T1

定义记号 $log_a(x) = t$ 表示 $a^t \equiv x \pmod p$

我们考虑将两个数 u, v 的大小关系定义为 $log_a(u), log_a(v)$ 的大小关系，比较两个数 u, v 时，如果 $u \times v^{-1}$ 不存在于序列中，说明 $log_a(v) > log_a(u)$ ，也就是说 v 应该排在 u 后面。

但这样做有个问题，就是 a^{-k} 可能在序列中。

因此我们要把所有逆元也存在于序列中的数从序列中去掉，然后得到一个新的序列。由于新的序列中的任意一个元素的逆元都不存在于序列中，因此 a^{-k} 就必然不在序列中。使用重定义的比较符号来对新的序列找最小值即可。

$n = 2, a = p - 1$ 时新的序列会为空，要注意特判。

T2

答案的配对方式肯定是一段前缀和一段后缀按顺序进行匹配。

随便维护一下即可。

T3

令 f_i 表示现在在第 i 格，跳到"往左走若干格后能跳到的最右侧的点"经过的挂的编号。

g_i 表示现在在第 i 格，跳到"往左走若干格后能跳到的次右侧的点（不严格）"经过的挂的编号。

每次查询时 ban 了一个挂，所以每次跳的边要么是 f 要么是 g。只要先倍增跳 f，当要跳的 f 边被 ban 了时就倍增跳 g ，直到 f 不再被 ban 然后就接着倍增跳 f 即可。

T4

记 f_i 表示以 i 为右端点，最小的左端点满足答案为 Yes。

显然 f_i 是单调递增的。

我们考虑给每一位开一个并查集，第 i 个并查集会从 i 开始不断添加标号递减的边，直到不能再添加时就求出了 f_i。（先不考虑时空复杂度问题）

考虑先求出 f_n，直接从第 n 条边开始按编号递减顺序加入并判断即可，由于 $\forall i < n, f_i \le f_n$，所以 $\forall i \in [f_n, n]$ ，我们可以直接在 $[i, n]$ 这个区间加入边 i。

接着考虑求出 f_{n-1}，方法类似，从第 $f_n - 1$ 条边开始按编号递减顺序加入并判断即可。
$\forall f_{n-1} \le i \le f_n$，我们依然可以直接在 $[i, n-1]$ 这个区间加入边 i。

正解即为用线段树分治维护上述过程，复杂度 $nlog^2$