宿雾若水遥（mizue）

> std 长度：2993 Bytes

考虑一个 $w(l,r)$ 怎么查询，扫描线扫 r，对每个点 u 维护 p_u 表示最小的 l 使得编号在 $[l,r]$ 中的链可以覆盖 u 节点，那么 $w(l,r)$ 的值就是 $p_u \geq l$ 的点的个数；每次相当于把一条链上的 p_u 覆盖为 r。

考虑用线段树维护序列 c_i 表示 $p_u \geq i$ 的元素个数，然后树剖+颜色段均摊，每次相当于对 c 做一个区间加，以及单点查询。那么查询 $[L,R]$ 的所有子区间元素和就相当于一个历史和问题。

于是线段树维护区间加，区间历史和即可。总复杂度 $O(m\log m\log n+q\log m)$。如果把树剖+颜色段均摊改为 LCT 就可以做到严格的 1log，本题对此不做要求。

缠忆君影梦相见（asano）

> std 长度：1740 Bytes

$O(m^4)$ 做法：直接枚举三条边删掉，然后 $O(n+m)$ 判断是否强连通。

注意到判断强连通只需要检查对每个 i，是否有 1 和 i 互相可达。这可以通过对原图和反图分别做一遍 BFS 来 check；可以压位做到 $O(n^2/\omega)$ 时间内 BFS，于是我们就优化到了 $O(n^2m^3/\omega)$。

上述过程给我们一些启发：考虑直接求出原图和反图的 BFS 树，这两棵树的树边一共有不超过 $2(n-1)$ 条，如果这些边都没有被删掉，那么图一定仍然是强连通的。于是，我们就把枚举量降低到了 $O(n)$。

具体地，设 $\operatorname{solve}(E,k)$ 表示在边集 E 中删掉 k 条边使得其不强连通的方案数，那么：

- 若当前 $G=(V,E)$ 已经不强连通，则返回 $\binom{|E|}{k}$
- 否则，求出原图和反图的 BFS 树，得到至多 $2(n-1)$ 条边，枚举这些边中的一条边 e 删掉，递归到 $\operatorname{solve}(E-e,k-1)$。

该算法的时间复杂度为 $O(\frac{n^5}{\omega})$，其中多的 n^2/ω 是判断强连通 + 找 BFS 树的复杂度。

晓月又经宵（tsuki）

> std 长度：6721 Bytes

下文不区分 n,m。

- Subtask 3 的做法

最后相当于选取 a,b 中各一个区间，最大化它们的 $\min$ 乘上 len 之和。不难发现，我们选取的区间一定满足往左右扩展会导致 min 减小，或者干脆就无法扩展也就是顶到边界。那么 a 中选取的区间肯定是笛卡尔树上的区间；b 中选取的区间则要么是完整的笛卡尔树区间，要么是某个原序列笛卡尔树区间和询问区间 $[l,r]$ 的交。

在 subtask 3 中，询问区间总是笛卡尔树区间，于是只需要考虑选取一个完整笛卡尔树区间的情况。我们可以先 $O(n\log n)$ 提前对 $O(n)$ 个笛卡尔树区间预处理答案，然后做一个二维数点。总复杂度 $O((n+q)\log n)$。

- $O((n+q)\sqrt{n})$ 做法

根据上面的分析，我们发现可以在 $O((n+q)\log n)$ 时间内解决选取完整笛卡尔树区间的 case，现在就剩下选取不完整区间的情况。这意味着一定是选询问区间的一段前缀或者一段后缀。我们不妨就假设选取的是一段后缀。

设 A_i 表示序列 a 中满足所有数都 $\geq i$ 的区间的最长长度，我们考虑扫描线扫 r，维护序列 B，其中 B_i 表示以当前 r 为右端点的区间中，$\min \geq i$ 的区间的最长长度；那么如果没有 l 那边的限制，答案就是 $i \times (A_i + B_i)$ 的全局最大值；进一步我们发现 l 那边的限制其实可以这样处理：我们钦定的 B 这一侧的最小值必须得严格大于整个询问区间 $[l, r]$ 的最小值，对于选取整个区间的情况单独考虑。这样可以发现区间就不会超出 l 了。

现在考虑如何维护 B，发现 $r \to r+1$ 的时候相当于把 $[1, b_{r+1}]$ 做区间 $+1$，以及 $[b_{r+1}, \infty)$ 覆盖为 0；以及还需要查询一个后缀中 $i \times (A_i + B_i)$ 的最大值，其中 A_i 是提前预处理的定值。这个问题可以使用分块配合凸包维护，做到 $O((n+q)\sqrt{n})$。实测是能过的。

- $O(n \log^3 n + q \log^2 n)$ 做法

我们考虑假如询问区间是 $[ql, qr]$，选取了笛卡尔树区间 $[l_i, r_i]$，笛卡尔树区间中的最小值为 m_i，那么此时实际的区间是 $[ql, qr] \cap [l_i, r_i]$。不妨考虑 $ql \leq l_i \leq qr \leq r_i$ 的情况，其贡献为

$$f_i(qr) = \max_{y \leq m_i} y \times (A_y + qr - l_i + 1)$$

其中 A_y 表示序列 a 中最长的一个区间，满足区间中的数全部 $\geq y$。

这是一个关于 qr 的函数，我们可以提前按照 m_i 递增扫描线，维护一个可持久化李超树就可以支持 $O(\log n)$ 单次查询出一个 $f_i(x)$ 的值。现在我们要查询所有满足 $ql \leq l_i \leq qr \leq r_i$ 的 i，他们的 $f_i(qr)$ 的最大值。

注意到一个性质：任意两个 i, j，他们的 $f_i(x), f_j(x)$ 两个函数至多只有一个交点。因此我们可以使用李超树来维护 $f_i(x)$ 的单点处最大值；对于 $ql \leq l_i \leq qr \leq r_i$ 的限制，扫描线扫 ql，每次遇到一个 $[l_i, r_i]$ 的时候做一个李超树区间插入，查询只需要在 $O(\log n)$ 个节点上求一次值。

最后把另一种情况反过来再做一遍，复杂度就是 $O(n \log^3 n + q \log^2 n)$。

（碎碎念：有人能猜到明天的题目名吗？）