历到时所属的联通块就是当前栈顶的较深端点的联通块，由于总共有$O(k)$个连续段，暴力处理每个连续段就可以保证效率。

最终我们在两个log的时间复杂度内解决了本题。值得注意的是，虽然时间复杂度瓶颈在于点分治，求LCA的步骤仍有较大的时间开销，原因在于连续段数和每次合并直径的询问次数都有不小的常数，使用一个较快的LCA算法可以改进算法效率。根据常数及实现，期望得分$95\sim 100$。

算法六

仔细分析点分治的时间复杂度，发现瓶颈在于每次求解时需要对询问点的半径相关参数排序。但是注意到这个参数每次只会加上或减去其到分治中心的距离，如果我们提前对原本的询问半径排序并在递归时保持该顺序，那么只需要快速完成每个数加减一个有限值域内的数后的排序即可。具体来说，我们有一个长为len的有序数组，现在将每个数加上一个$O(B)$的整数，我们希望在$O(len+B)$的时间内对新数组排序。我们可以参考基数排序的做法，分别对元素除以B下取整和对B取模的余数排序：前者由于变化$O(1)$，可以转化为$O(1)$个有序数组的归并；后者直接桶排即可。处理得两个数组后，直接用类似基数排序的方式合并即可。再配合预处理$O(1)$LCA，可以得到理论一个log的解法，期望得分100，但是由于常数原因在实际表现上并没有显著快于直接实现的点分治。

参考资料

本题与程乐平同学有讨论交流。