对题解还有疑问可以私信 qq 1289494791。

最小极差 (range)

考虑将对每个前缀查询弱化成全局查询怎么做。

有一个显然的贪心，每次选出全局最大值进行操作，可以证明是不劣的。

考虑倒着求出 $k = n, n-1, \dots, 1$ 的答案，首先通过上述贪心解决 $k = n$ 的情况。

设当前我们求出了 $1 \sim k$ 这段前缀的答案，第 i 个元素被操作了 c_i 次，当前值为 v_i。

递推到 $1 \sim k-1$ 这段前缀时，第 k 个元素将从序列中被删去，并多出 c_k 次操作。

对于 $1 \le i \le k-1$，c_i 的值显然不会因此减小，进行 c_k 次在 $1 \sim k-1$ 中找 v_i 最大值进行操作即可。

但是 c_k 有可能很大导致时间复杂度无法接受，注意到若 c_k 被操作了 $> \log V$ 次就意味着剩下所有数已经被操作成 0，可以特判这种情况。

使用优先队列找出当前 v_i 最大值，时间复杂度 $O(n \log n \log V)$。

序列变换 (sequence)

luogu P13969，可以去看那题的题解。

城市旅游 (tour)

$type = 2$ 的原题是：P11704。

考虑 $type = 1$ 暴力怎么做：将每个名胜古迹归为第一条或第二条路径，然后直接用乘法原理把组合数乘起来即可。

但是有一个问题，就是可能会有名胜古迹同时被第一条和第二条经过，这样就会算重。

容斥，在交点处乘以 -1 的系数。这样就可以按照 $x_i + y_i$ 将关键点排序后跑一个 $O(n^2)$ 的 DP。

考虑 $type = 2$，相当于一条路径是 $(1,2) \to (n-1,m)$，另一条路径是 $(2,1) \to (n, m-1)$，套用 LGV 引理和上面的做法即可。

序列查询 (query)

操作分块，以 B 为块长，下面的复杂度分析都是对单块的。

类似 odt 维护连续段，每进行一次修改 / 查询操作分裂连续段，这样一个块内就有 $O(B)$ 个连续段。

对于每个连续段，记录这一段被哪些查询操作覆盖到，使用 `bitset` 存储，这部分时间复杂度 $O(B^2)$。

然后从小到大扫描值域 $0 \sim n-1$，找一下当前这个数在哪些连续段内出现过，把对应连续段的 `bitset` 求并。

如果 `bitset` 的并覆盖了所有查询就接着扫描，否则那些没有被覆盖的查询操作的答案就是当前值，暴力找出那些询问，并在所有连续段的 `bitset` 中删去。

`bitset` 求并的总次数不超过 $\sum\limits_{i=0}^{n-1} cnt_i = O(n)$，其中 cnt_i 为 i 出现的总次数，所以这部分时间复杂度 $O(\frac{nB}{w})$。

每个查询操作只会被删除一次，所以删除的时间复杂度是 $O(B^2)$。

所以单块内操作时间复杂度 $O(n+B^2+\frac{nB}{w})$，一共有 $\frac{q}{B}$ 个块，总时间复杂度 $O(\frac{nq}{B}+qB+\frac{nq}{w})$。

取 $B=\sqrt{n}$ 可得总时间复杂度 $O(\frac{nq}{w}+q\sqrt{n})$，空间复杂度 $O(n+q)$。