浙江农林大学第二十七届程序设计竞赛题解

2024 年 10 月 20 日

您需要先同意参赛须知

- 图灵测试题，输出即可。

你是 gcd 吗

- 本题为签到题，既然是和所有数求一次 gcd，那么就是这个数的所有因数。从中找一个不为 1 的最小因数即可。

密文

- 构造题，每个 b_{a_i} 左侧有 $i-1$ 个 i，右侧有 $n-i$ 个 i。

密文

- 构造题，每个 b_{a_i} 左侧有 $i-1$ 个 i，右侧有 $n-i$ 个 i。
- 考虑贪心，先排序，对于小的点，要优先保证在左侧有足够数量的该数。这样从左到右扫一遍。

密文

- 构造题，每个 b_{a_i} 左侧有 $i-1$ 个 i，右侧有 $n-i$ 个 i。
- 考虑贪心，先排序，对于小的点，要优先保证在左侧有足够数量的该数。这样从左到右扫一遍。
- 对于大的点，优先保证在右侧有足够数量的该数，从右到左再扫一遍。
- 复杂度 $O(n^2)$。

奶龙

- 许多同学没有理解有一人相同则认为支持在题目中如何处理。样例说明已经指出，如果一个方案中的两人都被同一人指名，那个人也只算一票。
- 可以做一个容斥，先当成每个人单独计算票数，求出大于 p 的方案数，这可以用二分来处理。
- 然后对于每组被完全提名的方案，单独计算一次票数，如果本来大于 p 但处理后小于 p，则减去。
- 复杂度：$O(n \log n)$。

求求你回来吧，我什么都会做的

- 可以用差分实现，这里讲一个不用差分也能做的思路。
- 对于 $n > (k+1)*2$ 的情况，一定可以从全是 0 的数组中构造出一个单调增的数列，既然能构造出一组，那么把这组数列无穷次的加到原数列上，就可以构造出单调增的序列，因此一定可以。
- 对于 $n = (k+1)*2$ 的情况，每个数都有被更新的机会。我们可以发现，如果一个序列是单调递增的，那么一定可以根据题目中的变换改造成 $x, x+1, x+2, \cdots, x+n-2, y$ 的形式，x 是一个大许多的数，y 是一个更大的数。
- 你只需要构造一个这个形式的数列，让 x 取的很大，y 任意大，从左到右扫一遍，求出新数组和原数组的差，对于 i 和 $i+k$，如果差是左边大于等于右边，那么多出来的部分可以留给后面用，否则就不行。

求求你回来吧，我什么都会做的

- 对于 $k+1 < n < (k+1) * 2$ 的情况，和上一种情况类似，但中间有数无论如何都无法修改。因此如果中间无法修改的部分是单调递增的，构造出这些数左侧是尽可能大的差 1 的序列，右侧是尽可能小的单调递增序列，然后做和上一种相同的操作。
- 对于 $n < k+1$ 的情况，没有任何一个数能被修改，只能看原数组是否单调递增。
- 当然用差分实现思路也类似，复杂度：$O(n)$。

爆!

- 首先，想要冲击波发生装置覆盖的区域最大化，那么装置一定是放置在没有被覆盖过的边缘位置。那么每一次放置装置就相当于选择一条连续的横向或者纵向通道进行覆盖。
- 将每列每一组连续的通道作为一个点，将每行每一组连续的通道作为一个点。这样就得到了两个点集，而两个点集之间的边就是行列连续通道在图中的交点，这些边的数量等于图中的通道数量。
- 于是选择点集里的一个点（也就是图中的一个连续通道）放置冲击波装置，这个点连接的所有边会被覆盖，也就是图中在这个连续通道上的所有通道被覆盖。那么问题就从最少冲击波装置数量覆盖所有通道转化为选择最少的点使得边被全覆盖，而在二分图中最小点覆盖等于最大匹配数，这样就可以用匈牙利算法求解。

亵渎

- 模拟题，题面写的很长，但实现其实不难，因为是模拟所以要注意细节。
- 唯一的难点在于，模拟的过程中不能对当前在场上的所有随从都处理一次，这样复杂度就会来到 $O(n^2)$。
- 因此对于每个在场上的随从，可以记录当前的血量和随从在场上出现的时间，然后用两个优先队列来处理在场上的和死亡等待触发亡语的随从。
- 既然用优先队列不处理所有随从，那么可以不减去随从的血量而是累计已经造成的伤害，从而判断随从是否死亡。
- 树状数组也可以做。复杂度：$O(n \log n)$。

你是无法躲开这个忍术的。。。

- 状压 dp。注意到必须是质数的幂次，因此先筛所有数的质因子。
- 每个质数之间计数独立。对于一个单独的质数，可以把所有数里出现的次幂列出来，最高的次幂是 2 的次幂，在 log 级别，对于 $1e6$ 的大小最多 19 次。
- 对于这个数量级，可以使用状态压缩，用二进制位来表示对应的幂次是否有值。
- 对于一个二进制位是 1 的位置，左侧所有二进制位都右移那个次数，用这位右侧的数按位与上那个右移后的数，可以用这个状态 $+1$ 来更新现在的状态。
- 接下来就可以把每个素数所对应的状态枚举出来，用预处理的结果挨个累计。
- 复杂度：$O(n \log n)$。

小杨是大坏蛋

- 给定一个区间 l, r 区间求任选 n 个数字并分别除去他们的最大公因数再相乘的所有方案之积, 实际上只是求和问题。
- 对于分子上的乘积来说是可以直接被计算的分子上的最大公因数直接考虑反演即可。
- 注意指数上的运算可以运用两次整数分块, 时间复杂度为 $O(n \log n)$。

邋遢小黄

- 对于一整个序列来说，我们更希望能将其分成若干区间分别进行一定操作使得若干区间的条件符合题意进而得到一整个符合题意的区间。
- 这里有一个显然的贪心是，对于任意一个小区间来说里面每一个元素的所有同类型元素都要包含在这一个区间中。
- 且对于每个小区间来说它们对最终答案的贡献为 $num - max_i$ ，其中 num 为这段小区间的元素数量，max_i 为当前区间中出现次数最多的元素。（这里可以简单证明，将两个小区间合并一定不会使答案更优）

邋遢小黄

- 接下来我们考虑如何较快的将大区间分为若干小区间
- 对于某一个元素来说我们可以记录它出现的第一个位置与最后一个位置这里记作 l, r 我们可以将 $[l, r)$ 这个区间加上 1 , 通过这样标记之后显然区间中某个点若为 0 , 则该点至上一个 0 点或边界就是上述的小区间。
- 至此我们已经解决了区间划分的问题。这一点在后续我们修改的时候可以用 set 维护。

邋遢小黄

- 接下来我们考虑如何快速维护修改后区间内的答案, 这里我们可以对每一个元素开头的位置加上这个元素的数量并建立线段树维护最大值。
- 同时我们区分区间的方式是序列中 0 的位置, 因此建立线段树的同时我们维护 0 点的信息, 对于一个区间来说若其不存在 0 点则直接合并即可。
- 若存在 0 点可以通过两个区间比较大小的方式合并讨论左边或右边小以及两者相等时的情况, 以此确定合并方案。最后我们查询整颗线段树的区间即可统计答案即可。
- 时间复杂度 $O(nlogn)$ 。