resource

第一档部分分可以直接搜索。考虑按照元素大小从小到大考虑，最小值只能保留一个，其它的至少要 $+1$，那么一定选择 b 最大的保留，其它的加一之后继续处理。

使用优先队列维护当前等于 i 的元素的 b 值，i 从 1 到 10^5 枚举，每次先把原数组中值等于 i 的 b 放入优先队列，弹出 b 最大的那个，剩余元素的 b 之和累加到答案并继续这个过程。

注意到每次有效枚举至少会减少一个元素，那么有效枚举实际上只有 $O(n)$ 次，那么对于一大段值域上的部分是不用扫描的，可以加速到 $O(n\log n)$。如果不使用优先队列或实现有误也可以通过 $b_i=1$ 的部分分。

mst

对于单次询问，可以从高到低按位考虑，即枚举 $i=29\to 0$ 与初始等于原边集的边集 E，检查能否使用边权第 i 位是 0 的边组成一棵生成树，如果可以，则只保留 E 中边权第 i 位是 0 的边，答案的第 i 位设为 0，否则 E 不变，答案此位设为 1。这样做一次是 $O(n\alpha(n)\log V)$ 的，每次重头处理这个过程可以通过 60 分。

接下来考虑每组独立的加零边操作对答案的影响。依然考虑 $i=29\to 0$，如果在上述对原图执行的过程中答案的第 i 位不是 1，那么这次操作对 i 无影响，否则考虑能否利用新边使得答案这一位的答案为 0。首先，如果在上述过程中这一步仅适用第 i 位为 0 的边有 >2 个连通块，那么新边不能改变这一位的答案，否则连通块 $=2$，如果新边恰好架在它们之间，这一位的答案就可以变为 0。

下一个问题是，这条边需要添加到维护的边集中并且继续此后的答案计算过程（一条边可能不止改变答案的其中一位，且不能独立的检验答案的每一位是否可以改变），这导致复杂度依然没有改变。

考虑如果在某一位 b 我们使用上述方法判断出了答案可以减小，那么实际上后续过程只与 b 有关而非 u_i,v_i，我们只需要预处理对 不使用零边得到的两个连通块的并集 分别跑之后的 mst 过程，即 不要求这两个连通块被原边集的边连起来 而算出的答案，就可以快速回答每个询问了。

$O((n+m)\log^2 V+q\log V)$。

book

前 25 分可以在 Trie 上模拟一下题意。接下来考虑没有 2 操作的部分，即每次相当于给一个 Trie 上子树做子树加，可以利用 Trie 的 dfn 序配合线段树完成。

字符集大小为 1 时 Trie 的结构是一条链，每次相当于做后缀加，但是由于有加入新串的操作，所以容易想到给这些新串预留位置，在其加入时查询一下在其加入之前对这个位置加的总和，并在之后对其的询问中减去这一部分。

这个想法同样适用于字符集大小不为 1 的情况，只需要离线将这些串都插入 Trie 中，正常做子树加，并在新串加入的时刻记录其对应结点的值，此后回答时用数据结构查询出来的值减去这个值即可。区间加单点查，使用树状数组。$O(L(|\Sigma|+\log L))$，其中 $|\Sigma|=26$。

一个实现细节是需要查找一个串 i 的前缀在 Trie 树上的位置并支持在线插入，可以适用倍增解决。

card

预处理去掉每个数的平方因子，则此时每个数的 $\omega\le 7$。记 t_i 表示 i 的质因子个数，则两个数 x,y 要操作 $t_x+t_y-2t_{\gcd(x,y)}$ 次才能满足 $x\times y$ 是完全平方数。

当 a_i 是质数，答案是 0 当且仅当区间里有两个相同的数，否则答案是 2。

考虑若 $a|b$，则 $t_a \le t_b$，要最小化 $t_x + t_y - 2t_{\mathrm{gcd}}$，则如果只处理一次询问，可以枚举 $d|x, d|y$ 统计 $t_x + t_y - 2t_d$ 的最小值。

枚举 d 拉出序列中 d 的倍数，注意到处理后的 a_i 不含平方因子，则其因子个数为 $2^\omega \le 128$ 个，则总共只有 $128n$ 次被拉出来统计。对于确定的 d，希望最小化 $t_x + t_y$，也就是需要找到最小和次小的 t。

对于一次询问，暴力地统计其中的最小值和次小值。对于多次询问，考虑真正有贡献的（最小值，次小值）点对实际只有 $O(len)$ 对。具体地，枚举次小值所在的位置，找到其左，右第一个比自己小的作为最小值，则每个区间的情况都被包含了。正确性考虑截取其中的一个区间 $[l, r]$，找到区间最小值，则区间里的次小值会找到它作为某一边第一个比自己小的数计入答案。则问题转为区间包含的线段的最小权值，容易离线扫描线。

线段有 $2^8 n \le 5 \times 10^7$ 个，而询问只有 10^6 个，扫描线时需要支持的操作是单点修改，后缀求 $\min$。注意到答案首先不超过 14，那么可以改为前缀取 $\min$ 和单点查值，前缀取 $\min$ 则值单调不降，暴力地往前更新 f 值直到不能更新，则更新次数容易均摊到 $2\omega n \le 14n$，这样统计答案的部分就快了不少。

单调栈部分复杂度是 $O(2^\omega n)$，统计答案部分复杂度是 $O(2^\omega n + q + \omega n)$，则总复杂度是 $O(\omega 2^\omega n + q + V \times \omega)$。