1.Spy

把$\{a_i\}$、$\{b_i\}$分别写成列向量 A、B，暴力翻译题设得转移矩阵 F，则有 B = F * A，那么显然能得到一个 $O(N^3)$的高斯消元做法。

考察(i or j) xor i，原式 = (not(i or j) and i) or ((i or j) and not i) = (not i and not j and i) or ((i and not i) or (j and not i)) = not i and j。

现在我们考虑 n 位二进制数 F(not i and j)的值，若用行表示 i、列表示 j，就可以用一个 $2^n * 2^n$ 的矩阵 F_n 表示所有的值，利用数学归纳法我们有：

$$F_1 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

$$F_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$F_{n+1} = \begin{bmatrix} F_n & \text{not}\, F_n \\ F_n & F_n \end{bmatrix}$$

用类似的方法我们亦可以发现，若令：

$$G_1 = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$G_{n+1} = \frac{1}{2}\begin{bmatrix} G_n & G_n \\ -G_n & G_n \end{bmatrix}$$

则 $F^{-1}{}_n$ 只在矩阵的最右上角一个元素比 G_n 少 1。

若用该法得到 F 的逆矩阵并应用矩阵乘法，则得到一个 $O(N^2)$的算法。

根据分块矩阵的乘法法则，很容易能把 $F^{-1}{}_{n+1}$ * B[p ... q]的结果用 $F^{-1}{}_{n}$ * B[p ... mid]和 $F^{-1}{}_{n}$ * B[mid ... q]拼出来（方括号内左闭右开，mid 为 p、q 的均值）。不难看出这与 FFT 的原理十分相似，仿照 FFT 即得 O(NlogN)的正解。

2.Encrypt

各种姿势水平的暴力可以得到不同的部分分；满分算法可以利用后缀数组或后缀自动机，这里只提一下复杂度更优的 SAM。

我们每次询问者是 i 后缀与 0 ~ i - 1 后缀分别取 LCP 的最大值是多少、由哪个后缀得到。当我们询问 i 时，先尝试匹配（这里只需要按顺序不断检查下一个元素是否存在并转移）、然后在后缀自动机中扩展字符串，直到失配，其正确性显然（注意需在 SAM 的结点上记录它所对应的子串在字符串上的最左出现位置）。之后移动指针后再重复上述过程即可。

时间复杂度：O(N)。

3.Network

$O(N^2\log N)$以及 $O(N^2)$的暴力可以拿到很多部分分，配合利用最大公约数已为 1 后继续扩展必然还为 1 进行剪枝。

令 f(i)为树上边权为 i 的倍数的路径的条数，根据莫比乌斯反演我们有：

$$Ans=\sum_{i=1}^{10^6} f(i)\cdot\mu(i)$$ 。

莫比乌斯函数容易用欧拉筛法求出，现在我们求 f(i)。我们只需要考虑所有边权为 i 的倍数的边，因而我们建出一张点不变但只有这些边的新图。可以通过并查集维护连通分量来计算这些边在图中形成的路径条数。若每次修改后重建，代价是很大的。

对于本题来说，修改的次数有限，因而我们可以把和修改无关的边先用并查集维护，对于每次修改（含初始状态）我们考虑在这种状态中出现在该图的边后再删除，用并查集启发式合并来支持还原并查集。这样做我们的复杂度为 $O(S(N + Q^2)\log N)$，S 为权值平均因子数，可以通过所有数据。