CCPC Final 2025 题解

CCPC Final 2025 出题组

May 2, 2026

Statements

给你一个 $n \times m$ 的 01 矩阵，你可以交换相邻两个元素。问最多交换多少次，能从 A 变成 B。

$n \le 10^5, m \le 6$。

Author: apiad, Prepared by apiad & Qingyu

Solution

首先把问题建模成费用流，相邻两个点连容量为 ∞ 代价为 1 的边。

源点向 A 中的 1 连边，B 中的 1 向汇点连边。

求这个图最小费用最大流即可。

Solution

讲这个费用流对偶之后得到的问题为:

- 给每个点 (i,j) 一个标号 $x_{i,j}$。
- 相邻点的标号不超过 1。
- 最大化 A 中的 1 的标号和减去 B 中的 1 的标号和。

直观理解: 这些 1 按照这个标号进行“梯度下降”。

Solution

使用轮廓线 dp，记录一下当前每一列的标号 x。

不妨设这一列的最小值为 0，那么轮廓线上的标号只有 $O(3^m)$ 种不同的标号。

当做完这一列，如果发现最小值不为 0，那么可以整体对左边所有点的标号进行加减，使得这一列最小值为 0。因为对偶后的问题只和标号的相对大小有关。

时间复杂度 $O(nm3^m)$。

Solution

HoMaMaOvO 在比赛中发现，一定存在最优解，使得相邻两个点的标号差恰好为 1。

上述做法可以改进到 $O(nm2^m)$。

Statements

交互题。给一个随机的 1 到 n 的排列，你可以查询两个数 $\bmod 3$ 的距离，要求还原这个排列。

询问次数限制：$25n$。

$n \le 10^4$。

Author: apiad, Prepared by apiad

Solution

本题的查询限制略小于 $2n\log n$，目标是希望各种查询次数比这个更优的做法都可以通过。

基于数据随机性[1]，这个题存在很多种不同的做法，都可以通过。

最优的解法可以到期望 $\sim 13.5n$ 左右，而 $n\log n\sim 13.3n$。

[1] 如果不是随机，那么你也可以手动打乱排列询问。

Solution 1

首先可以通过 $O(n)$ 次询问把所有数分成三类，每类位置 $\bmod 3$ 相同。不妨设 A, B, C 三组，并且按这个顺序。

组内数字无法通过内部比较确定顺序。比较 $a \in A, b \in B$ 的时候，如果距离等于 1，那么 $pos_a < pos_b$，反之亦然。

可以首先合并 A, B，合并完之后对于 C 组能直接通过二分，进行插入排序。

Solution 1

关于合并 A, B，可以首先把 A, B 交替合并，依次插入。在插入过程中，维护段，段内部数字顺序不能区分，段与段之间按顺序，并且来组不同组。

每次插入一个数 x（不妨来自 A 组），先在 B 组对应的段里面二分，每个段选择一个代表元素，找到这个数会插在哪两段之间。

然后暴力枚举 B 的两个段，判断 x 把这两个组的元素如何划分成两个部分，将这些元素重新划分成若干段。

因为随机，所以二分到位置的两边的段大小是 $O(1)$ 的。

比较容易发现这个做法询问次数是 $n\log n + O(n)$ 的。

Solution 2

类比快速排序，每次选择一个元素 x，然后比较另一组元素和 x 之间的大小顺序。

然后继续分治，实现细节留给选手自己补全。

这个做法询问次数是 $O(n\log n)$ 的。根据实现，可以做到 $18n \sim 20n$。

Statements

给定一个线性方程组 $\sum_{i=1}^{4} x_i a_i \equiv m \pmod p$，求非负整数解，并且最小化 $\sum_{i=1}^{4} x_i b_i$。

$p \leq 1.01 \times 10^9$。

Author: apiad, Prepared by apiad

Solution

引理：存在一组最优解，满足 $\prod(x_i+1) \le p$。

反证法，记最优解为 X，如果有多个解，取字典序最小的。

如果大于 $\prod(x_i+1) > p$，根据抽屉原理，一定存在两个向量 V_1, V_2 满足 $V_{i,j} \le X_j$，并且 $A \cdot V_1, A \cdot V_2$ 相同。

令 $\Delta V = V_1 - V_2$，那么 $X + \Delta V, X - \Delta V$ 都是合法解，一定值更小，或者字典序更小。

Solution 1

先枚举四个数的大小顺序，不妨有 $x_1 \le x_2 \le x_3 \le x_4$。
那么有 $x_1 \le p^{1/4}, x_2 \le (\frac{p}{x_1})^{1/3}, x_3 \le (\frac{p}{x_1 x_2})^{1/2}$。
枚举 x_1, x_2，一共有 $O(p^{1/2})$ 对，再枚举 x_3，有 $x_3 \le p^{1/2}$。
x_4 可以被表示成 x_1, x_2, x_3 的线性关系，可以用折半搜索快速完成这两部分的合并。
时间复杂度 $O(\sqrt{p}\log p)$。

Solution 2

先枚举四个数的大小顺序，不妨有 $x_1 \le x_2 \le x_3 \le x_4$。
那么有 $x_1 \le p^{1/4}, x_2 \le (\frac{p}{x_1})^{1/3}, x_3 \le (\frac{p}{x_1 x_2})^{1/2}$。
直接枚举 x_1, x_2, x_3，根据计算，不会超过 $O(p^{3/4})$ 组。
时间复杂度 $O(p^{3/4})$，优秀的常数实现可以通过。

Solution 3

枚举最大的元素，不妨设为 x_4，剩下三个元素不超过 $\sqrt{p}$。
令 $M = \lceil p^{1/4} \rceil$。
对 $(x_1, \lfloor x_2/M \rfloor), (x_2 \bmod M, x_3)$ 两部分折半。
时间复杂度 $O(p^{3/4})$，优秀的常数实现可以通过。

Statements

给你一个字符串 S，有 n 种操作，每次可以在当前串的无限循环串，找到一个长度为 l_i 的前缀。

你可以按任意顺序执行这些操作，问字典序最大的串是多少，并且给出排列。

$n \le 10^5, l_i \le 10^9$。

Author: Sugar & Qingyu, Prepared by Qingyu & Sugar

Solution

如果这个串被更短的长度截断了，那么只需要关心更短的长度。

所以只需要关心操作序列后缀最小值即可，并且把最大值放在最后一定更优。

将所有操作按长度排序。在做的过程中，我们只关心当前串的无限循环结果。并且因为最后的答案串串长是个定值，所以循环串对应的字典序更大一定是更优的。

所以可以贪心，每次比较当前串的无限循环串与把这个串截取前缀后无限循环的串的字典序，保留更大的。

Solution

对于两个字符串比较字典序，我们可以考虑用二分 + Hash 的方法。

考虑当前串是 s_0 经过 $l_1, l_2 \dots l_k$ 后得到的，我们希望求出其前 k 位的哈希值。

找到第一个 $l_i \le k$，则我们需要前 l_i 位的串重复 $\lfloor \frac{k}{l_i} \rfloor$ 次的哈希值，和前 $k \bmod l_i$ 位的哈希值。

前者可以在加入一个 l_i 时预处理，询问时快速幂；后者取模只会发生 $O(\log l)$ 次。

Solution

快速幂这部分，我们可以把结果表示为 $\frac{B^{lx}-1}{B^l-1}$，分子部分可以使用根号预处理，$O(1)$ 查询。分母部分只有 n 个不同的 l，预处理所有的逆元即可。

所以整体时间复杂度为 $O(n\log^2 l)$。

如果直接快速幂，做法为 $O(n\log^3 l)$，也可以通过。如果用可持久化平衡树维护当前字符串，也可以通过。

Solution

关于 log 次二分，我们可以在 Trie 上面二分，如果当前数字在子树的 max 和 min 之间就递归下去。并且预处理前缀一堆 0 走到的节点。

如果递归下去了 k 层，那么数字一定变小了 $1/2^k$。均摊也是一个 log 的。

但是这部分不是瓶颈，所以这部分三个 log 也不显著影响效率。

Statements

给定一个字符串 s，求其有多少个子串 $[l, r]$ 是 Lyndon 串。
Lyndon 串定义为比所有真后缀字典序均小的字符串。

$|s| \le 2 \times 10^5$。

Author: unputdownable & apiad, Prepared by unputdownable

Solution 1

考虑分治，对于跨过分治中心 m 的区间 $[l,r]$，将开头在 m 之前和之后的后缀的限制分别处理。

对于在 m 之后的后缀的限制，实际上是要求 $s_{[m,r]}$ 的最小后缀的字典序大于 $s_{[l,r]}$。求出 $s_{[m,r]}$ 的最小后缀，由于区间 $[l,r]$ 的长度大于最小后缀的长度，所以最小后缀大于 $s_{[l,r]}$ 等价于最小后缀大于 $s_{[l,n]}$，即右端点 r 对左端点 l 的限制为 $rk_l \leq lim_r$。

对于在 m 之前的后缀的限制，每个 $s_{[l,m)}$ 的后缀 t 的限制为 $s_{[l,r]} < t + s_{[m,r]}$。这个限制一定要么对于所有 r 都不成立，要么对于某个后缀的 r 成立，即左端点 l 对右端点 r 的限制为 $r \geq lim_l$。

Solution 1

求出限制后的计数是二维数点，整体时间复杂度 $O(n\log^2 n)$。

对于右端点的限制，用 Duval 算法可以求出所有前缀的最小后缀，在后缀树或后缀数组上容易求出最小后缀的 rk，这部分的时间复杂度为线性的。

对于左端点的限制，首先若 $s_{[l,m)}$ 不是 Lyndon 串或者 pre-Lyndon 串（某个 Lyndon 串的前缀），则对于任意的 r 均不满足条件。

否则只需要考虑其所有 Border，剩余的后缀均大于原串。可以证明，所有限制中，一定是最长 Border 的限制最严格，于是只用求出每个后缀的最长 Border，然后进行一次最长公共前缀查询即可求出限制，时间复杂度同样为线性的。

Solution 2

对于每个 Lyndon 串 s，找到其字典序最小的真后缀 w 可以将 s 分解为两个 Lyndon 串 vw。递归下去直到长度为 1 可以得到一个树形结构，其本质是后缀数组的笛卡尔树。

可以证明，对于 Lyndon 子串 $[l, r]$，求出 Lyndon Tree 中点 $[l, l]$ 和 $[r, r]$ 的 LCA，设其对应区间 $[l_u, r_u]$，儿子节点对应区间 $[l_u, m_u)$ 和 $[m_u, r_u]$，则一定有 $l = l_u$，$m_u \leq r \leq r_u$。

由于区间 $[l_u, r_u]$ 是一个 Lyndon 串，可知满足 $l = l_u$，$m_u \leq r \leq r_u$ 的区间 $[l, r]$ 是 Lyndon 串的充要条件为其没有 Border。

Solution 2

可以证明，若 $[l, r]$ 有 Border，则其最小 Border 的长度一定不超过 $\min(m_u - l_u, r - m_u + 1)$。

于是可以暴力用 KMP 求出区间 $[l_u, m_u)$ 的每个前缀有没有 Border，对于每个没有 Border 的前缀二维数点，可以求出最小 Border 为该前缀的串数量，即可不重不漏地统计所有存在 Border 的串个数。

对于节点 u 统计答案的时间复杂度为 $O(\min(m_u - l_u, r_u - m_u) \log n)$。由树链剖分的结论可知总时间复杂度为 $O(n \log^2 n)$。

Statements

给定一个长度为 n 的，包含 n 种左右括号和问号的括号序列，求有多少个子段存在一种替换问号的方式，使得每种括号的子序列分别构成合法括号序列。

$n \leq 2 \times 10^5$。

Author: Sugar, Prepared by: Qingyu & Sugar

Solution

一个子段合法的充要条件为：

- 子段的长度为偶数；
- 对每种括号消除后得到的形如))..))((..((的括号，将每个左括号匹配一个其左侧的问号，每个右括号匹配一个其右侧的问号，存在一组匹配。

对于左括号和右括号的匹配，可以分别贪心。容易证明上面的条件二等价于左右括号均可以分别匹配，且问号的总数不小于需要匹配的括号个数。

Solution

两个方向分别做扫描线可以求出每个左端点第一个不合法的右端点，和每个右端点第一个不合法的左端点。

统计答案的时候，扫描线维护区间的问号个数与要匹配的括号个数的差。需要支持区间加减权值，单点修改，区间查询某个奇偶性的权值非负的位置个数。这是一个经典的可以分块解决的问题，时间复杂度 $O(n\sqrt{n})$。

Statements

给一个 1D 的 Fillomino 谜题，判断是不是有解并且构造。

$n \le 10^6$

Author: apiad, Prepared by apiad

Solution

如果存在一段完全没有线索，那么可以证明一定可以用不超过 4 的数字。

如果一段长度大于 4，那么把它分成两组有至少 4 种方法，一定存在一个合法的分组方法。

$dp_{i,j}$ 表示前 i 格，是否可以分成若干段，使得最后一段长度为 j。

根据前面的讨论，只需要保留 $j = 1, 2, 3, 4$ 以及前一个非零的数字即可。转移的时候这一段长度，只需要能够快速判断一段内是不是只有一种数字即可。

时间复杂度为 $O(n)$。

Statements

给定一棵 n 个点的树，对于值域为 n 的序列，每次可以选两个任意位置上的数，若对应标号的点在树上相邻则可以交换。对于 $\ell = 1$ 到 m，输出长度为 ℓ 的可排序的序列数量，对 $10^9 + 7$ 取模。

$n \le 200$，$m \le 10^5$

Author: Kevin, Prepared by Kevin

Solution

找到所有在序列中出现至少一次的数在树上构成的连通块，每个连通块内部可以任意交换两个位置。所以只需要每个连通块内部排序之后整个序列有序即可。

也就是说，只需要对于每个连通块内部求出确定顺序的方案数，然后直接相乘即可，可以把答案的生成函数写成：

$$\sum_{\text{连通块划分}\mathcal{P}} \prod_{\text{连通块}\mathcal{C}\in\mathcal{P}} \sum_{t=0}^{\infty} S(|\mathcal{C}|, t)x^t$$

其中 $S(n, k)$ 是第二类斯特林数。

Solution

展开斯特林数的 OGF 得到：

$$\sum_{\text{连通块划分}\mathcal{P}} \prod_{\text{连通块}\mathcal{C}\in\mathcal{P}} \sum_{i=0}^{|\mathcal{C}|} (-1)^{|\mathcal{C}|-i} \binom{|\mathcal{C}|}{i} \frac{1}{1-ix}$$

由于所有连通块的大小和不超过 n，所以最终分式中分母里 $(1-ix)$ 项至多只出现 $\lfloor \frac{n}{i} \rfloor$ 次。又因为分子的次数不超过分母，所以分子分母的次数都是 $O(n\log n)$ 级别的。

树上背包的时候暴力维护分式可以做到 $O(n^3\log^2 n)$ 且需要卷积。修改成对分子插值即可优化到 $O(n^3\log n + n^2\log^2 n)$。

最终展开分式求答案直接暴力可以做到 $O(nm\log n)$。

Statements

给定一棵 n 个点的树，给每个点分配一个权值使得权值构成 1 到 n 的排列，且树上相邻的两个点权值和不超过 $n+1$，求方案数对 4 取模的结果。

$n \le 10^6$。

Author: Kevin & zhoukangyang, Prepared by Qingyu & Kevin

Solution

以 $1, n, 2, n-1, 3, n-2, \ldots$ 的顺序考虑所有点，问题变成，n 轮每轮选一个点删去，要求第偶数轮选择的点的邻居已经全部已被删去。

特判 $n=1$ 后，由于最后两个点的对称性，显然答案是 2 的倍数。考察每个奇数轮操作选点后，孤立点的数量。由于孤立点之间是对称的，所以这个数量永远不会到达 4，且不会在最后一步操作之前到达 2，否则一定会让方案数成为 4 的倍数。

Solution

于是可以讨论出有限种孤立点数量序列：

- n 为奇数时：$[1, \ldots, 1, 2, 0]$；
- n 为偶数时：$[1, \ldots, 1]$；
- n 为偶数时：$[1, \ldots, 1, 3, 1]$。

对于前面的若干个 1，每个奇数轮操作和紧接着的操作选择的两个点一定相邻，于是三种情况分别对应的树结构为：一个叶子个数为 $1, 2$ 或 3 的菊花，拼上若干组匹配。

整个删点的过程为：先删去外层的所有匹配，最后处理中心的菊花。

Solution

在删去外层匹配的过程中，如果有一步操作把树分成了至少两个连通块，则被分出去的连通块的最后一次操作一定会提供 2 的系数，而菊花本身的操作方案数也一定是偶数，于是这种情况总方案数一定是 4 的倍数。

故一定是按照从叶子到菊花的方向删除，这时显然每个匹配会在奇数轮删去靠近菊花的点，在偶数轮删去另一个点。

枚举菊花的中心之后换根 DP 计算拓扑序数量即可。时间复杂度 $O(n)$。

Statements

在 n 个回合中，第 i 个回合首先将中毒层数增加 x_i，然后选择是否将触媒层数增加 1。回合结束时，触发触媒层数次中毒，每次若中毒层数非负，造成等量伤害然后将层数减一。

$n \leq 5000$，$\sum n^2 \leq 10^8$。

Author: Gellyfish, Prepared by Gellyfish

Solution

直接给出结论：上触媒的位置一定是一个后缀，或者一个后缀去掉某个点。

设上毒的位置记为 1，不上毒的位置记为 0，那么是否上毒的序列构成了长度为 n 的 01 串，此时：

- 如果序列的末尾为 0，直接将末尾替换为 1 一定不会变劣。
- 如果序列中存在 10，则：
- 若在该 10 后存在中毒触发失败，即触发中毒时中毒层数为零的情况，则将 10 调整为 01 定不会变劣。
- 否则如果后续所有中毒均触发成功。则若在 10 后存在 01，即存在形如 ...10...01... 的结构，则将两者交换，改为 ...01...10... 一定不会变劣。

Solution

重复如上调整，容易发现一定可以将任意一个 01 串调整为我们想要的结构，即一个后缀或后缀扣一个点。

对于是一个后缀的情况，可以暴力枚举所有后缀处理。

而对于后缀扣一个点的情况，可以考虑先枚举是从哪个后缀扣的点，然后从后往前扫扣了哪个点，并均摊维护当前毒的触发情况即可。

时间复杂度 $O(n^2)$。

Solution

对第二种情况可以做进一步的简化。

根据前面的交换论证，如果某次触发时中毒层数已经为零，它可以被调整成不劣的纯后缀方案。

所以在考虑后缀扣一点方案时，可以认为所有触发都成功。即如果某次触发时中毒层数已经为零，也直接把它扣到负数。这样并不会影响答案。

所以这部分可以使用前缀和的方法直接计算，不需要真正维护毒的触发情况。

Statements

对于一个序列，每次操作可以将第 i 个位置修改为第 1 到 $i-1$ 个位置上所有值的 mex。求进行若干次操作后序列末尾元素的最大值，和达到最大值需要的最小操作次数。

$n \le 10^6$。

Author: Coffee_zzz, Prepared by Qingyu

Solution

若末尾元素不超过 $n-1$，无法通过任何操作使得末尾元素超过 $n-1$，于是最大值为 $\max(a_n, n-1)$。

若 $a_n \geq n-1$ 则操作次数最小显然为 0，否则一定需要将前面的 $n-1$ 个位置操作成 0 到 $n-2$ 各一个。可以证明，贪心从前到后扫，每次遇到一个位置的值超过 $n-2$ 或者与前面重复，就对该位置进行一次操作。

时间复杂度线性。

Statements

给你 n 个数，问有多少个本质不同的由一段区间的数字生成的的线性空间。

$n \le 10^5, 0 \le a_i < 2^{32}$

Author: Sugar, Prepared by Sugar & Qingyu

Solution

首先，扫右端点，用贪心的方法维护区间线性基，会得到一共 $O(n \log W)$ 个线性基。

接着对每个线性基，考虑消出它的规范型（即唯一表示），然后进行 Hash 去重。

通过贪心方法得到的线性基，按下标顺序插入每个数字。插入一个数的时候，可以先用更低位的数字消这个数，然后用这个数去消掉更高位的数字，在 $O(\log W)$ 的时间复杂度得到新的标准型。

整体时间复杂度 $O(n \log^2 W)$。

Statements

有一个二维的网格，你要支持在线删点，查询割点个数。

$n, m \le 500$, $q \le nm$。

Author: bananabot & nzhtl1477, Prepared by bananabot & nzhtl1477 & apiad

Solution

考虑把删除的点当成障碍，维护障碍的八连通块。

一个点是割点，当且仅当它的八相邻格子中存在两个格子来自于同一个障碍块，并且这两个障碍块被四相邻的格子隔开。

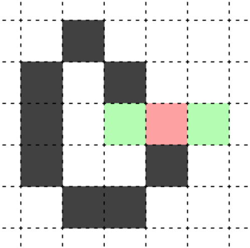

Figure: 割点的例子

Solution

启发式合并维护障碍的的八连通块。每次合并之后，如果这个格子八相邻的格子中存在刚合并的障碍块，才有可能成为新的割点。

合并以后，暴力检查小块所有的八相邻格子，以及新加的障碍的四相邻格，更新这些格子是不是割点。

时间复杂度 $O(q\log q)$。