

第 10 届中国大学生程序设计竞赛女生专场 题解

杭州电子科技大学

2024 年 11 月 3 日

Overview

	Easiest											Hardest	
Guess	A	C	M	H	L	E	K	G	F	B	I	D	J
On Site	A	C	H	M	L	E	G	F	K	B	D	I	J

A. 盒子

Description

给出三维坐标都与空间直角坐标系 $O - xyz$ 的三条轴平行的长方体盒子的底面 z 坐标 z_0 , 高度 h , 底面对角线坐标 $(u_0, v_0), (u_1, v_1)$ 和 q 个点 (x_i, y_i, z_i) , 问每个点是否在盒子内。

数据范围：

$$1 \leq q \leq 10^3, 0 \leq z_0, h \leq 10^6, -10^6 \leq u_0, v_0, u_1, v_1, x_i, y_i, z_i \leq 10^6$$

Solution

直接将盒子变成三个维度上的区间限制，询问的时候每一维分别处理即可。

时间复杂度 $O(q)$ 。

C. CCPC

Description

给定字母构成的字符串 S ，问重排列后子串 CCPC 最多的个数。

数据范围： $1 \leq |S| \leq 10^6$

Solution

统计字符串 S 中 C 和 P 字符的个数 cnt_C, cnt_P 。

若 $cnt_C < 3$ 或 $cnt_P < 1$ 则答案为 0。

否则答案为 $\min(\lfloor \frac{cnt_C-1}{2} \rfloor, cnt_P)$ 。

时间复杂度 $O(|S|)$ 。

H. 平方根

Description

给一个 01 串 s ，将若干 1 改成 0 后最大化所有极长的由 1 构成的连续段的长度开方和。

数据范围： $1 \leq |s| \leq 10^6$

Solution

每一段连续的 1 之间是独立的，我们只需要关心一段连续的 1 的结果。

可以证明对于一段连续的 1，最优策略是将其划分成多个单独的 1 以及可能余下的连续两个 1。具体来说，对于 k 个连续的 1，如果 k 是奇数，最优策略是将其划分为 $\frac{k+1}{2}$ 个单独的 1，答案为 $\frac{k+1}{2}$ ；否则最优策略是将其划分为 $\frac{k}{2} - 1$ 个单独的 1 和两个连续的 1，答案为 $\frac{k}{2} - 1 + \sqrt{2}$ 。

时间复杂度 $\mathcal{O}(|s|)$ 。

M. 覆盖一棵树

Description

给定一个 n 个点的树，使用一些线段覆盖树的边，要求最小化线段长度的最大值。

数据范围： $2 \leq n \leq 10^5$

Solution

解法一：贪心。

对每个叶子按照深度排序后，深度小的先覆盖其所有没有被覆盖过的祖先结点。简易证明：对于一个边，如果是两个叶子到根的必经之路，且目前由较深的叶子来覆盖此边，则可以替换成由较浅的叶子覆盖， $\max$ 线段长度不会增加。

时间复杂度 $\mathcal{O}(n \log n)$

解法二：树形 dp。

令 dp_u 为 u 的子树下还没有覆盖的叶子的深度。合并时，当子树个数 ≥ 2 时，则

$ans = \max(ans, \text{第二浅没合并的深度} - \text{当前的深度})$ ，只留下最浅的叶子。到根时，答案再对取最浅的叶子取 $\max$ 。

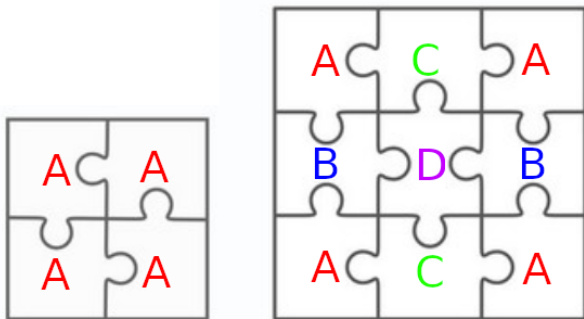
时间复杂度 $\mathcal{O}(n)$ 。

L. 拼图

Description

如图所示，给出拼图的 4 个基本块个数，问最多用多少块拼成矩形。要求矩形边无凹凸，每个基本块相邻的边必须凹凸。

数据范围： $1 \leq a, b, c, d \leq 10^3, 1 \leq T \leq 10^4$



Solution

矩形的边无凹凸，所以四角只能使用 a 块，四边只能使用 b 、 c 块，中间只能使用 d 块。发现 b 、 c 块必须相邻使用， a 块相邻的不能同是 b 或同是 c 。从顺时针角度考虑， bc 交替使用，因此 bc 使用个数一定相同。其余中间总是可以使用 d 填满。

因此可以枚举矩形的长，计算在该长度下，矩形最大的宽度。

时间复杂度 $\mathcal{O}(Td)$ 。

实际上，可以直接枚举矩形的长与宽，因为 $d \leq 10^3$ ，根据整除分块等知识，可知合法的长宽数量级别为 $\mathcal{O}(d\sqrt{d})$ 。

时间复杂度为 $\mathcal{O}(Td\sqrt{d})$ 。

E. 重心树

Description

有一棵 N 个点的树，给定每个点的儿子子树的重心，还原出一颗符合条件的树。

数据范围： $1 \leq N \leq 2 \times 10^5$

Solution

注意到其实重心并不重要，其实只需要知道他是子树内的一个点，然后找到这个点所在连通块的根即可。

根据 $p_i < i$ 的这个性质，我们只需要倒着枚举，就可以从下到上依次连边，使用并查集维护连通块以及连通块的根，这个连通块所在的根其实是整个连通块的最小值。

时间复杂度是 $\mathcal{O}(n\alpha(n))$ 。

G. 递增序列

Description

给定长度为 n 的数列 a 和整数 k , 询问有多少整数满足 $x \in [0, k]$, 且有 $a_1 \oplus x, a_2 \oplus x, \dots, a_n \oplus x$ 是单调不降序列。

数据范围: $1 \leq n \leq 2 \cdot 10^5, 1 \leq k \leq 10^{18}$

Solution

只考虑任意相邻的两项什么时候满足 $a_i \oplus x \leq a_{i+1} \oplus x$, 容易发现充要条件是从高到低找到 a_i 和 a_{i+1} 第一个二进制下不同的位置 pos , 若 a_i 在 pos 这一位为 0, 则 x 这一位必须为 0, 若 pos 这一位为 1, 则 x 这一位必须为 1。

问题转变为有多少个整数 $x \in [0, k]$, 有些位在二进制下为 0, 有些位在二进制下为 1,

类似数位 DP 将 k 拆分, 例如 $k = (10100)_2$, 可以拆成 $(0xxxx)_2, (100xx)_2, (10100)$ 共 $\log$ 个前缀, 每个前缀的若干位固定, 后面可以任取 0/1, 可以直接统计答案。

时间复杂度 $\mathcal{O}(n \log V)$, 其中 V 为值域。

F. 完全平方数

Description

给出一个长度为 n 的正整数序列 a , 求所有序列 d 满足 $d_i | a_i$ 且 d_i 的乘积为完全平方数的乘积开方之和。

数据范围: $1 \leq n \leq 10^6, 1 \leq a_i \leq 10^6$

Solution

注意到实际上每个质因子 p 的贡献是独立的。具体来说，对于某个质因子 p ，我们构造 $a'_i = p^{k_i}$ ，其中 k 是最大的非负整数满足 $p^k | a_i$ ，对序列 a' 也就是只提取质数 p 的幂次部分做原问题的答案为 $F_p(a)$ ，那么答案就是所有 $F_p(a)$ 的乘积。

考虑怎么求 $F_p(a)$ ，设其指数序列为 c ，即 a' 序列为 $p^{c_1}, p^{c_2}, \dots, p^{c_n}$ ，注意到 $c_i = 0$ 是没有用的，有效的项总共只有 $\mathcal{O}(n \log n)$ 项。

Solution

对于一个指数序列 c , 选择序列 q 满足 $0 \leq q_i \leq c_i$ 且 q_i 之和为偶数, 此时 q 对 $F_p(a)$ 的贡献为 $p^{\frac{\sum_i q_i}{2}}$ 。

对这个东西很容易想到直接做背包 DP 计数, 设 $f_{i,j}$ 表示前 i 项 q 之和为 j 的方案数, 最后根据第二维确定贡献计算答案。

进一步思考, 不妨把一次性加 q_i 的过程拆成加 q_i 个 1 的过程, 背包 DP 时直接计算答案: 在 j 转移到 $j+1$ 时, 若 j 为奇数乘 1, j 为偶数时乘 p , 还原成一次性加 q_i 的转移也是容易的。

此时注意到第二维只有奇偶性有效, 可以压缩, 我们就成功将上述做法的时间复杂度降为 $\mathcal{O}(n \log n)$, 足以通过本题。

K. 小凯的省奖之梦

Description

一共有 n 个人，每个人每个学期都有智育、德育、体育三项得分，每个学期会根据这三个得分按照一定规则评选奖学金，在评完两个学期的奖学金之后会根据大家的奖学金获得情况，综测总分和智育总分决定省政府奖学金的得主。现在告诉你所有人这两个学期的成绩，你可以花费 p 代价提升小凯第一学期智育分 1 分， q 代价提升第二学期智育分 1 分，智育分上限 100 分，问小凯花费最少多少的代价能够获得省政府奖学金。

数据范围： $6 \leq n \leq 500, 0 \leq p, q \leq 100, 0 \leq a_{i,j}, b_{i,j} \leq 100$

Solution

首先枚举提升第一学期智育分多少分，第二学期智育分多少分。然后我们可以通过构建结构体存储这些信息，通过排序确定第一学期的奖学金获得情况、第二学期奖学金获得情况，以及省政府奖学金的获得情况，从而判断在这个情况下小凯能不能获得省政府奖学金，记录最小的代价即可。

可以在结构体的时候通过直接记录 *id* 的方式来记录排序后的每个人他们一开始的编号是多少，从而方便后面的奖项分的统计，进而避免使用 `std::map`。

同时注意到字符串的字典序比较较慢，故你可以预处理出两个字典序的大小关系存入结构体中（相当于求出了每个人的名字在所有人名字中的字典序排名），在比较时直接比较这个排名即可。

时间复杂度 $\mathcal{O}(100^2 \cdot n \cdot \log n)$ 。

B. Aho-Corasick 自动机

Description

求 n 个点字符集大小为 m 深度不超过 d （根的深度为 0）的 AC 自动机数量。

数据范围： $1 \leq n, m, d \leq 100$

Solution

首先注意到答案实际上是在数 Trie 的数量，与 Aho-Corasick 自动机无关。

对于这样一个 Trie 的结构，我们可以分层 DP，每次增加一层。设 $f_{i,j,k}$ 表示前 i 层包含 j 个点，其中最底下一层包含 k 个点的方案数，转移的时候只需要考虑枚举下一层的点个数 t ，需要乘上转移系数 $g_{k,t}$ ，这个转移系数并不能直接计算。

预处理转移系数 $g_{k,t}$ ，发现上一层每个点可以在下一层延伸出 $0 \sim m$ 个点，假设延伸出 s 个点，则转移系数为 $\binom{m}{s}$ ，做背包即可。

时间复杂度 $\mathcal{O}((\max\{n, m, d\})^4)$ 。

D. 优秀的拆分

Description

有一个长度为 N 的排列 P ，将 P 拆成 A 和 B ，求 A 的最长上升子序列的长度与 B 的最长下降子序列的长度相加的最大值。

数据范围： $1 \leq N \leq 2 \times 10^5$

Solution

对于给定的排列 P ，设 LIS 为 P 的最长上升子序列长度， LDS 为 P 的最长下降子序列长度。

可以发现答案一定是 $LIS + LDS$ 或者 $LIS + LDS - 1$ 。

减一的情况是：无论怎么选，上升子序列和下降子序列必定存在一个交点。

首先我们可以传统的对最长上升子序列计数。

令 f_i 表示以 i 结尾的最长上升子序列的长度， g_i 表示以 i 结尾的最长上升子序列长度为 f_i 的方案数， F_i 表示以 i 结尾的最长下降子序列的长度， G_i 表示以 i 结尾的最长下降子序列长度为 F_i 的方案数。

Solution

不妨令 $c_1 = \sum_{i=1}^n g_i[f_i = LIS]$, $c_2 = \sum_{i=1}^n G_i[F_i = LDS]$ 。

那么 $c_1 \cdot c_2$ 表示任意选择 LIS 和 LDS 的方案数。

令 h_i 表示必选 i 的情况下，最长上升子序列有多少种方案， H_i 表示必选 i 的情况下，最长下降子序列有多少种方案。

(如果 LIS/LDS 必定不经过 i 这一点，那么 h_i/H_i 为 0)

h_i ，可以再计算 m_i 为以 i 开头的最长上升子序列的长度， n_i 为以 i 开头的最长下降子序列的长度。结合 m_i 、 n_i 、 f_i 和 g_i 可以将左右的方案数相乘，得到 h_i 。 H_i 的求法同理。

只需要判定 $\sum h_i \cdot H_i$ 是否与 $c_1 \cdot c_2$ 相等即可知道答案是否需要减一。

I. 字符串复制

Description

给定一个字符串 S ，求 S 复制 m 次拼接起来的的字符串的本质不同子串个数。

数据范围： $1 \leq |S| \leq 3 \times 10^5, 1 \leq m \leq 10^9$

Solution

令函数 $f(S, m)$ 表示 S 复制 m 次的本质不同子串个数。

当 $m \leq 2$ 时，暴力即可。

注意到当 $m \geq 3$ 时，为 SSS 的子串而不为 SS 的子串，一定可以写成 $x[S]y$ 的形式 (x 为 S 的非空后缀， y 为 S 的非空前缀)，为 $SSSS$ 的子串而不为 SSS 的子串，同理可写成 $x[SS]y$ 的形式，发现 $x[S]y$ 和 $x[SS]y$ 的个数是相等的。

补充：如果 aSb 是 SS 的子串，那么 $aSSb$ 一定可以写成 $x[S]y$ ，并且满足在 SSS 中出现而不在 SS 中出现。

因此，答案为 $[f(S, 3) - f(S, 2)] \times (m - 2) + f(S, 2)$ 。

计算本质不同子串需要使用后缀数组或后缀自动机。

时间复杂度 $\mathcal{O}(|S|)$ 。

J. 最大公因数的平方和

Description

给定长度为 N 的排列 a, b , 多次询问 $\sum_{l=i}^r \sum_{j=L}^R \gcd^2(a_i, b_j)$ 。

数据范围: $1 \leq N, Q \leq 10^5$

Solution

考虑 $\gcd^2$ 要怎么处理，做法和欧拉反演类似，可以构造一个 $\sum_{d|n} f(d) = n^2$ ，直接通过这个性质来倒推出来 f 函数。 f 函数可以预处理得到。这样问题就变成了求 $\sum_{d=1}^N \sum_{i=1}^r \sum_{j=L}^R f(d)[d|a_i][d|b_j]$ 。

这个就可以通过枚举因子来计算贡献，但是如果暴力枚举所有的 d 显然无法通过。

考虑离线，把所有的询问通过差分拆分成若干个前缀询问相加减的形式。

注意到 a, b 是一个排列，每个数在 a, b 中分别只出现一次，考虑根号分治。

Solution

$$\text{求 } \sum_{d=1}^N \sum_{i=1}^r \sum_{j=1}^R f(d)[d|a_i][d|b_j].$$

对于 $d < B$ 的因数，我们可以直接暴力枚举因子用前缀和计算。

对于 $d > B$ 的因数，可以通过暴力枚举 d 的倍数 d_A 和 d_B ，他们对 $pa_{d_A} \leq r$ 和 $pb_{d_B} \leq R$ 的询问会有 $f(d)$ 的贡献。其中 pa_i 和 pb_i 分别表示 i 在 a 和 b 中的位置。可以把 d_A 和 d_B 看作二维平面上的一个点，那么就转化成了一个经典的二维数点问题。

$$\text{点数数量的级别是 } \sum_{d=B}^N \left(\frac{N}{d}\right)^2 = \mathcal{O}\left(\frac{N^2}{B}\right).$$

可以通过根号平衡使得单次修改的时间复杂度为 $\mathcal{O}(1)$ ，单次询问复杂度为 $\mathcal{O}(\sqrt{N})$ 。最终时间复杂度 $\mathcal{O}((N+Q)B + \frac{N^2}{B} + Q\sqrt{N})$ 。

Thank you!