tree

考虑以下做法：我们以任意顺序加入同一个颜色的点，同时维护这些点中距离最大的点对 (x, y)。加入一个点 a 时，检查 $dis(a, x)$ 或 $dis(a, y)$ 是否大于 $dis(x, y)$。如果是则更新 (x, y)。考虑证明正确性。我们使用归纳法，当点数 ≤ 3 时显然正确。考虑新加进去一个点 a，之前加进去了点 b，我们需要证明：当 $dis(a, x), dis(a, y), dis(b, x), dis(b, y)$ 均 $\le dis(x, y)$ 时，$dis(a, b) \le dis(x, y)$。考虑所有 $dis(a, x), dis(a, y) \le dis(x, y)$ 的点的集合。显然这个集合包含了路径 (x, y) 上的所有点，然后这些点又往外延伸了一段距离。对于这个集合中的点 r 我们可以找到与他对应的路径 (x, y) 上的点，设为 p_r。显然 $dis(p_r, r) \le min(dis(p_r, x), dis(p_r, y)) \le \frac{1}{2}dis(x, y)$。因此，若 $p_a = p_b$，则 $dis(a, b) \le 2(\frac{1}{2}dis(x, y)) = dis(x, y)$。若 $p_a \ne p_b$，
$dis(a, b) = dis(p_a, a) + dis(p_a, p_b) + dis(p_b, b) \le dis(x, y)$。原命题得证。

cartesian

先假设我们要交换 a_i, a_{i+1}。不失一般性，我们设 $a_i < a_{i+1}$。

考虑这两个元素在笛卡尔树上的位置。显然 a_{i+1} 为某个子树的根，a_i 要么是 a_{i+1} 的左儿子，要么是 a_{i+1} 左子树中某个点的右儿子。并且，a_i 没有右儿子。交换后相当于将 a_i 从 a_{i+1} 的左子树中删去，并加到它的右子树里。对于 a_i 的左子树，他们的 dep 都减少了 1。对于 a_{i+1} 的右子树，a_i 会作为某个节点的左儿子，并且这个节点原来的左子树会变成 a_i 的右儿子。他们的 dep 都增加了 1。再计算 a_i 的深度变化就可以得到答案。

找 dep 变化为 1 的区间可以去找 a_i 往前第一个比他大的节点和 a_{i+1} 往后第一个比 a_i 大的节点。我们使用两棵线段树分别维护权值与深度即可。

$a_i > a_{i+1}$ 的情况同理。

war

首先我们用每个点代表其父亲到这个点的边。

Subtask #1

暴力维护每个点每个时刻的权值，修改可以简单差分维护，最后做一次前缀和即可。

回答询问时暴力遍历每个点，检查区间内是否有 0 即可，同样可以前缀和预处理出 0 的个数，回答时差分检查即可。

时间复杂度 $O(n(m + q))$ ，期望得分 $20pts$ 。

Subtask #3

首先处理出每次派遣操作的持续时间段 $[l, r]$ ，操作即为 x 到 y 路径上的所有边的权值序列区间 $[l, r]$ 加一。差分处理操作，最后再用启发式合并和线段树求出每个点每个时刻的权值。

每次询问对每个点暴力检查的时间复杂度是 $O(qn \log m)$ 的，无法承受。

发现每次询问都只需要求出深度最大的满足条件的点，而启发式合并也是从深度大的合并向深度小的，因此可以在启发式合并的过程中处理询问。

考虑如果 $x = 1$ ，则只需要继承儿子的所有修改即可，不需要撤销。因为儿子的深度比当前节点大，因此儿子的每个全部不为 0 的区间当前节点都不需要考虑，只需要处理新产生的区间即可。

考虑每加入一个修改区间 $[l,r]$ ，新产生的最长全部不为 0 的区间 $[L,R]$ 显然可以在线段树上二分得到，其能够回答的询问为一个矩形。直接使用树套树维护空间和时间复杂度都会爆炸。发现每次如果找到一个满足条件的询问，记录答案后直接删除该询问即可，因此将所有询问 $[l,r]$ 在 r 上挂上 l ，再用一棵线段树维护区间的最大左端点，每次区间询问最大左端点是否大于等于 L 即可。

如果 $x\neq 1$ ，则需要撤销修改，只需要在 lca 的对应子树处将修改撤销即可。

时间复杂度 $O(m\log^2 m+q\log m)$ ，期望得分 $100pts$ 。

pow

我们考虑如果能质因数分解怎么做。对于每个质数，我们给予其一个哈希系数 $s(p)$。

然后对于一个数，令 $x=\prod_{p\in Prime}p^{\alpha}$，则 $h(x)=\sum_{p\in Prime}(\alpha \bmod k)\times s(p)$。

从 $h(\prod_{i=l}^{r}a_i)$ 的值可以从 $h(\prod_{i=l}^{r-1}a_i)$ 中修改在 $O(\sum\Delta\alpha)$ 内获取，因此所有哈希值可以在 $O(n^2\log a)$ 内解决。

不难发现这玩意本质上就是字符串哈希，找一个种子 $seed$，然后哈希系数依次为 $seed,seed^2,seed^3\dots$ 即可，因此正确性显然。

扫描线枚举 l_2，map 求值，可以轻松做到 $O(n^2\log n)$，这些都是简单的。

但是质因数分解不好做，这时候我们考虑一种新的模型。

UPD: 有人发过了，我们还是晚了一步 :(

Link: https://codeforces.com/blog/entry/108053

即找到一个数组 p，使得对于任意 $i<j$，$\gcd(p_i,p_j)=1$，且存在一个 $n\times m$ 的非负整数矩阵 b。满足 $a_i=\prod p_j^{b_{i,j}}$。

即 p 两两互质，且可以分解 a。

然后你考虑一开始 $p=[a_1]$，然后依次枚举 $a_2,a_3,\dots a_n$ 并尝试加入其中。

由于直接加入会破坏两两互质的性质，考虑如下加入方法，假设当前加入的是 x：

对于所有 p 内元素，假设当前在检索 y，我们求 $g=\gcd(x,y)$。

- 如果 $g=1$，显然无事发生，可以继续检索，看看会不会破坏互质性质，否则进行如下操作：

我们考虑 $g>1$ 的情况，假设 $x=gb,y=ga$，则我们先拆成 g,a,b，但由于 $\gcd(g,a),\gcd(g,b)$ 可能不是 1。因此还需要额外进行更复杂的操作。

我们引入一个临时的数组 q。

我们先不考虑 g,a 的关系，将 a 加入 q，我们先只考虑 g 和 b 的关系，如果 $\gcd(g,b)>1$，则重复上述操作，将 g 再次拆成 $\gcd(g,b),\frac{g}{gcd(g,b)}$，并将 b 变成 $\frac{b}{gcd(g,b)}$。一直重复该过程，直到 $\gcd$ 为 1 为止。对应原来的 x,y，会将 x 除掉一定的因子，y 则被直接拆分成了一个序列，即刚刚说的 q。

用 x 可以更新剩余的 p 中的元素，而由 y 拆分出的序列，我们假设其为 q，你会发现，这时候我们发现，如果把 q 视为 a，实际上我们就是做了一遍原问题。

q 的长度是多少呢？首先不难发现 q 的乘积就是 y，1 又可以略去，因此长度就是 $O(\log a)$。

你会说这样递归下去啥时候是个头，但是没关系，我们有神奇的复杂度证明！

令 $f(\alpha,x)$，表示 x 一直除 α 直到不能整除，能除掉多少个 α。

令 $g(\alpha)=\sum_{x\in p}f(\alpha,x)$，我们有如下结论：

当出现一次 $\gcd > 1$ 时，至少存在一个 α，使得 $g(\alpha)$ 减小！

为什么？因为我们不难发现，一次 $\gcd > 1$ 即对应着一次 ga, gb 被拆分成 g, a, b，乘奇总和减少了 g。也即对于 α，$g(\alpha)$ 减小 $f(\alpha, \gcd)$。

加入一个 x 使得 $\sum_{\alpha\in Prime} g(\alpha)$ 至多增加 $O(\log a)$，因此生成的 q 总长度不超 $n\log a$。而用 $\log a$ 的长度遍历过去，总使用 $\gcd$ 次数 $O(n\log^2 a)$，此部分复杂度 $O(n\log^3 a)$。

而整个序列检索过去，p 长度显然不超 $n\log a$，而用 n 的长度遍历过去，总使用 $\gcd$ 次数 $O(n^2\log a)$，复杂度 $O(n^2\log^2 a)$。

你会说这两个都好大啊！根本过不去，没关系。我们可以将 10^6 以下质数先除掉，这样每个数的质因数个数至多 5 个，也对应了 $\sum_{\alpha\in Prime} f(\alpha, a_i) < 6$，实际上复杂度做到了 $O((n5^2 + n^2 5)\log a + n\frac{10^6}{\ln 10^6})$，足以通过！

拆分后大家都会做了。

但你要注意到每个 p_i 可能已经是 k 次幂，4 次幂之类。没关系，我们有如下方法！

初始令 $t = k$，对于所有 k 的质因数 w，重复尝试将其开 w 次根，如果开根出来是整数则直接开 w 次根，且 $t := \frac{t}{w}$。

然后实际上 $p_i^t, p_i^{2t}, p_i^{3t}\dots$ 就是 k 次幂，哈希的每个数的指数不是模 k，而是对于每个质数单独模其对应 t 值即可。

而且注意到 w 只需要枚举到 $O(\log a)$，又有素数分布密度在 $O(\ln n)$ 左右，因此自己写一个二分去开根也是 $O(n\log^2 a)$。

来自组题人：还有一种不好卡的方法，枚举两个数把他们的 $\gcd$ 加进去。