摸底测试讲评

2024 年 7 月

feecle8146

前言

本次摸底测试的题目难度和 CSP-S 相比，大约为：

- T1：小于 T1
- T2：T1~T2
- T3：T3~T4
- T4：T3

但是，本次测试的部分分多，难度小，仅凭暴力也能获得可观的分数。由于许多涉及的算法还没有系统地讲解过，今天的重点是暴力怎么写。

题目分析

T1

求和操作

给定一个数字串，每次可以把相邻两位替换为它们的和。

求使得数字串只剩下一位的最小操作次数。

无论采用什么操作顺序，操作次数都是一样的。

一种做法是，每加入一个字符，就用操作把当前字符串变成一个字符。

题目分析

T1

求和操作

给定一个数字串，每次可以把相邻两位替换为它们的和。

求使得数字串只剩下一位的最小操作次数。

为什么？

最后剩下的数字一定是原串 mod 9。

每次操作要么使字符串长度减少 1（不改变原串数字之和），要么使原串数字之和减少 9（不改变串长）。所以，两部分的减少互不干扰，最终操作次数自然一样。

敢想敢写

如果想到一个容易写成代码的做法，一时也无法找到反例，不妨先写出代码测测大样例。

（注意：如果实现难度较大，要三思；同时，写一个不能确保正确性的（由直觉得来的）做法之前，先花点时间找反例）

在现在的比赛中，大样例一般不会刻意造弱，一定要把它利用好。

题目分析

T2

数论结构

q 次询问正整数 c,d,x，问有多少对 $(a,b)\in \mathrm{Z}_+$ 满足 $c\times \operatorname{lcm}(a,b)-d\times \gcd(a,b)=x$。

$$q\le 10^4, c,d,x\le 10^7$$

40% 的数据满足 $q=1,c,d,x\le 30$。

大胆猜测 a,b 不会太大。在时间允许的范围内，枚举尽量多的 a,b 计算答案。例如，本题中枚举到 $a,b\le 5000$ 肯定足够了。

在限制范围内尽量多枚举

根据时限以及对时间复杂度的预估，尽量枚举足够多的方案，是一种常见的骗分手段。

有时本方法还有更高级的运用，例如 dp 转移时，只执行按照某种贪心方式选出的 1000 个转移。

题目分析

T2

数论结构

q 次询问正整数 c, d, x，问有多少对 $(a, b) \in \mathrm{Z}_+$ 满足 $c \times \mathrm{lcm}(a, b) - d \times \gcd(a, b) = x$。

$$q \le 10^4, c, d, x \le 10^7$$

首先需要化简 lcm：$lcm(a, b) = a \times b / \gcd(a, b)$。

注意：$lcm(a, b)$ 也是 $\gcd(a, b)$ 的倍数，故上式意味着 $\gcd(a, b) | x$ 。

可以枚举 x 的因数 $g = \gcd(a, b)$，问题变为有几对 $\frac{abc}{g} - dg = x$。

题目分析

T2

数论结构

q 次询问正整数 c,d,x，问有多少对 $(a,b)\in \mathrm{Z}_+$ 满足 $c\times \operatorname{lcm}(a,b)-d\times \gcd(a,b)=x$。

$$q\le 10^4, c,d,x\le 10^7$$

枚举 x 的因数 $g=\gcd(a,b)$，问题变为有几对 $\frac{abc}{g}-dg=x$。

令 $u=\frac{a}{g}, v=\frac{b}{g}$，$(u,v)=1$，只需算出有几对 $uvgc-dg=x$，也即 $uv=\frac{\left(\frac{x}{g}+d\right)}{c}$。

题目分析

T2

数论结构

q 次询问正整数 c, d, x，问有多少对 $(a, b) \in \mathrm{Z}_{+}$ 满足 $c \times \mathrm{lcm}(a, b) - d \times \gcd(a, b) = x$。

$$q \leq 10^4, c, d, x \leq 10^7$$

设右侧有 s 个质因子，则 (u, v) 就有 2^s 种选法。

因数、质因子个数可以用线性筛预处理，时间复杂度 $O(N + qd(N))$，$N = 10^7$。

题目分析

小结

本题中，有几个值得注意的思维方式：

1. 枚举 gcd，并除掉 gcd 化为互质。
2. 分离变量，将方程化为简单的形式。
3. 考虑每个质因子。

之后的课程中，我们将总结更多类似的思维方式。

题目分析

T3

交换次数

有一个 $1\sim n$ 的排列。每次操作可以交换相邻两个数。

你希望用最小的交换次数，使得排列中存在一个子区间，恰好为 $[1,2,\dots,m]$。记答案为 $f(m)$。

q 次给出 m 求 $f(m)$。不同询问互相独立。

20% 的数据满足 $n\le 5, q=1$。

可以 dfs。dfs 的过程中，记忆化每个排列达到的最小次数。

若当前次数已经超过最小次数，则剪枝。

题目分析

交换次数

有一个 $1\sim n$ 的排列。每次操作可以交换相邻两个数。

你希望用最小的交换次数，使得排列中存在一个子区间，恰好为 $[1,2,\ldots,m]$。记答案为 $f(m)$。

q 次给出 m 求 $f(m)$。不同询问互相独立。

40% 的数据满足 $n\leq 8$。

把排列看成点，题目中的操作看成边，这是边权为 1 的最短路问题！

可以使用 BFS 解决，用 map 存储到每个排列的最小操作次数。

BFS 与 DFS

很多时候（例如写暴力骗分/对拍/找规律），我们都需要一个绝对正确的程序。

对于状态变化类的题目，可以直接将状态看作点，变化看作边，在该图上执行 BFS/DFS/最短路。

可以直接用 map+vector（若状态不止一个数组，可以用结构体）存储这个图上每个点的信息。

题目分析

交换次数

有一个 $1 \sim n$ 的排列。每次操作可以交换相邻两个数。

你希望用最小的交换次数，使得排列中存在一个子区间，恰好为 $[1,2,\dots,m]$。记答案为 $f(m)$。

q 次给出 m 求 $f(m)$。不同询问互相独立。

部分数据满足 $m=n$。

此时，就是将排列排序，经典结论告诉我们，答案为逆序对数。

套用求逆序对数的算法即可。

阅读部分分

在正式比赛中，一定要记得阅读每一题的每一档部分分。

部分分，作为赛题的特殊情况，有时就是经典题目/原题。

对于每档部分分，都应花一点时间（例如至少 5 min）来思考其做法。

这样可以最大程度避免丢失能力范围内的分数。

当然，如果部分分算法较麻烦，分值又少，就需要权衡利弊。

题目分析

T3

交换次数

有一个 $1 \sim n$ 的排列。每次操作可以交换相邻两个数。

你希望用最小的交换次数，使得排列中存在一个子区间，恰好为 $[1,2,\dots,m]$。记答案为 $f(m)$。

q 次给出 m 求 $f(m)$。不同询问互相独立。

一般情况下，可以将过程分为独立的两步：

1. 将 $1 \sim m$ 聚在一起。
2. 聚在一起后，把这个连续段排序。

题目分析

T3

交换次数

有一个 $1 \sim n$ 的排列。每次操作可以交换相邻两个数。

你希望用最小的交换次数，使得排列中存在一个子区间，恰好为 $[1,2,\dots,m]$。记答案为 $f(m)$。

q 次给出 m 求 $f(m)$。不同询问互相独立。

第一步里，不会改变 $1,2,\dots m$ 的相对顺序。
假设 a_i 最终移动至 p，需要 $|i-p|$ 次交换。本题就是给定了 m 个数 $u_1,\dots,u_m$ 要找到 p，使得 $\sum|(p+i-1)-u_i|$ 最小。

只需求出 $u_i-(i-1)$ 的中位数。这就给原题找出了多项式算法。

题目分析

交换次数

有一个 $1 \sim n$ 的排列。每次操作可以交换相邻两个数。

你希望用最小的交换次数，使得排列中存在一个子区间，恰好为 $[1,2,\dots,m]$。记答案为 $f(m)$。

q 次给出 m 求 $f(m)$。不同询问互相独立。

对于多次询问的情况，可以按照 m 从小到大的顺序，依次求两部分的答案。

逆序对数，可以对已有 $\leq m$ 元素的位置数维护树状数组。
中位数，可以在上述树状数组上二分，或直接二分+树状数组。

题目分析

T3

小结

本题中，有几个值得注意的思维方式：

1. 寻找独立性。
2. 先设计多项式算法。
3. 套用经典模型（本题中是中位数）。
4. 设计求值顺序。

之后的课程中，我们将总结更多类似的思维方式。

题目分析

T4

树上路径

给定一棵树，求树上的一条路径，使得 路径长度 − 路径点权最大值 + 路径点权最小值 最小。

$$n \le 10^6$$

20% 的数据 $n \le 50$。

使用各种算法都可以求出每一条路径的长度、最大值、最小值，例如 Floyd。再枚举路径端点即可。

题目分析

树上路径

给定一棵树，求树上的一条路径，使得 路径长度 – 路径点权最大值 + 路径点权最小值 最小。

$$n \le 10^6$$

50% 的数据 n^2 较小。

从每个点出发 dfs，途中记录经过的所有点的点权最大值、最小值、长度，即可求出所有以它为起点路径的权值。

时间复杂度 $O(n^2)$。

优化不必要的循环

例如，求所有区间最大值的和。

不好的暴力：枚举每个区间，再枚举区间内的每个元素求最大值。

更不好的暴力：枚举每个区间，再用 ST 表/线段树/......求区间最大值。

好的暴力：枚举每个区间。固定 l，在枚举 r 时，维护变量 mx，表示当前最大值。每一步，令 $mx \to \max(mx, a_r)$，再累加 mx。

题目分析

T4

树上路径

给定一棵树，求树上的一条路径，使得 路径长度 - 路径点权最大值 + 路径点权最小值 最小。

$$n \le 10^6$$

注意到要求的是最小值，故可以放宽限制为：

求树上的一条路径 P，再从路径上选两个点 u, v，使得 $len(P) - a_u + a_v$ 最小。

进一步，为了最小化 len，总可以认为 P 的两端就分别是 u, v！

题目分析

T4

树上路径

给定一棵树，求树上的一条路径，使得 路径长度 – 路径点权最大值 + 路径点权最小值 最小。

$$n \le 10^6$$

求一条路径 $u \to v$，使得 $dis(u,v) + a_u - a_v$ 最小。

枚举 LCA，求子树内 $dep + a$、$dep - a$ 分别的最小值即可。时间复杂度 $O(n)$。

题目难度排序仅供参考

在摸底测试中，可能对于部分选手，T4 比 T3 其实更简单。

有时，即使在正式比赛中，组题人也难以控制题目难度顺序；就算控制了，对于不同选手，同一题难度也有差别。

在正式比赛中，应当保证每道你没有秒掉的题都有一定的（在 CSP-S 级别，一个可供参考的数值是 20~30min）思考时间，切忌一道题做一整场，不看剩下的题。

题目分析

T4

小结

本题中，有几个值得注意的思维方式：

1. 放宽限制。
2. 枚举 LCA。

之后的课程中，我们将总结更多类似的思维方式。