Solution:

1.由乃：

线段树+倍增+搜索

利用鬼畜结论，询问区间过大直接输出 Yes;然后我们只需要考虑询问区间长度<=13，我们首先需要求出区间中的数是那些，区间立方貌似根本不好维护，但是我们发现立方之后%了 v，而 v<=1000，那么这些数一直在 v 以内变动，我们可以用倍增数组 f[i][j]来表示 i,立方 2^j 次之后的数是什么；因为区间<=13 我们可以暴力一个个单点查询；那么我们只需要维护每个位置被立方了多少次，这个可以用线段树就能很好的维护了；把这<=13 一个个求出来之后，我们搜索的复杂度为 3^13，因该是跑不过的，但是利用人类智慧随便加一点剪枝和调整搜索顺序就可以爆搜跑过而不用双向搜索；

2.树

Map+前缀和乱搞；

因为我们只能从深度小的往深度大的走，那么我们需要对从根出发的每条链单独处理；我们记链上的距离前缀和为 dep[]，那么对于利用前缀和知识，相当于对于 j,查询 dep[j]-dep[fa[i]]=s 的 i 的数量有多少；我们移项 dep[j]=dep[fa[i]]+s，那么我们可以每搜一个点，就把 dep[fa[i]]+s 丢进 Map 中，然后每个点都查一下 Map[dep[i]]，因为 Map 维护的是树上一条路径的信息，那么我们在 dfs 回溯的过程中需要把该点的信息删除；

3.This problem is too simple

树链剖分+动态开节点线段树

这个题和 Sdoi 2014 旅行的做法几乎一模一样；我们当然希望对于每一个不同的权值都开一棵线段树，然后我们查询的话就直接用树链剖分在该权值的线段树上查询即可；但是直接开是会爆空间的，所以我们用动态开节点线段树(传一个&)的操作来现就不会 MLE 了;

4.oi 树

倍增

考虑到每个节点上对于每个字符的移动方式都是唯一确定的，那么我们可以通过倍增来优化每一次暴力一直模拟的操作，因为对于不同字符的移动方向不同，那么我们可以用 f[i][k][j],表示从 i 出发，需要去的是第 k 个字符，走了 2^j 步后到达的点，然后就同正常的倍增题一样的做

5.二分图

线段树维护 CDQ(按时间分治)+带权并查集

考虑到 CDQ 分治的结构类似于一棵线段树，我们可以定义 Solve(x,l,r),表示只处理[l,r]是[st,ed]的子集的边，同时当前在线段树的 x 号节点；然后我们在线段树上的每个节点上开一个动态数组，表示能一直影响该节点所管辖的区间的边，相当于一个不下放的 lazy 数组；我们可以用类似于线段树区间查询的方法，把每一条边的影响划分为 log 个区间，存入线段树的 log 个节点中；然后我们考虑如何判断图中是否有奇环，一个经典的做法就是带权并查集，维护每个点到根节点路径的奇偶性
dep[x],dep[x]^dep[y]就是 x-y 路径的奇偶性，因为 dep[lca(x,y)]被 xor 两次之后就抵消了，所以得到的就是 x-y 路径的奇偶性；
我们对于分治的每一层，就把对应的线段树的节点的 vector 中的边依次加入图中，然后判断是否能产生奇环，注意产生奇环之后就可以停止递归了，因为往下的区间中必然也会出现该奇环；然后由于递归回溯的时候我们需要撤回之前加的边，我们可以用类似最小公倍数那题的做法，搞一个带撤销的并查集，具体做法就是用栈来记录每一次合并前的状态，然后从后往前撤回，至此，该问题得到了很好的解决；

6.情报传递

离线+树链剖分+树状数组

考虑到查询路径上危险值>C 的点貌似不好搞，要树套树或者可持久化；我们可以把询问按照 i-C 进行排序，把节点依次标记为 1，那么可以保证在询问时的所有标记过的点的危险度都>C；那么我们相当于只有单点修改和区间查询的操作；

7.消耗战

虚树入门题

考虑到∑ki 为线性，我们用虚树来解决这一问题；虚树的核心构建方法就是维护右链，具体实现方法是用栈，构建出的虚树只含有关键点和一些关键点的 lca;具体操作就是把关键点按照 dfs 序排序，从小到大插入栈中，然后对于每个点比较和栈顶的 lca 的关系分情况讨论来弹栈并连边；

具体构建方法：http://www.cnblogs.com/qt666/p/7481862.html

8.Peaks 加强版

Kruskal 重构树+主席树+树链剖分

考虑到做 Kruskal 的过程是，把两个点合并时，新建一个虚拟点，点权为边权；那么这样有一些性质：

1.原树点都是叶子结点。

2.子结点比父结点点权小(大根堆)。

3.原树与新树两点间路径最大值与新树相等，且等于新树两点 lca；

所以只要从 v 点往上跳到深度最浅的<=x 的点，用链剖+二分实现，然后查询子树第 k 大即可，查询子树 k 大用主席树实现;

9.上帝与集合的正确使用方法

欧拉降幂大法+递归

发一个链接：http://www.cnblogs.com/mmlz/p/4308314.html

10.Savage

暴力枚举+exgcd 检验；

因为保证了 m 很小那么我们可以枚举 m，然后对所有的野人两两 check 一下；

如果会相遇的话,假设相遇的年份为 x 则

(p[i]*x+c[i])%M=(p[j]*x+c[j])%M;

整理之后 (p[i]-p[j])*x+M*y=c[j]-c[i],那么我们只要判断方程无解或者最小整数解至少大于一个人的寿命即可；