嘟嘟

将每个方格看作点，对于每对相邻方格，由方格内数较大的一方向较小的一方连边，由方格表的性质易知建成的图一定是一个 无出度的点仅有一个的 dag，那么对于任意一个点，沿着出边走，一定能走到 1 所在点。

可以得到一个随机算法：随机询问方格，如果遇到一个较小的数，每次在其四周找到比它小的数走到那里，一直沿着出边走，直至找到 1 所在点，设阈值为 n 可使得期望复杂度为 $O(n)$。

但是随机算法无法满足要求，考虑在方格表上二分。

第一轮，先询问整个方格表最中间的两行，未询问方格组成上下两部分，考察这两行的最小值 x，如果这个最小值在两行中的上面一行，那么 1 的位置一定在上面部分，反之则在下面部分。

第二轮，不妨设 1 在上面部分，询问上面部分最中间的两个半列，目前可能出现 1 的未询问方格组成左右两部分。还是考察这两个半列的最小值 y，不妨设 y 在两个半列中左边的半列。如果 $y < x$，那么 1 的位置在左面部分，反之，1 的位置在与 x 相邻的那部分。

以此类推，每次可以把 1 的位置限定在全局询问过的方格的最小值的相邻部分，总询问次数为 $2 \times (2n + n + \frac{n}{2} \cdots) - 2n = 6n$，期望得分 55。

进一步发现，我们并不需要知道两行的最小值，只需要先询问一行的最小值，再在这个最小值上下找到比该最小值更小的一个值，1 的位置也随之确定了，于是总询问次数降低到 $n + \frac{n}{2} + \frac{n}{2} + \frac{n}{4} \cdots = 3n$，期望得分 100。

由于交互库没有使用自适应，有一些随机算法通过了，在此表示对造交互库的同学的强烈谴责。

滴滴

为了下文表达方便，认为排列从 0 到 $n-1$，构造完后整体加一即可。

观察到 n 相对于 k 巨大，也许我们并不需要找到一些归纳的构造方式。

容易发现一个非常简单的 $n = 2k+1$ 的构造，即任意选择 p_1，并取 $p_i + k \equiv p_{i+1} \pmod n$，容易验证其正确性，并且其具有重要性质 $|p_n - p_1| \in \{k, k+1\}$。将序列反转可以知道，可以任意选择满足 $|p_n - p_1| \in \{k, k+1\}$ 并且在范围内的 p_1, p_n，都有满足条件的构造。

如果在上述构造中取 $p_1 = 0$，并且在序列最后加上 $2k+1$，我们得到了一个 $n = 2k+2$ 的构造，但这个构造没有更好的性质。

设 $2k+1$ 的构造在 $p_1 = x, p_n = y$ 时为 $A(x, y)$，$2k+2$ 的构造为 $B(x, y)$。

发现当 n 巨大的时候，可以使用不超过 $2k+1$ 个 B 以及很多个 A 来拼接出 n 的构造，现在只需要满足其两段拼接处的条件，容易发现两个 A 之间的拼接由于 A 自由度高，比较容易，需要找出一种方案能将两个 B 连接在一起。

考虑如下构造，设当前的 B 为 $B(a, a+2k+1)$，可以构造
$B(a, a+2k+1) - A(a+3k+1, a+4k+1) - A(a+5k+1, a+6k+1) - \cdots - A(a+2k^2-k+1, a+2k^2+1) - B(a+2k^2+k+1, a+$
最后一个 A 对应的构造中最大数为 $a + 2k^2 + k$，可以拼接上。这里的所有构造值域均是相接的，可以计算验证。

于是我们用 $k-1$ 个 A 将相邻两个 B 连接了起来，多余的 A 可以简单排列在最后一个 B 之后。

通过计算知道这样的构造方法能构造成功的 n 下界为 $4k^3 + O(k^2)$，在本题的数据范围下成立。

叭叭呜

考虑 (u, fa_u) 关于 $[l, r]$ 好的条件，转化为 u 子树内有 $[l, r]$ 之间的点且 u 并不是 $\text{lca}(l, l+1 .. r)$ 本身或其祖先，其中后面条件可以容斥掉，只需要减去询问区间的所有子区间 lca 深度和即可。

现在计算单个区间权值转化为两个问题：区间 lca 深度，有多少个子树内有该区间之间的点。

- 区间 lca 深度。

即解决询问区间子区间 lca 深度和，考虑扫描右端点，维护所有左端点对应 lca 深度，并维护历史和。

对于一个当前的右端点 r，其所有左端点的对应 lca 一定在一条链上，那么加入 $r+1$ 时，将 $r+1$ 与这条链的最低点也就是 r 取 lca，将那些比这个 lca 还深的对应 lca 都改成这个 lca，更浅的不变，可以直接实现区间覆盖区间历史和，也可以每次暴力修改一段左端点区间的对应 lca，实现区间加区间历史和，修改次数均摊线性。这一部分是 $O(n \log n)$ 的。

- 有多少个子树内有该区间之间的点

考察一个点 u 不计入 $[l, r]$ 答案的时候，将 u 子树内点按编号排序，那么这个 $[l, r]$ 一定是介于两个编号排序后相邻的点之间的，可以抽象成矩形加。

考虑树上启发式合并，对每个子树维护一个 set 或一棵平衡树之类，内部存子树内点的编号，每次把小子树的某点合并到大子树里的时候，将大子树对应平衡树中该点编号前驱后继取出，并将这个矩形加终止，换用新的两个矩形加。

问题变为 $O(n \log n)$ 次矩形加，询问即为矩形查询，可以在 $O(n \log^2 n)$ 的时间内得到解决。

综上，整个问题在 $O(n \log^2 n)$ 的时间内得到解决。