good

即求 S 至少有多少个元素，用 dfs 或者 bfs 搜索出所有 x 可到达的点即可。对于第二类边，可以说明其至多只有 $O(\sqrt{n})$ 条。注意建边如果全部用 vector 可能会在搜索之前就 MLE，需要使用链式前向星建图。对于第一类边，也可以不显式建出，直接枚举 $f(y) \leq 63$，常数较小。甚至可以将二者进行平衡，只对某些 $f(y)$ 建边。

number

数位+状压 dp，记录前面的 $k-1$ 位是什么，每次加入一位就进行检查是否形成回文串，再记一下有没有顶上界。时间复杂度 $O(n2^k)$。

考虑匹配回文串的过程，类似 KMP，每次在后面加一个字符，如果能匹配就匹配，否则失配了，就找到下一个可能可以匹配的位置开始。这样只用记前 $\lceil \frac{k}{2} \rceil$ 位以及现在往后还匹配了多少位，后面的位置可以用前面的位置对称表示出来。状态数减少到 $O(2^{k/2}k)$，需要预处理失配跳到的位置，时间复杂度 $O(2^{k/2}k^2 + n2^{k/2}k)$，可以通过。

看到考场同学的一种更好理解的做法，就是把所有可能的回文串插入到 AC 自动机中，然后做匹配，维护现在在 AC 自动机上的哪个位置。由于可能的回文串只有 $O(2^{k/2})$ 个，因此状态数只有 $O(2^{k/2}k)$，降低了预处理

时间复杂度。

xor

对于树，只有一条路径 $val_{i,j}=val_{1,i}\oplus val_{1,j}$。

对于仙人掌，直接爆搜状态数大概是 $O(2^{n/2})$，可以通过前两个子任务。

考虑建出圆方树，相当于 a,b 在圆方树上经过的每个原图为环的方点都有两种选择，一种是顺时针，一种是逆时针。不妨仍定义 val_i 表示 1 走到 i，环上都按顺时针走的路径权值，那么所有可能的答案即为 $B=\{x|x\oplus val_i\oplus val_j\in A\}$，其中 A 表示 a,b 圆方树路径上每个环的权值形成的线性基，因为顺逆时针异或起来就是整个环。那么只用求一条链上的线性基即可，再将两个线性基合并起来。暴力从每个点出发求一遍即可做到 $O(n^2\log V)$，由此联想到点分治即可做到 $O(n\log n\log V)$。一种更优美的方法是类似区间线性基做法，只维护到根链的线性基，并对每个值维护时间戳等于深度，在插入时遇到时间戳更早的基就交换，查询时只保留时间戳大于 LCA 深度的基，时间复杂度 $O(n\log V)$。

对于查询，直接将两条链的基合并即可做到 $O(q\log^2 V)$，线性基的第 k 小值可以在 $O(\log V)$ 内求出。对于 $s_i\neq 0$，只需要再维护线性基内每个数 $b_i\&s_i$ 的基，每次确定是否要异或一个值时检查异或后能否凑出含 s 的

数，如果能，那么能凑出的个数即 $2^{\text{原基大小}}/2^{\text{保留}s\text{的基大小}}$。总时间复杂度 $O(n\log V+q\log^2 V)$。

war

考虑扫描线，扫描 x 时维护 $f_{i,j}$ 表示矩形右上角为 (i,j) 是否可行。

显然只有每个点的坐标有用，因此可以离散化。

分为三种转移：

向右上拓展：$f_{i,j}\wedge 2(k+l)-(i+j)\le K\to f_{k,l}$

向右：$f_{i,j}\wedge 2(k-i)+j\le K\to f_{k,j}$

向上：$f_{i,j}\wedge 2(k-j)+i\le K\to f_{i,k}$

注意只关心可行性，将状态改为 f_i 表示扫到 x 时最大可行的 (f_i,i)。每次移动 $x\to x'$，考虑有哪些 f_i 变成 x'，并激活一些可用的 f。由于在 x 处已经完成了向上的转移，那么只能是先向右或先向右上再向上。

用线段树维护 f_i。先做向右的转移。设当前 x' 对应的最小的 y 为 y_0，我们需要将所有 $i\ge y_0$ 的 f_i 推到 x'。注意如果一个 $f_i>0$ 且这次没能向右转移，那么它只能等一个 f_j 向上转移到它，这个 f_i 也就暂时没用了，可以设为 -1。因此考虑在线段树上暴力递归到**除了 -1 外每个 f 相同**且都能转移或都不能的区间，进行一个区间赋值**（赋值非 -1 的激活点）**。注意我们可以顺便把向上的操作也做了，即将一段 f_i 设为 x' 后做一个区间延

伸，这个延伸只用考虑能走多远，并不需要考虑 -1，因此也是个区间赋值**（赋值所有激活点）**。所以前面要求 f 相同的原因是非 -1 的 f 形成单调不降的连续段，可能有两段之间的 -1 还没有更新到。

接下来按 y 从小到大考虑新增的点，每次可能需要激活一个新的 i，然后看一下 f_i 能否为 x'，分向右上转移过来和向上转移过来即可。如果可以，那么同样进行一个区间延伸。否则先标记 $f_i=-1$，等到后面可能激活某个点后能转移过来再转移。

由于暴力递归部分有颜色段均摊，时间复杂度 $O(n\log n)$。