T1 集合

分析性质。

发现这个“好的区间”具有单调性——若区间 $[l,r]$ 是好的，则所有被它包含的区间也是好的，因为这样最长值域连续段长度不会变大，一定也满足条件。

考虑扫描线。将右端点 r 从推，因为有单调性，所以以 r 为右端点的好的区间的左端点一定是 $[1,r]$ 的一个后缀。令最小的的满足 $[l,r]$ 好的 l 为 L，则根据单调性，当 r 往右推的时候，L 不会变小。

因此就可以双指针了。维护当前区间的数集，从左到右扫描右端点，同时维护 L。每次加入 a_r，如果当前区间 $[L,r]$ 不合法，就删除 a_L 并令 `L++`，直到 $[L,r]$ 数集是好的。对于每个 r 都 `ans+=r-L+1`。

维护数集可以使用值域线段树。对于每个线段树节点维护该区间中最长的值域连续段，包含左、右端点的极长连续段长度，合并儿子节点是简单的。只需要支持单点修改即可。

时间复杂度 $O(n\log n)$。

T2 差后队列

签到题，显然不是最大值的数是等价的。

对于每次删空，实际上是多测，每一段单独处理。

对于每个数，找到后面第一个比它大的位置，然后每次删除时成功的概率实际上是 前面没删掉的概率 $\times\frac{1}{k}$。其中 k 是删除完后剩下的数的个数。

这个可以做个后缀和之类的维护。

对于每个位置，从前往后扫当一个数不是最大值后就可以把它加入贡献，每次遇到一个删除就用 $\frac{1}{k}$ 更新答案，然后整体乘以 $1-\frac{1}{k}$。

复杂度 $O(n)$。

T3 蛋糕

容易想到 dp 设 $f_{l,r,k}$ 表示区间 $[l,r]$ 内同时减去了 k 的答案。状态数 $O(n^3)$，暴力转移可以 $O(n)$。

更优秀的转移可以考虑枚举区间最大值的最底下的块如何划分。一种方法是选择最大值所在的一列；否则设选择区间 x,y 同时减去其最小值，容易发现这种操作改为先操作 $[l,r]$ 再操作 $[x,y]$ 代价不变，并且周围两边更小了，因此这样不劣。因此转移变成了 $O(1)$。

注意到 dp 中相同的值得顺序没有影响，给相同的值随意钦定一个顺序让它们不同。

考虑优化，如果你写一个哈希记搜发现就能过了。这是因为有效状态数实际上是 $O(n^2)$ 的。证明如下：

首先发现必要条件是 $l=1\lor a_{l-1}<k\lor a_{l-1}>\max_{i\in[l,r]}a_i$ 且 $r=n\lor a_{r+1}<k\lor a_{r+1}>\max_{i\in[l,r]}a_i$ ，具体来说这个条件是考虑缩小边界的时候要么用最小值更新 k 要么用最大值缩小，所以区间两个端点应该都要满足这两题条件。

然后发现对于每个 k，其满足条件的 $[l,r]$ 区间的数量实际上是 $O(n)$ 个的。实际上后面这个东西是相当于把所有 $a_i > k$ 的极长段建笛卡尔树。

具体找状态可以参考 std。

T4 字符替换

原题（洛谷P8885）

考虑一种常见的计算不同子序列个数的方法，维护总方案数和以每个字符结尾的方案数。在模 2 意义下每次在原串后面加入一个字符等价于交换以它结尾的方案数和总方案数。

然后考虑一个串，有哪些子串是好的，大眼观察我们可以发现可以将所有位置（两个字符之间和首尾）划分为字符集+1个等价类，等价类内两两可以得到一个好的子串。具体地，考虑 `a......a` ，从最左边开始，过了第一个 `a` 后dp中的 1 就被换走了，一直到下一个 `a` 才能换回到总数那里，因此这么一个子串是好的。然后将若干个这种结构拼在一起，也会是好的 `a......ab.....ba.....a` 。那么就容易发现这些位置构成一个等价类。

然后考虑维护每个等价类的大小，总共有4个等价类，每次加入一个字符会交换（当前等价类）和（【某个字符】的等价类），并将（当前等价类）大小+1

然后要么维护每个等价类大小模4，要么维护每个等价类大小模2和当前已经统计到的答案模2。总之就是状态数优化问题。

再套一个猫树，只有 01 复杂度 (64,8)， `012` 复杂度 (256,16)，但是std后面多乘了一个2。

多提一嘴，原题的其他做法都被012卡掉了。