

2025 年第二届西华师范大学程序设计大赛

题目详解

西华师范大学

A-O 题目题解

2025 年 5 月 24 日



题目难度分布

- **Sign:** A、B (<800)
- **Easy:** N、L、D、M (800-1000)
- **Mid:** C、J、K、F (1000-1200)
- **Hard:** O、E、I、G (1400-2000)
- **AK:** H (2200)

A: 题目信息

- 难度: 0 (Sign)
- 标签: 签到

题目背景

欢迎来到西华师范大学计算机学院主办的第二届程序设计竞赛！

你的第一个任务非常简单：编写一个程序，输出包含特定欢迎信息的字符串，以此开启你的竞赛征程。

A: Solution

Solution

直接用你所擅长的编程语言输出即可！

时间复杂度

$O(1)$

B: 题目信息

- 难度: 400 (Sign)
- 标签: 签到, 模拟

题目背景

虾仁在程序设计竞赛中企图作弊，但监考系统设置了严格的检测规则。每当发生作弊行为时，系统会根据以下规则判断是否记录违规：

1. 传纸条：若行为时间能被 3 整除且不是 5 的倍数。
2. 偷看屏幕：若行为时间在 $[10, 50]$ 或 $[100, 150]$ 区间内。
3. 使用小抄：若行为时间精确等于 7 的倍数 (7, 14, 21, ..., 294) 且不超过 300。

请编写程序计算虾仁在竞赛中累计被抓次数。

N: Solution

Solution

本题直接根据题意进行模拟即可，考察分支循环等基本语法。

时间复杂度

$O(n)$ ，其中 n 是虾仁的作弊次数。

N: 题目信息

- 难度: 800 (Easy)
- 标签: 模拟, 字符串, 双指针

题目背景

虾仁学习了很多字符串算法, 比如马拉车、Z 函数、字符串哈希、KMP、字典树、AC 自动机、后缀自动机, 那么接下来这个 AK 自动机你会吗? 给定一个字符串 S , 请找出 S 的一个前缀和后缀 (前后缀长度必须相等, 可以为空), 使得它们拼接后是一个回文串, 请输出拼成的回文串最长长度。回文串: 正着读和反着读一样的字符串。

N: Solution

Solution

简单题意就是找到长度相同的前缀和后缀，要使得构成的字符串为回文串。直接双指针模拟， $l = 0, r = |S| - 1$ 。

- 如果 $S_l = S_r$ ，则左指针向右移动 ($l++$)，右指针向左移动 ($r--$)。
- 指针每移动一步，答案计数器加 1。

直接遍历完所有字符串，最后答案为当前计数器值的 2 倍。

时间复杂度

$O(n)$ ，其中 n 是字符串 S 的长度。

L: 题目信息

- 难度: 800 (Easy)
- 标签: 模拟, 打表, 字符串, 数学

题目背景

虾仁刚刚遇到了一个非常简单的题, 请输出第一个数乘第二个数的值。当然这个数是有规律的, 第一个数是由 n 个 3 组成, 第二个数是由 $m - 1$ 个 3 然后拼接一个 5 构成。答案可能非常大, 请使用高效的输出方式!

L: Solution

Solution

通过观察, 所得的答案只包含 0 1 5 6。进一步打表观察, 当 $n = m$ 时, 答案只有 1 5, 并且 1 的数量为 n , 5 的数量为 m 。所以本题直接分类讨论:

- $n = m$: 答案为 n 个 1, m 个 5, 直接拼起来。
- $n < m$: 答案为 n 个 1, $m - n$ 个 0, n 个 5, 直接拼起来。
- $n > m$: 答案为 m 个 1, $n - m$ 个 6, m 个 5, 直接拼起来。

小 Tips: 直接通过字符串构造函数生成, 一个一个字符的输出可能会超时, 即使用 `string a(n, '1')` 生成一个包含 n 个字符 1 的字符串 a 。

D: 题目信息

- 难度: 800 (Easy)
- 标签: 构造, 贪心, 数学

题目背景

虾仁非常怀念现代的游戏时间, 奈何在古代可没有游戏机玩, 心血来潮的他打算和熟人 yhgg 玩一局好玩的游戏。这个游戏一共有 n 轮 ($1 \leq n \leq 10^5$), 在第 i 轮获胜会获得 i 分, 没有平局。现在给出虾仁和 yhgg 的得分, 求是否有一种方案符合得分。

D: Hints

Hint

两人进行 n 局游戏后的总得分一定是 $\frac{n(n+1)}{2}$ 。如果不是，则说明得分不合法。

Hint

合法后，题目让我们从 $1 \sim n$ 中挑出几个数加起来为 a 。

时间复杂度

$O(n)$ 。主要是贪心构造部分。

D: Solution

Solution

首先通过总得分 $a + b$ 计算游戏局数 n 。计算 n 可以通过解二次方程 $N(N + 1)/2 = a + b$ 得到 $N = \frac{\sqrt{8(a+b)+1}-1}{2}$ 。若 N 不为正整数，则无解。有解的时候需要构造虾仁获胜的局数。采用贪心的思路：

- 从 n 开始反向枚举到 1。
- 如果当前的 a 大于等于 i ，则令 $a := a - i$ ，并将 i 加入解集。
- 直到 a 变为 0 退出循环。

可以证明在游戏局数 n 合法的情况下，这种贪心策略总是能够找到可行的构造。最后输出 n 和找到的解集。

M: 题目信息

- 难度: 1000 (Easy)
- 标签: 枚举, 前缀和, 计数

题目背景

虾仁在当锦衣卫的时候破获了许多离奇案件，当下有个非常棘手的密室。他进入密室后发现了许多暗器朝他飞来，他猛的回头看去只见门已经关上了，上面有一个谜题需要解答了才能开门，只见上面写着：“现有 01 串于此，则于所有子串中 0 之数同于 1 之数为几何？”简单题意：给定一个长度为 n 的字符串 S ，并且 $\forall S_i \in \{0, 1\}$ ， $i = 1, 2, \dots, n$ ，请问所有子串中 0 的个数和 1 的个数相同的子串的数量是多少？

M: Hints

Hint

子串中 0 的数量等于 1 的数量，可以转换为：将所有的 0 换成 -1，将所有的 1 换成 1。所求即为元素和为 0 的子串个数。

Hint

可以使用前缀和 S_i 快速计算区间 $[l, r]$ 的和 $(S_r - S_{l-1})$ 。一个子串 $[l, r]$ 的和为 0 当且仅当 $S_r = S_{l-1}$ 。

时间复杂度

 $O(n \log n)$

M: Solution

Solution

将字符串中的 '0' 替换为 -1, '1' 替换为 1。计算前缀和数组 S , 其中 S_i 表示前 i 个元素的和 ($S_0 = 0$)。问题转化为找到多少对 (l, r) 使得区间 $[l, r]$ 的和为 0, 即 $S_r - S_{l-1} = 0 \implies S_r = S_{l-1}$ ($1 \leq l \leq r \leq n$)。使用一个 map 或哈希表记录每个前缀和值出现的次数。

- 初始化 map, 'count[0] = 1' (对应空前缀)。
- 遍历 i 从 1 到 n :
 - 计算当前前缀和 S_i 。
 - 答案加上 $\text{count}[S_i]$ (这表示有多少个 $l-1$ 使得 $S_{l-1} = S_i$)。
 - 更新 $\text{count}[S_i]$, 即 $\text{count}[S_i] := \text{count}[S_i] + 1$ 。

最终的答案即为累加结果。

C: 题目信息

- 难度: 1000 (Mid)
- 标签: 枚举, 模拟, 数学

题目背景

沉浸在黑暗的牢笼里吧! 虾仁在一个 $n \times m$ 的二维网格中, 放置了几个黑点。每隔一秒钟, 所有与至少两个黑点相邻 (即上下左右四个方向) 的网格都会生成一个黑点。要使整个网格都存在黑点, 请问虾仁一开始至少需要放置多少个黑点?

C: Solution

Solution

首先, 观察到最优策略之一是在对角线上放置黑点。对于一个 $n \times n$ 的正方形, 放置 n 个对角线上的点即可最终覆盖整个正方形。现在考虑 $n \times m$ 的长方形 (假设 $n \leq m$)。我

们可以先在左上角放置一个 $n \times n$ 的正方形的对角线, 即放置 $(1,1), (2,2), \dots, (n,n)$ 共 n 个黑点。例如, 3×5 的网格, 初始放置 B 的位置:

B	W	W	W	W
W	B	W	W	W
W	W	B	W	W

这些点经过一段时间后会填满左边的 $n \times n$ 区域。

C: Solution

Solution

当左侧 $n \times n$ 被填满后，考虑右侧 $n \times (m - n)$ 的区域。以 3×5 为例，状态变为：

B	B	B	W	W
B	B	B	W	W
B	B	B	W	W

接下来需要填充右侧的白色区域。观察发现，在剩余的 $n \times (m - n)$ 区域中，每两列只需要再放置一个黑点即可。例如，对于第 4 列和第 5 列，只需要在 $(1, 5), (2, 5), (3, 5)$ 中任意一个位置放置一个黑点，这两列最终都会被填满。总共需要处理 $m - n$ 列。每两列需要一个额外的黑点。

C: Solution

Solution

因此，需要的额外黑点数为 $\lceil \frac{m-n}{2} \rceil$ 。注意，这个公式可以写成 $\lfloor \frac{m-n+1}{2} \rfloor$ 或直接使用整数除法 $(m - n + 1) / 2$ 。

C: Solution

Solution

因此，总共需要的黑点数量为：

- 初始 $n \times n$ 对角线需要的 n 个点。
- 填充剩余 $n \times (m - n)$ 区域需要的 $\lceil \frac{m-n}{2} \rceil$ 个点。

总数为：

$$n + \left\lceil \frac{m - n}{2} \right\rceil$$

假设 $n \leq m$ 。如果 $n > m$ ，交换 n, m 即可。

时间复杂度

$O(1)$ 。

J: 题目信息

- 难度: 1200 (Mid)

- 标签: RMQ, 数学, 数据结构

题目背景

给定一个长度为 n 的数列, 数列的每一项为 a_i 。你需要回答以下 q 个问题, 每个问题会询问一对 l, r , 你可以从 $a_l, a_{l+1}, \dots, a_r$ 中选择一些数, 不能不选。假设你选了 m 个数 $a_{p_1}, a_{p_2}, \dots, a_{p_m}$, 你需要保证 $\gcd(a_{p_1}, a_{p_2}, \dots, a_{p_m})$ 是这个区间里对于任意一种选数情况的最大值。请回答这个最大的 $\gcd(a_{p_1}, a_{p_2}, \dots, a_{p_m})$ 是多少。

J: Hint

Hint

$\gcd(a_1, a_2, \dots, a_m)$ 随着 m 的增加, 其值是单调不增的。也就是说, 选择的数越多, GCD 不会变大。

时间复杂度

对于 q 次查询, 选择 ST 表更优。使用 ST 表: 预处理 $O(n \log n)$, 查询 $O(q)$ 。总时间复杂度 $O(n \log n + q)$ 。使用线段树: 预处理 $O(n)$, 查询 $O(q \log n)$ 。总时间复杂度 $O(n + q \log n)$ 。

根据 Hint 1, 要使得 GCD 最大, 我们应该尽可能少地选择数。只选择一个数 a_i 时, GCD 就是 a_i 本身。因此, 在

区间 $[l, r]$ 中，能得到的最大 GCD 就是该区间内的最大值 $\max(a_l, a_{l+1}, \dots, a_r)$ 。因为我们可以只选择这个最大值所在的那个数，此时 GCD 就是它本身，且不可能有比它更大的 GCD。问题转化为求解区间最大值（Range Maximum Query, RMQ）。可以使用以下数据结构：

- **ST 表 (Sparse Table):** 预处理 $O(n \log n)$, 查询 $O(1)$ 。
- **线段树 (Segment Tree):** 预处理 $O(n)$, 查询 $O(\log n)$ 。

K: 题目信息

- 难度: 1200 (Mid)
- 标签: 前缀和, 数学, 排序, 中位数

题目背景

虾仁的好盆友思怡有很多书, 共摆了 n 堆, 每堆书籍有 a_i 个。虾仁可以通过增加或者降低某些堆的高度来保持他们高度一致。设降低的总高度为 d_1 , 增加的总高度为 d_2 , 现在思怡想要让这些书变得相同高, 同时 $d_1 + d_2$ 最小, 如果有多个同时满足 $d_1 + d_2$ 最小的高度 h , 那么选择高度 h 最小的一个。请问虾仁选择的目标高度是多少?

K: Solution I

Solution

解法一 (中位数定理) 中位数定理, 将一个数组变为同一个数所需要变化次数最小的方案, 直接将数组的数字变为数组的中位数即可。所以答案是数组的中位数, 但是需要输出最小的, 所以排序后输出下取整后下标位置的值。

Solution

解法二 (枚举与前缀和) 将数组排序后从小到大枚举, 我们只需要变化高度为数组中存在的数字, 不然不会更优。求个前缀和, 模拟每次变为 a_i 时 $d_1 + d_2$ 的值, 更新答案和当前最小值。

K: Solution II

时间复杂度

时间复杂度: $O(n \log n)$

F: 题目信息

- 难度: 1200 (Mid)
- 标签: 模拟, 枚举, 离散化, 计数

题目背景

本题为虾仁的珠宝差异简单版本，两题的唯一区别在于本题的数据范围更小。虾仁想知道有多少区间的珠宝差异度恰好等于 k 。定义区间的珠宝差异度 $f(l, r)$ 的值为区间 $[l, r]$ 内珠宝的个数 $(r - l + 1)$ 减去珠宝种类数。简单来说，给定一个长为 n 的序列 a ，请计算：

$$\sum_{1 \leq l \leq r \leq n} [f(l, r) = k]$$

F: Hint

Hint

注意到数据范围很小 ($n \leq 2000$), 可以使用 $O(n^2)$ 时间复杂度的算法通过。

时间复杂度

$O(n^2)$

F: Solution

Solution

目标是计算满足 $(r - l + 1) - \text{distinct_count}(l, r) = k$ 的区间 $[l, r]$ 的数量。双重循环枚举区间，对每个区间进行统计，需要一个计数数组，在每一次枚举新区间的时候清空，然后根据题意进行模拟即可，但是时间复杂度为 $O(n^3)$ ，需要优化。

我们可以让右端点枚举的时候不清空计数数组，因为需要知道前面的信息，所以计数数组只需要在左端点变动的时候清空。

O: 题目信息

- 难度: 1400 (Hard)
- 标签: 模拟, 树状数组, 二维偏序, 数论, 概率

题目背景

虾仁去参加开门大吉节目。节目有 n 首歌, 共 n 关。第 i 关的歌曲是 A_i 。虾仁的脑中按顺序存储了 n 首歌的信息, 顺序为 B 。脑容量空间从 1 到 n 编号, 编号越小记忆越清晰。在第 i 关开始时, 假设当前脑中还剩 m 首歌, 且第 i 关的歌曲 A_i 在脑中存储位置为 p (从 1 开始计数)。虾仁能想起这首歌的概率是 $\frac{m-p+1}{m}$ 。通过一关后, 脑中关于这首歌的记忆被删除, 后面的歌曲信息依次向前覆盖。求虾仁能通过所有 n 关的概率, 结果对 $10^9 + 7$ 取模。

O: Hints

Hint

最终的概率是每一关成功概率的乘积。分母是 $\prod_{i=1}^n (n - i + 1) = n!$ 。我们需要计算概率的分子，然后乘以 $n!$ 的逆元。

Hint

需要快速知道在第 i 关时，歌曲 A_i 在虾仁当前脑中的位置 p 。

时间复杂度

最后将分子乘上分母的逆元就是最后答案。总时间复杂度： $O(n \log n)$ 。

O: Solution

Solution

我们需要模拟这个过程，并计算每一关成功的概率，然后将它们相乘。关键在于快速找到第 i 关歌曲 A_i 在当前脑中（未被删除的歌曲序列）的位置 p 。

- 当前枚举到第 i 首节目组的歌曲 A_i ，虾仁脑中这首歌曲所在的位置为 id_{A_i} ，还剩下的歌曲数为 $m = n - i + 1$
- 那虾仁脑中在这首歌之前已经想起的歌曲个数是多少呢？
- 使用树状数组可以快速求前缀和，并且枚举完成后在树状数组上这个位置加 1，那么分子就可以写成 ' $m - (id[A[i]] - fen.query(id[A[i]])) + 1$ '

E: 题目信息

- 难度: 1600 (Hard)
- 标签: 树形 DP, 背包

题目背景

柳树迎着春风飘荡, 将柳树抽象为一颗树, 根节点为 1, 大小为 n , 有 $n-1$ 条边。柳色 k 定义为删去最少的边后与根节点连通的节点数 (包含根节点) 为 k 。请对每个 $k = 1, 2, \dots, n$ 输出柳色 k 的值 (即最少需要删除多少条边)。

E: Hints

Hint

删去一条边 (u, v) (假设 u 是 v 的父节点), 相当于将以 v 为根的子树从原树中分离出去。代价为 1。

Hint

如果你要得到柳色 k 不妨根据 Hint 1 凑出 $n - k$ 。

E: Solution

Solution

树上背包，注意要用上下界优化能把时间降到 $O(n^2)$ 。

时间复杂度

$O(n^2)$ (使用树上背包和上下界优化)。

I: 题目信息

- 难度: 1600 (Hard)
- 标签: 模拟, 二分, STL, 数据结构

题目背景

虾仁被追捕。有 $n + 1$ 个地区, 编号 $0 \dots n$ 。洪武大帝在 0 号地区。监视器显示虾仁所在地区编号 a_i 。共有 m 次显示。

- 操作 1: 若 $a_i \in [1, n]$, 大帝派人去 a_i 区。虾仁立即将 a_i 区与其左侧区域交换位置 (若左侧是 0 区则不交换)。此次行动消耗金钱为 a_i 区当前实际位置到 0 区的距离。
- 操作 2: 若 $a_i = 0$, 大帝不动。虾仁将 $1 \dots n$ 区域反转。不消耗金钱。

求 m 次操作总共消耗的金钱。数据范围: $1 \leq n \leq 10^9$, $1 \leq m \leq 10^5$



I: Hints

Hint

n 很大 (10^9), 不能直接用数组模拟 $0 \dots n$ 的排列。只能记录那些被操作影响了区域。

Hint

交换 $1 \sim n$ 区域, 实际上只在逻辑上交换, 使用一个 *bool* 变量控制当前是什么状态即可, 没交换计算的是到 0 的距离, 交换后计算的是到 $n+1$ 的距离。

I: Hints

Hint

需要记录每个被影响的区域当前实际在哪个位置，或者记录每个位置当前是哪个区域。由于 n 很大，只能记录被操作过的区域。考虑维护一个数据结构（如平衡树或 map）来存储 $(position, region_id)$ 的映射，只存储那些位置与 $region_id$ 不一致的区域。或者维护 $(region_id, position)$ 的映射。还需要处理交换和反转。反转操作对所有 $1 \dots n$ 的区域都有影响，难以高效处理。

时间复杂度

时间复杂度： $O(m \log m)$ (set/map 操作为 $\log m$ ，共 m 次操作) 或 $O(m \log n)$ (如果 set 中元素数量与 n 相关) 更准确地说是 $O(m \log(\text{number of affected regions}))$ ，最坏情况下是 $O(m \log m)$ 。

I: Solution

Solution

使用 map 记录当前区域 x 偏移了多少, 使用 set 记录当前区域的左侧区域交换后的位置, 然后分情况模拟。

- 当我们枚举到 i , 当前位置的区域为 x , 偏移了 $map[x]$
- 那么实际的位置应该是 $x - map[x]$
- 接着在 set 中二分查找实际位置左边的一个区域是啥
- 让它的偏移度增加, 然后更新 set 的值 (将原来的删掉, 新的位置加进去)。

G: 题目信息

- 难度: 2000 (Hard)
- 标签: 计数, 二分, 数据结构, 离散化, 莫队?

题目背景

本题为虾仁的珠宝差异困难版本, 两题的唯一区别在于本题的数据范围更大。给定一个长为 n 的序列 a , 请计算有多少区间 $[l, r]$ 满足珠宝差异度恰好等于 k 。

$$\sum_{1 \leq l \leq r \leq n} [f(l, r) = k]$$

其中 $f(l, r) = (r - l + 1) - (\text{number of distinct elements in } a_l, \dots, a_r)$ 。数据范围:
 $1 \leq n \leq 2 \times 10^5, 0 \leq k \leq n$ 。

G: Hints

Hint

$f(2, 2, 1) = 1$, 此时加入一个没出现过的珠宝 3 ,
 $f(2, 2, 1, 3) = 1$, 如果此时加入一个出现过的珠宝 2 ,
 $f(2, 2, 1, 3, 2) = 2$, 由此可得函数 f 单调不降, 考虑二分。

Hint

如何计算一个区间不同珠宝种类数量? 如果你先预处理一个 col 数组, col_i 表示 $[1, i]$ 不同种类的数量, 这很容易。那么你要询问任何一个区间 $[l, r]$ 只需要考虑到 $[1, l)$ 的贡献并减去他们即可。

G: Solution I

Solution

- 我们从左到右遍历，计算 $\sum_{i=1}^n \sum_{j=i+1}^n [f(i,j) = k]$
- 根据 Hint 1，我们不需要实际遍历 $[i+1, n]$ ，而是二分出满足条件的两个端点 l, r ，答案加上 $r - l + 1$ 即可
- 二分的 *check* 即区间长度减去不同种类数量与 k 比较。

G: Solution III

时间复杂度

总结需要区间减和单点求和操作，所以需要一些支持这些操作的数据结构，如线段树，树状数组等。时间复杂度： $O(n \log^2 n)$

H: 题目信息

- 难度: 2200 (AK)
- 标签: 数学, 推公式, 数论, 调和级数, 容斥原理, 莫比乌斯反演?

题目背景

这里有一串长度为 n 的序列 a 。请计算：

$$\sum_{i=1}^n \sum_{j=i+1}^n \sum_{k=j+1}^n f(a_i, a_j, a_k) \pmod{10^9 + 7}$$

函数 $f(x, y, z)$ 定义为：先将 x, y, z 升序排列为 $x' \leq y' \leq z'$ ，则 $f(x, y, z) = \text{lcm}(x', y')$ 。数据范围： $3 \leq n \leq 10^5$ ， $1 \leq a_i \leq 10^5$ 。

H: Hints

Hint

对序列 a 排序不影响最终答案, 因为 f 函数内部会进行排序。排序后 $a_i \leq a_j \leq a_k$ 对于 $i < j < k$ 成立。此时 $f(a_i, a_j, a_k) = \text{lcm}(a_i, a_j)$ 。

Hint

$$\text{lcm}(a, b) = \frac{a \times b}{\text{gcd}(a, b)}$$

Hint

$1 \leq a_i \leq 10^5$ 。考虑对 a_i 进行因子分解或利用值域相关的性质。

H: Solution

Solution

推导

$$\text{Ans} = \sum_{i=1}^n \sum_{j=i+1}^n \frac{a_i \times a_j}{\gcd(a_i, a_j)} \times (n - j)$$

枚举 $d = \gcd(a_i, a_j)$

$$\text{Ans} = \sum_{d=1}^{\max(a_i)} \frac{1}{d} \sum_{j=2}^n a_j (n - j) \sum_{i=1}^{j-1} [\gcd(a_i, a_j) = d] a_i$$

令 $h_d(j) = \sum_{i=1}^{j-1} [\gcd(a_i, a_j) = d] a_i$ 。

H: 时间复杂度

Solution

$$\text{Ans} = \sum_{d=1}^{\max(a_i)} \frac{1}{d} \sum_{j=2}^n a_j(n-j)h_d(j)$$

时间复杂度

$\sum_{\text{gcd}=d} a_i$ 由容斥完成：先记录所有 d 的倍数，再减去其倍数中的重复，具体做法是：我们从大到小枚举 a_i 的因子 d 先让 $h_d = s_d$ ，此时如果存在一个数 t 且 $t \mid d$ ，那么 h_t 应该减去 h_d ，所以还要用调和级数预处理出因子。时间复杂度： $O(V \log V + n \log V)$ 或 $O(V \log V + n\sqrt{V})$ (其中 V 是 a_i 的最大值)，取决于 $h_d(j)$ 如何计算。更精细的分析可能是 $O(V \log V + n)$ 。



Thanks!