区间加全局求 MEX（add）

记 m 为原 MEX。$m=n$ 的时候显然是不能进一步增大的，否则在必要的 $0\sim m-1$ 各一个以外至少有一个无用元素，将其改为 m 就可以使得 MEX 增加至少 1，但为了恰好增加 1 需要将所有的 $m+1$ 也修改成 m，据此完成判断即可。$O(n)$ 或 $O(n\log n)$。

蚂蚁（ants）

首先，如果有一边的障碍物损坏，问题是简单的；如果至少有一边的障碍物在 n 次撞击以内损坏，问题也是简单的：只需要模拟 $O(n)$ 次碰撞即转换为前述情形。第一个做法是二分找到第一个障碍物损坏的时刻；第二个做法是在 $2L$ 时间后两边的障碍物各被撞 n 次且所有蚂蚁回到原位，其中 $L=10^9+1$，据此也可以进行 $O(n)$ 次模拟完成本题，模拟过程即为用两个队列存储下一次左/右障碍物被撞的时刻，然后再根据此障碍物是否损坏，判断是否将这个时刻 $+L$ 放到另一个队列中即可。

$O(n)$ 或 $O(n\log L)$。

修建隧道（build）

考虑什么时候不得不加边：如果在我们 DFS 的过程中走到了 i，并且不可能在下一步走到 $i+1$，那么我们必须添加一条边使得下一步能走到 $i+1$。这等价于在当前的 DFS 栈中，最深的一个有未访问邻居的点与 $i+1$ 不相邻。 直接模拟这个过程即可。显然不按照这个过程加边不能使得答案变得更优。
$O(n+m)$。

大赢（bigwin）

题意即为选出总长度最大的若干不交子区间，使得每个子区间的和都非负。做前缀和之后设 dp_i 表示考虑 $1\sim i$ 之后得到的答案，那么 $dp_i=\max_{j<i,s_j\le s_i}(dp_j+i-j), dp_i=\max(dp_i,dp_{i-1})$，得到 $O(n^2)$ 的 dp，然后可以做自然的线段树/树状数组优化。此外，考虑求 $\max_{j<i,s_j\le s_i}(dp_j-s_j)$ 可以维护 (s_j,dp_j-j) 的二元组放进一个 `std::set`，那么第一维越小越可能转移，第二维越大转移越优，据此维护一个第一维单增，第二维单减的 set，转移时进去 `lower_bound` 查询即可，实现复杂度更低。

循环序列（cyc）

一个序列开头为 x 等价于 $t\equiv b\pmod k$ 形式的方程，一段开头都是 x 可以用裴蜀定理判断是否有解（只考虑两个方程 $t\equiv b_1\pmod{k_1}, t\equiv b_2\pmod{k_2}$，等价于 $k_1x_1-k_2x_2=b_2-b_1$ 是否有解），由单调性，双指针对每个 l 求出最远的 r 使得 $[l,r]$ 合法，但需要处理这里的判断不能删除的问题，一个暴力的每次扩展之后遍历 $[l,r]$ 判断，但复杂度不对，考虑到 k 很小，实际上对于每个 k 只能有至多一个不同的 b，据此，每次扩展都只需要进行 $O(k)$ 次合并方程，则总复杂度 $O(k\times\sum k_i)$，根据去重实现可能多一个 $\log$ 因子。