JXOI2017-2018 解题报告

water_mi

2019 年 3 月 28 日

PS1：加红的字为可以跳转的链接或重点

目录

1 JXOI2017

1.1 sequence

1.1.1 Description

给定一个长度为 n 的整数数列 r_i，构造一个长度为 n 的整数数列 A，满足如下条件：

- $1 \le A_i \le r_i$
- 对于 $\forall 3 \le i \le n$，令 R 为 A_1 到 A_{i-2} 中大于等于 A_{i-1} 的最小值，L 为 A_1 至 A_{i-2} 中小于等于 A_{i-1} 的最大值，A_i 必须满足 $L \le A_i \le R$。

求共有多少不同的数列 A。
$1 \le n \le 50, 1 \le r_i \le 150$

1.1.2 Solution

假设我们当前需要确定位置 i 的值（假设我们已经知道了此时的 L, R），很显然有 A_i 与 A_{i-1} 都在区间 $[L, R]$ 之内（暂且先不考虑 $r[i]$ 的限制），所以对于 $\forall k < i-1$，必有 $A_k < A_i \& A_k < A_{i-1}$ 或者 $A_k > A_i \& A_k > A_{i-1}$，所以我们只需要考虑 A_i 到底是在 $(A_{i-1}, R]$ 内取值还是 $[L, A_{i-1})$ 内取值。

当然，我们可以令 $A_i = A_{i-1}$，这样一来很显然当你选择 A_{i+1} 时，$L = R = A_{i-1}$，所以此时整个后缀都相等了，当然，你在取 $A_i = L$ 或者 R 时（此时这个边界值一定是 A 中的某个数，否则根本取不到）也会使得后缀相等。

所以我们令 $f[x][l][r]$ 表示从左到右数第 i 位的取值（不考虑 $r[i]$）在区间 $[l, r]$ 内，数列 $[i, i+1, ..., n]$ 满足条件的方案数，则有转移方程（假设我们当前枚举到 x 的取值为 i）：

$$f[x][l][r] \text{+=} \begin{cases} f[x+1][i][i] & i \in \{l, r\} \\ f[x+1][l][i] + f[x+1][i][r] - f[x+1][i][i] & \text{otherwise} \end{cases}$$

边界：$f[n+1][][] = 1$

很显然需要从后往前转移，时间复杂度为 $O(nr^3)$。
Tag: 分析，区间动态规划

1.2 add

1.2.1 Description

现有一个长度为 n 的数列 A 和 m 个区间，你需要从中选出 k 个区间，然后对于每一个区间 $[L_i, R_i]$，令 $a_{L_i}....a_{R_i}$ 都加上 a，你需要最大化整个数列的最小值，输出这个最大的最小值，有多组数据。

$1 \leq \sum n, \sum m, T \leq 2 \times 10^5, 1 \leq a \leq 100, 1 \leq A_i \leq 10^8$

1.2.2 Solution

考虑二分，我们二分出整个序列的最小值 low，则我们需要通过选出 k 个区间来使得每个数的值都大于等于 low，将区间按照左端点为第一关键字，右端点为第二关键字排序，先从左到右将所有小于 low 的数放进队列中。

接着开始弹出队列，维护一个指针扫描给定的 m 个区间，对于每一个包含当前扫到的数的区间，我们在剩下的队列中找出最右侧且在当前区间内的数，然后将其放入到大根堆中，因为一次可能扫到多个满足条件的右端点，此时考虑贪心，用大根堆找出最右的数，将这个从当前数所在位置到这个最右数所在位置的值加上 a，重复此过程直到这个数字大于 low，如果此时添加此数大于 k 或者堆为空，则证明这个二分值为假。区间加单点查值可以用差分树状数组。

Tag: 二分，贪心

1.3 color

1.3.1 Description

给定一个长度为 n 的数组 A，你可以从中选出一些权值（两两不同），然后消去所有 A 中取值为你选出权值的数字，一种选择方案是合法的当且仅当存在剩下的数字且剩下的没被消去的数字有且仅有一段最长连续数列。

你需要求出所有本质不同的选择方案，两种方案被认为是不同的当且仅当存在至少一个权值只被这其中一种方案选出（不选（即 $\varnothing$）也是一种方案）。有多组数据。

$1 \leq T, \sum n \leq 3 \times 10^5, 1 \leq A_i \leq n$

1.3.2 Solution

1 40pts

很显然我们最后只会留下一段区间，考虑一种 $O(n^3)$ 的做法，先预处理出每一个子区间内每种颜色的个数，考虑一个区间 $[l,r]$ 是不是一个合法区间，我们只需要求出这个区间里的每种颜色在这个区间外面是否存在，如果存在则代表消去这种颜色后区间会断裂。

2 60pts

根据上面一种做法，我们可以顺着这种思路改进一下，先预处理出每种颜色 i 的被包含长度最小的区间，记为 $[L[i],R[i]]$，这样我们只需要判断一个区间内的颜色的这种区间是否在这个区间内，虽然这样子直接算还是 $O(n^3)$ 的，但是我们不妨固定一个区间的左端点 l。

先 $O(n)$ 地利用之前求出的 L,R 数组，预处理出 $awayl[i]$ 表示区间 $[l,i]$ 中最左侧的 $L[col[i]]$，$awayr[i]$ 表示最右侧的 $R[col[i]]$，接着我们开始枚举右端点 r，一个区间若是合法的则有 $awayl[r]\geq r$ 且 $awayr[r]\leq r$。

这样子做是 $O(\frac{n*(n+1)}{2})$ 的，可能有些复杂，也许有更简单的做法吧。

3 100pts

考虑倒着枚举左端点，然后维护其对应可用的右端点（正着枚举你怎么维护... 至少我不会），考虑到现在枚举到左端点 l，如果 $L[col[l]]$ 不是它本身则说明从这个点开始向右的点都不能作为当前这个区间的右端点，我们将它先加入到栈中（理由待会解释），否则的话，则说明区间 $[L[col[l]],R[col[l]]-1]$ 内的点都不能作为这个区间的右端点。

现在考虑我们之前开的栈有什么用，它存了许多之前找出的“从它开始向右都不能作为右端点”的点，但是可能这其中的一些点已经不再是这种点，所以我们开始弹栈（此时栈顶是栈中位置最左的点，如果连这个点都无法取，其它的点也不需考虑了），一个点 i 可以被弹出当且仅当 $L[col[i]]\leq l$。

现在我们找出第一个无法被弹出的元素，记它为 r（如果没有则令 $r=n$），显然，目前貌似可用的右端点个数就是 $r-l+1$，但是这些点只考虑了 $L[]$ 对它的影响，所以我们还需要减去一些考虑 $R[]$ 后的无用点，即先前我们所说的另一种情况，具体如何维护这种情况，我们需要做到区间标记和区间计算多少个被标记的点，可以用线段树来实现。

Tag: 固定一侧考虑另一侧贡献，转化思维，线段树

2 JXOI2018

2.1 sort

2.1.1 Description

有一个长度为 n 的数组，一种复杂度为 $O(n \times n!)$ 排序方式是将它随机打乱然后检查是否有序，如果是则停止，否则继续打乱。现在要你在这个数组后面插入 m 个大小在 $[l,r]$ 内的数，使得某种方案中的排序的期望打乱次数最多，输出这个次数。共有 T 组数据。

$T \leq 10^5, n \leq 2 \times 10^5, m \leq 10^7, 1 \leq l \leq r \leq 10^9$

2.1.2 Solution

期望打乱多少次后有序，就是要我们求出一次打乱就完成的期望的倒数，那我们考虑如何求出一个数组打乱多少次后有序，对于一个没有重复元素的数组，显然期望为 $\frac{1}{n!}$，如果有重复元素的话，由于这里没有对相对位置的限制，我们假设每个数字 i 出现了 $cnt[i]$ 次，则根据排列可以求出期望打乱次数为：$\frac{n!}{\prod cnt[i]!}$

现在我们要让期望打乱次数最大，就是使得 $\prod cnt[i]!$ 最小，所以我们需要求出怎么分配这 m 个数使得 $\prod cnt[i]!$ 最小假设现在我们有一个只含有 a，b 两个元素的数组，其中 $cnt[a] < cnt[b]$，则（只能添加 a,b 两个元素）我们需要比较加入一个 a 元素或 b 元素后哪个更小，作除法得：

$$\frac{(cnt[a]+1)! \times cnt[b]!}{cnt[a]! \times (cnt[b]+1)!} = \frac{cnt[a]+1}{cnt[b]+1} < 0$$

所以我们考虑每次加入 $[l,r]$ 内元素数量最少的元素。

1 50pts

考虑用堆来维护区间 $[l,r]$ 内元素的个数，每次取出最少的即可，复杂度与 m 有关，不可取。

2 100pts

对于数组内在区间 $[l,r]$ 内的元素计一个 cnt 表示出现次数，对这个 cnt 数组排序，考虑从左往右扫，对于 $cnt[i]$，我们将它及其前面的所有 cnt 填平到 $cnt[i+1]$ 的高度，答案乘上 $\left(\frac{cnt[i+1]!}{cnt[i]!}\right)^i$。若某次填平会导致操作次数

大于 m，设 c 为此时剩余操作次数，此时只要填平到 $\lfloor \frac{c}{i} \rfloor$ 的高度就可以了，然后再处理一下剩余的 $c\%i$ 次操作即可。

考虑到 l 和 r 的跨度可能较大，中间会有很多元素是原数组里没有的，所以我们计算出 $[l,r]$ 中在原数组里没有出现的元素个数 $rest$，在排序后的 cnt 数组头部加一个 0，并且给它附上一个系数 $rest$ 即可。

Tag: 数学，统计

2.2 game

2.2.1 Description

转化一下题意（原题面与现大部分 OJ 题面不同），不难发现在乘上 $n!$ 后题意变为：给一个区间 $[l,r]$, 每次选一个数，把它和它的倍数去掉，耗时为当所有数都被消掉时你选出的数的个数，问所有不同选择方式的总耗时。

$1 \le l \le r \le 10^7$

2.2.2 Solution

考虑排除 $l=1$ 的情况，此时每个序列的答案就是选到 1 的时候它所在的位置，因为 1 在你没选之前是不会被选到的，但是只要你一选到，整个区间就选完了，即 $(n=r-l+1)$：

$$\texttt{ans} = \sum_{i=1}^{n} i \times (n-1)! = \frac{(n+1)!}{2}$$

前面表示固定 1 所在的位置，然后计算剩下 $n-1$ 个数的位置。

接着考虑 $l!=1$ 的情况，此时我们每次整个区间内的伪素数（这里伪素数定义为约数只有它本身在区间 $[l,r]$ 的整数），可以使用埃氏筛法求出来。这些伪素数是必选的，设伪素数个数为 tot。

设 $f[i]$ 为第 i 次就能消掉所有数的方案数，则：

$$f[i] = tot \times C_{n-tot}^{n-i} \times (i-1)! \times (n-i)!$$

最后的答案是 $\sum_{i=1}^{n} i \times f[i]$

Tag: 数学，分类讨论，筛法

2.3 guard

2.3.1 Description

在一个平面直角坐标系上有 n 个点，第 i 个点的坐标为 $(i, h[i])$，我们称一个点 j 能被点 i 监视当且仅当 $j < i$ 并且这两点的连线不在它们中间任何一个点 $k \mid i \le k \le j$ 的下方（存在其中某个点落在直线上也是不行的），特殊地，一个点能监视它本身。现在令 $f[l][r]$ 表示区间 $[l, r]$ 至少要在这个区间内设置多少个特殊点才能保证整个区间内的点都能被某个特殊点监视到，求出 $\sum_{l=1}^{n}\sum_{r=l}^{n} f[l][r]$。

$n \le 5000, h[i] \le 10^9$。

2.3.2 Solution

用斜率来判断能否监视。考虑区间 dp，就拿上面的 f 数组（含义相同）来写吧。

与传统的区间 dp 不同，我们很难想到怎么合并某个区间的两个子区间。但是我们可以发现对于一个区间 $f[l][r]$，端点 r 是一定要被设置成特殊点的，所以我们照着这个思路下去可以发现如果对于端点 r，我们找到它所监视不到的最大区间 $[l', r']$，那么我们一定要在 r' 或者 $r'+1$ 出设置一个特殊点（就算 $r'+1$ 不能监视到，动态规划保证我们最后肯定会选出全局最优），所以：

$$f[l][r] = \sum_{l', r'} min(f[l'][r'], f[l'][r'+1]) + 1$$

这样子暴力转移是 $O(n^3)$（即70pts），考虑如何优化，我们可以发现对于一个固定的 r，这个最大的区间是不变的，所以我们顺序枚举 r，然后在每一次循环时逆序枚举 l，把每一个最大的区间 dp 值记录下来，最后更新一下即可。

Tag: 区间动态规划