NOIP2022 模拟赛

（2022.09.24　8:00～12:00）

命题人：某 NOI 金牌

解题报告

皮胚

题目大意

给定两个串 s, t，其中 t 包含特殊字符 '*' 和 '.'，分别可以将上一个字符复制若干次、匹配任意一个字符。求 s 有多少个前缀可以被 t 匹配。

算法 1

用 dp 处理匹配，令 $dp_{i,j}$ 表示当前匹配的指针为 s_i，t_j。

转移时按照 t_j 的情况讨论。

1. t_j 为 '.' 或 $s_i = t_j$ 则转移到 $dp_{i+1,j+1}$。

2. 若 t_j 为 '*' 且 $s_{i-1} = s_i$，转移到 $dp_{i+1,j}$。

3. 若 t_j 为 '*'，转移到 $dp_{i,j+1}$。

处理 dp 的时间为 $O(nm)$。

核冰

题目大意

给定 n 个数，每次可以选择两个相同的数 x 合并为 $x+1$，问最终最多有多少种不同的数字。同时维护对于某一个初始数的修改。

算法 1

在无修改时，有一简单的贪心策略：从小到大看所有的数字，在保证这个数字依然出现的前提下，尽可能地合并（也就是当 x 出现至少 3 次时进行合并）。

算法 2

考虑用线段树维护该贪心，修改操作可以看成是一次删除和一次插入操作，设每一个数出现的次数为 cnt_i。

对于插入操作，设插入一个数 x。

增加一个位置的值可能导致连续的多次修改，从 x 开始 $cnt_x = 2$ 的数都会在加入一个数之后被合并（最终 $cnt_i = 1$），最后一个数出现次数 +1。

对于删除操作，设删除一个数 x。

在删除 x 时，需要考虑从 $x+1$ 的位置取回一个数。$x+1$ 可以被取回的条件为：

1. 删除操作之后 $cnt_x = 0$。
2. 曾经合并得到 $x+1$ 过。

额外维护一棵线段树表示每一个数 x 合并的次数，在两棵线段树上二分即可找到位置。其余操作均可以归纳为区间加法。

方珍

题目大意

给定 n 个数组 $a_{i,j}$ 和一组常数 w_i，设每一个数列第 k_i 大的区间 mex 为 f_i。求 $\max\{f_i+w_i\}$。

算法 1

考虑如何对于一个数组求出 k_i 大 mex。

二分答案 m，求出所有 $\text{mex} \geq m$ 的区间数量。枚举右端点 r，尺取维护 $0 \sim m-1$ 都出现的最大左端点 l 即可。

复杂度为 $O(n^2 \log n)$。

算法 2

考虑先将所有的数组按照 w_i 降序排序，然后依次计算。维护一个答案指针 ans，每次对于一个数组，只需要判断 $ans+1-w_i$ 是否满足。由于 w_i 递减，容易发现 ans 最多增加 n 次。

故复杂度为 $O(n^2)$。

术劣

题目大意

给定一个序列，求有多少个区间排序之后为等差数列。

对于每一个前缀输出答案。

算法 1

枚举所有区间，排序判断，复杂度为 $O(n^3 \log n)$。

算法 2

枚举左端点，从小到大枚举右端点，依次插入 A_r，用一个 set 之类的数据结构来维护大小顺序。

再开一个桶维护不同差值的种类。

也可以倒着枚举 r，将插入 A_r 转化为删除 A_r，就可以用链表来维护删除操作。

判断每一个区间是否满足之后，做二维前缀和即可。

复杂度为 $O(n^2 \log n)$ 或者 $O(n^2)$。

算法 3

原问题等价于对于每一个右端点求有多少个左端点满足条件，考虑如何快速判断。

首先有一个简单的推论：

Lemma 对于一个数列 A_i（未排序），如果其满足条件，那么公差一定为 $\gcd(A_2-A_1, A_3-A_2, \cdots, A_n-A_{n-1})$。

得到公差之后，我们只需要判断 $\max\{A_i\} - \min\{A_i\}$ 是否恰好为 $d \times (n-1)$。

维护 $=0$ 较为困难，但是由于 $\max\{A_i\} - \min\{A_i\} \geq d \times (n-1)$，我们可以改成维护最小值以及最小值的个数。

依次处理每一个 r，用两个单调栈维护 max 和 min，每次弹出栈和加入栈时在线段树上做区间修改，即可得到 $\max - \min$。

在差分序列 B_i，对于每一个 r，$\gcd(B_l, \cdots, B_r)$ 最多变化 log 次（每次变化至少 /2）。

我们依然可以用一个栈来维护 gcd 的分段情况。每次加入一个数 B_r 时，就将栈内所有值改成 $\gcd(stk_i, B_r)$，如果有相同的就合并。栈内最多有 $\log n$ 个元素。

每次段内 gcd 发生改变时，就暴力枚举段内的每一个位置，在线段树上单点修改来维护 $\max - \min + l \cdot \gcd$。

由前面的分析，对于每一个左端点，其 gcd 发生变化的次数也为 $\log n$，故修改次数为均摊 $O(n \log n)$。

维护完上面的所有操作后，对于 gcd 相同的每一段，只需要查询 $\max - \min + l \cdot \gcd = r \cdot \gcd$ 的个数即可，即判断最小值是否满足条件。

复杂度为 $O(n \log^2 n)$。