A

在最优解中，一个静态库最多出现两次：把一次放在开头，一次放在结尾，那么所有与这个静态库有关的依赖关系都被满足了。

从左往右考虑最后的命令，令 $f[S]$ 表示已经添加了集合 S 的最优长度，枚举下一次加入的静态库 i，如果所有依赖 i 的库都已经被添加，那么这一次转移的代价为 1：在当前序列最后加上一个 i 即可；如果存在依赖 i 的库没有被添加，那么转移的代价为 2: 还要在序列的最后加上一个 i 来满足这些关系。

时间复杂度 $O(2^n n)$

B

问题等价于求 $\frac{a}{b} \le \frac{c}{d}, \gcd(a,b)=1, b \le N$ 的最简分数 $\frac{a}{b}$ 有多少个。如果不考虑互质，那么只需要满足 $ad \le bc, b \le N$，这相当于数一条线段下方的整点个数，是一个欧几里得算法的经典问题，令这个问题的答案是 $f(N)$.

考虑如何去掉不互质的情况，可以枚举 $d=\gcd(a,b)$ 然后用简单的容斥得到答案为 $\sum_{i=1}^{N} \mu(i) f(\lfloor \frac{N}{i} \rfloor)$。因为 $\lfloor \frac{N}{i} \rfloor$ 只有 $O(\sqrt{n})$ 种取值，所以可以暴力计算出这一部分所有需要的值。枚举 $\lfloor \frac{N}{i} \rfloor$，问题转化成了求 $\mu(i)$ 的前缀和，这是一个杜教筛的经典问题，可以在 $O(N^{\frac{2}{3}})$ 内求解。

时间复杂度 $O(\sqrt{N}\log N + N^{\frac{2}{3}})$.

C

可以把 $[0,m]$ 拆成 $O(\log m)$ 个区间，每个区间形如 $[k\times 2^i, (k+1)\times 2^i)$. 一段区间的所有数异或上 n，结果必定还是一个相同形式的区间。因此最后的答案可以拆成 $O(\log m)$ 段区间内所有数的乘积。

问题转化成了求阶乘，因为模数已经给定，所以可以直接分段打表。即选取 L，在程序里作为常量打下所有 $(iL)!$ 的值，这样在求阶乘的时候只要算 $O(L)$ 次乘法就可以了。标程选取了 $L=2\times 10^5$.

时间复杂度 $O(L\log m)$.

D

极限的意义为当答案趋于无穷大时，让答案再增加一最少需要多少次操作。

令 $f(u)$ 表示在以 u 为根的子树中进行游戏，先手和原游戏运行到这个点相同时的极限。考虑如何转移：

1. 如果 u 是叶子，那么 $f(u)=1$
2. 如果 u 是第一个人先手，那么第一个人可以选择一个 f 最小的孩子。因此此时只需要对这个孩子进行操作就可以了，其他孩子的子树都可以不管。$f(u)=\min f(v)$
3. 如果 u 是第二个人先手，那么操作时必须让所有孩子的答案同时增加，不然只要有一个孩子的答案较小，第二个人都会选择这个孩子移动过去。$f(u)=\sum f(v)$.

直接进行树形 DP，时间复杂度 $O(n)$

E

令 $f(n)$ 为 $m=n$ 的三元组个数。根据中国剩余定理，$f(n)$ 是一个积性函数，因此只要对所有 p^k 求出答案，就可以用线性筛筛出所有值。

当 p 是奇数时，在模 p^k 下考虑数对 $(a+2b, a-2b)$，即 $((x+y)^2, (x-y)^2)$ ，这样的数对是和 (a,b) 一一对应的。且给出 $x+y, x-y$ 必有 x, y 满足条件。所以只需要求二次剩余的个数就行了，答案就是打次剩余个数的平方。不难发现二次剩余的个数为 $\sum_{2i\le k} \frac{\phi(p^{k-2i})}{2}$.

当 p 是 2 时，没有奇数这么好的性质，但是打表之后可以发现答案是关于 k 的线性递推。找到规律后即可求解。

时间复杂度 $O(n)$.

F

问题是询问 n 条直线在一个矩形的框内的相交对数。如果一条直线和这个框不相交，那么可以直接忽略。否则这条直线把框的边界分成了两个区域。

两条直线在框内相交当且仅当一条直线的交点分落在第二条直线的两个区域内，问题被转化为了给出 n 个区间，问这些区间中相交且互不包含的对数有多少，这是一个二维数点，可以排序后树状数组求解。

时间复杂度 $O(n\log n)$.

G

引理：对于一个联通无向图，它有至少一个桥 (i,j,k)，如果我们接一个新点 t 到这个图上，至少会产生一个新桥。

反证，假设加上新点 t 后没有产生新的桥。若 (t,u) 有边，因为没有产生新的桥，所以 t 和所有 u 的相邻点都有边。因此 t 必须和这个联通无向图的所有点都有边。此时 (i,j,t) 是一个桥，矛盾。

任何一个存在桥的联通无向图，都可以看成从这个桥出发，把剩下的点依次接入。因此在这个过程中，至少产生了 $n-2$ 个桥。

所以在这道题中，因为桥的个数小于等于 8，所以大部分的联通块都是团，只有一些大小不超过 10 的块有可能含有桥。可以暴力搜索出所有 n 个点 m 个桥（$n\le 10$）的图的个数，通过一些位运算优化和剪枝可以在 10 分钟之内搜出来。之后就是一个非常简单动态规划了。

时间复杂度 $O(n^2)$.

H

-1 当且仅当第一扇门为 P，接下来忽略这种情况。

设 A_i 为连续走过前 i 道门后获得的能量。如果一个人无法直接到达终点，那么他一定停留在他能走到的 A_i 最大的位置。

考虑一个一个地加入新的人。不难发现每一次加入新的人后，终止移动时所有人一定停留在一个相同的位置。因为如果新加的人停留在了更远的位置，那么他带来的能量一定大于其他人，这时其他人一定可以至少移动到他的位置。

因此可以不停的加入新的人，维护在最终停下时，所有人在的位置以及获得的能量。如果新加的人可以字节走到终点，那么当前的人数就是答案。

因为停留的位置一定是单调向右的，所以可以用指针直接扫过去。

时间复杂度 $O(n)$.

I

无论有没有放置镜子，光线射入后的运动轨迹都是唯一的。因此无论镜子如何放置，所有可能的光路要么连接着两个射入点，要么在格子内形成了环。因为光路的总长度不变，所以答案就是环的总长度。

轮廓线 DP，状态为轮廓线上的光路的联通状态，每一条的光路的长度以及放置的镜子数。直接暴力转移会 TLE，可以进行如下优化：

1. 如果一条光路在轮廓线上只出现了一次，那么它就不可能成环，可以直接删去。
2. 如果存在一个状态严格劣于另一个状态（连通性相同，每一条光路的长度都不更长且放置的镜子不更少），那么这个状态就可以删去。

优化后就可以通过这个题了。

J

判断后缀 i 能否成为最大后缀，每一个其他后缀相当于给出了一对限制 (a, b) 表示字符 a 要比字符 b 大（限制来自于两个后缀第一个不同的字符）。判断这些限制能否同时满足只要用进行一次拓扑排序即可。

因为拓扑排序不是瓶颈，考虑如何快速得到所有限制。对串建后缀树，那么对于一个后缀来说，所有的限制都出现在它到根路径上的分叉节点中。只需要对每一个分叉存下它产生了哪些限制，然后 DFS 一遍树把限制统计起来即可。

时间复杂度 $O(n|S|^2)$，其中 S 为字符集。