2026 深圳职业技术大学程序设计竞赛
题解

SZPU-ACM

2026-05-10

目录

预期难度：

Easy ： A,C,E,F,I,J,M

Midium ： B,G,L

Hard ： D,H,K

所有题目时空限制均在标准程序 3 倍及以上。

A. 简单的前后缀问题

题意

给定 S, T，判断 S 是否同时为 T 的前缀与后缀。
数据范围：$1 \le |S|, |T| \le 15$。

A. 简单的前后缀问题

直接模拟即可，也可以用各个语言自带的函数快速完成。

C++:`starts_with`,`ends_with`
Java:`startsWith`,`endsWith`
Python:`startswith`,`endswith`

F. 找索引

题意

给定一个长度为 n 的单调不递减序列，询问 q 次值为 x 的下标，不存在则输出 `-1`。
数据范围：$1 \le n \le 2 \times 10^5, 1 \le q \le 2 \times 10^5, 1 \le x \le 10^9$。

F. 找索引

做法有很多种，任意一种都可以通过：

1. 用 `map/TreeMap`(C++/Java)、`dict`(Python) 存储每个值最后一次出现的索引，然后查询时直接输出即可。
2. 用二分查找找出第一个大于等于 x 的值，然后对比即可。

只要能在 $O(\log n)$ 复杂度内求出答案即可。

如果你使用了 `std::unordered_map` 等以哈希存储的结构，可能会因为哈希冲突导致 TLE

C. 能见度

题意

给定一个 $n \times m$ 初始全为 0 的带权矩阵，进行 q 次操作。

1. 区间高度增加 k。
2. 查询 (x, y) 的能见度。即 (x, y) 仅通过四个方向的直线能碰到所有高度不大于 $h_{x,y}$ 的数量。

数据范围：$1 \leq n, m \leq 100, 1 \leq q \leq 100$。

C. 能见度

直接按照题意模拟。

注意一下边界情况即可顺利通过。

C. 能见度

Fun fact:

原本这题有 Hard Version ，但是极致优化的暴力把 std 干翻了。

E. 简单的几何问题

题意

给定一个 $n \times m$ 的黑白格子，求黑白格子形成的周长。
数据范围：$1 \le n \times m \le 1000$。

E. 简单的几何问题

对于一个黑色格子，其对应的周长为周围四个格子白色格子的数量。

所以按题意模拟即可。

M. 困难的图论问题

题意

求 $n \in [l, r]$ 中有多少个 n 满足顶点数量为 n 的完全图是二分图。
数据范围：$1 \leq l \leq r \leq 10^9$。

M. 困难的图论问题

结论

n 仅在 $1, 2$ 时成立，其余不成立。

简单证明：

1. 当 $n = 3$ 时的完全图，显然不是一个二分图。
2. 当 $n > 3$ 时，由于一定存在 $n = 3$ 的生成子图，该生成子图一定不是二分图。但是由于二分图的**所有**生成子图一定是二分图，所以该图的原图不是一个二分图。

I.?! 强强!?

题意

给定一个长度为 n 的正整数序列 a，保证所有数字无前导零。可以对每个数字进行任意次修改，每次修改为将某一位数字从 x 改到 y，代价为 $c = |x - y|$。
求 $\sum b_i + c$ 最大化结果，其中 b_i 为修改后的数字。
数据范围：$1 \le n \le 2 \times 10^5, 1 \le a_i \le 10^9$。

I.?! 强强!?

操作一个数字时，将该数字每一位数都增加一定不劣。

证明：

记从右到左第 i 位将 x 修改成 y 对最终结果的影响为 $(y-x)\times 10^{i-1}-|y-x|$。当 $y>x$ 时，整个结果 ≥ 0。

所以我们只需贪心将每一位修改成 9 一定不劣。

时间复杂度为 $O(n\lg V), V=10^9$。

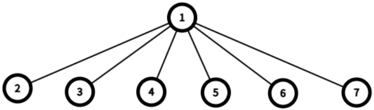
J.DFS 序对 (Easy Version)

题意

给定两个长度为 n 的 dfs 序 p, q，你需要构造出一颗存在这两个不同的 dfs 序的树。
数据范围：$1 \leq n \leq 2 \times 10^5, p_1 = q_1 = 1$。

J.DFS 序对 (Easy Version)

构造以 1 为根的菊花图即可。



该图可以满足所有长度为 n 且 $p_1 = q_1 = 1$ 的 dfs 序。

L. 简单的区间问题

题意

定义一个区间 $[l, r]$ 的权值为
$f(a[l \dots r]) = \max(a[l \dots r]) + \min(a[l \dots r])$。
求每个区间权值的总和。
数据范围：$1 \le n \le 2 \times 10^5, 1 \le a_i \le 10^8$。

L. 简单的区间问题

由于 min 和 max 满足结合律，所以对于区间 $[l, r]$ 的下标 i，若该值为最大值，那么所有 $[l, r]$ 范围内包含下标 i 的子区间中，下标 i 也一定是最大值。最小值同理。

那么可以作为下标 i，求出该下标分别作为最大值时的极大区间 $[l, r]$。使得该区间内最大值是 a_i 且区间 $[l-1, r]$ 和 $[l, r+1]$ 的最大值不是 a_i。
对于该极大区间 $[l, r]$，一共有 $(i-l+1) \times (r-i+1)$ 个不同的子区间满足 a_i 是最大值。

可以用单调栈快速求出每个下标 i 极大区间，从而得出答案。
最小值的处理方式同理。

需要注意一下同值情况下对最大最小值的影响。

时间复杂度为 $O(n)$。

G. 富有挑战的图上问题

题意

给定一个由 n 个节点组成的有向无环图 (DAG)，求该图的不同的哈密顿路径数量，并对 10^9+7 取模。
数据范围：$2 \leq n \leq 2 \times 10^5, n-1 \leq m \leq 4 \times 10^5$，保证图为 DAG。

G. 富有挑战的图上问题

DAG 性质

设该图拓扑序为 $v_1, v_2, \ldots, v_n$，则 v_i 一定无法到达 $v_1, v_2, \ldots, v_{i-1}$。

由哈密顿路径定义可得：在 DAG 上的哈密顿路径 $p_1, p_2, p_3, \ldots, p_n$，其一定恰好为该 DAG 的拓扑序。

G. 富有挑战的图上问题

若该图存在多个不同的拓扑序，则一定不存在哈密顿路径。

证明

若图 G 存在两个同时**不同**拓扑序 a, b，设顶点 x 在 a 中的下标为 i_x，在 b 中的下标为 j_x。
那么一定存在两个顶点 u, v，满足 $i_u < i_v$ 且 $j_u > j_v$。
由于 DAG 性质，对于拓扑序 a 中，v 无法到达 u；对于拓扑 b 中，u 无法到达 v。
与 DAG 上哈密顿路径的性质有悖，故不存在哈密顿路径。

G. 富有挑战的图上问题

由此，我们仅需进行拓扑判断是否存在多个拓扑序。

然后通过重边数量求方案数即可。

时间复杂度为 $O(n)$ 到 $O(n\log n)$。

B. 困难的树上 MEX 问题

题意

给定一棵大小为 n、根节点为 1 的有根树，每个节点有对应点权 w_i，且所有点的点权恰好形成一个 $0 \sim n-1$ 的排列。

对于每个节点 u，求出 $V(u) = \max_{v \in T_u}\Big(\mathtt{MEX}\big(\mathtt{path}(u,v)\big)\Big)$

数据范围：$1 \le n \le 2 \times 10^5, 0 \le w_i \le n-1$。

B. 困难的树上 MEX 问题

符号定义：
设节点 u 子树为 T_u,v 为 u 的直接儿子，权值为 w_u，该子树中所有**后代节点**的最小权值为 w'_u，节点 u 的答案为 $V(u)$。

由 `MEX` 定义可得，对于节点 u 而言，选择包含值为 w'_u 的路径作为最终答案的路径一定不劣。而由于所有节点的权值是排列，所以最优路径是唯一的。即最终非零答案的节点一定是一条以 1 号节点到达权值为 0 节点的链。

于是我们可以跑一次树形 dp 求出每个节点 u 下一步进入的子树 v，最后从链底部求出每个点的 $V(u)$ 值即可。

时间复杂度 $O(n)$。

D. 简单的填充问题

题意

给定一个 $n \times m$ 的网格，其中去除 k 个格子。
求在该网格内填充 2×2 正方形，且任意两个正方形公用边长度不超过 1。
求方案数，并对 998244353 取模。
数据范围：$2 \le n, m \le 12, 0 \le k \le n \times m$。

D. 简单的填充问题

考虑状压 dp。将每一个格子状压，然后暴力枚举上三层状态。这样的复杂度为 $O(n2^{3m})$，显然无法通过本题。

D. 简单的填充问题

做法 1

由于每一行有效状态数不会很多，可以考虑预处理求出只有不超过 100 个有效状态。
直接暴力枚举最后 3 层的状态，然后判断是否合法即可。
需要判断的 Case 比较多，精细实现即可通过。
时间复杂度为 $O(nB^3)$，其中 $B < 100$。

D. 简单的填充问题

做法 2

考虑三进制优化枚举，将每一个格子记录为当前格子距离最近正方形左上角的距离。
这样就可以只要确定上一维的状态即可。
然后用轮廓线 dp 优化逐格枚举优化枚举两行状态。
时间复杂度为 $O(nm^2 3^m)$，需要注意常数才能通过。
预处理每个状态在第 i 列下一步操作转换后的状态，可以优化成 $O(nm3^m)$，这样能轻松通过。

K.DFS 序对 (Hard Version)

题意

给定两个长度为 n 的 dfs 序 p, q，你需要构造出一颗存在这两个不同的 dfs 序的树，且深度总和最大化。
数据范围：$1 \le n \le 5000, p_1 = q_1 = 1$。

K.DFS 序对 (Hard Version)

当我们确定了以 u 为根、大小为 m 的子树时，该子树的 m 个节点深度均增加了 1。

所以我们考虑尽可能地创建子树，且这些子树尽可能地大，这样才能使得所有节点深度最大化。

K.DFS 序对 (Hard Version)

考虑什么时候可以新建一棵子树?

对于一棵 u 为根的子树，其 dfs 序内节点组成的集合一定相同，且该 dfs 序内第一个节点一定是 u。对于这样的一棵树，点 v 属于点 u 的后代，则点 v 的子树节点集合是点 u 的子树节点集合的子集。

由于所有子树呈包含关系且子树内部所有合法构造均不会对子树外的节点深度影响，所以当每个节点深度最大化时，深度总和才最大化，即贪心使得所有合法子树区间都构建。

K.DFS 序对 (Hard Version)

对于每个节点 u，找出最大的合法子树，然后递归建子树内剩余节点即可。

建树的方案有很多，可以通过预处理或者哈希快速判断集合是否一样，然后依次建树。

时间复杂度为 $O(n^2)$。

H. 简单的 RBS 问题

题意

给定一个长度为 n 的合法括号串 (RBS) S，可以对该串执行任意次（包括零次）操作：

- 选择一个子串为 RBS 的区间 $[l, r]$，翻转位置并翻转左右括号。

求最终可以形成多少种不同的 RBS。

数据范围：$2 \le n \le 10^5, n \in \texttt{even}, S$ 为 RBS。

H. 简单的 RBS 问题

对于单次操作，其本质可以看做是同深度的相邻括号串交换位置。设需要操作的串为 $R_1 = (r_1), R_2 = (r_2)$，设 S 翻转后的串为 S'，那么可以通过操作实现交换位置：

$R_1 R_2 \to R'_2 R'_1 = (r'_2)(r'_1) \to (r_2)(r_1) = R_2 R_1$。

那么，我们可以把操作看作是对同层的 RBS 重新排列。

H. 简单的 RBS 问题

由于可以通过操作更换前后顺序，于是只有括号深度对整体操作有约束，因此考虑建树。
可以将该括号串看作 dfs 过程进行建树，左括号看作想下新建儿子并进入，右括号看作回溯。

通过建树后，每次操作可以看作是对 u 的两个儿子 v_1, v_2 为根的子树进行位置交换。

H. 简单的 RBS 问题

本质不同的树：当且仅当两颗树无法通过操作达成相同的形状。

对于每棵子树 u，设其出度为 deg_u，本质不同的子树有 k 种，每种子树有 m_i 棵，每棵这种子树通过操作可以得到 x_i 种不同形态的树。

通过简单的排列组合可以得出：
以 u 为根的子树方案数为 $\frac{deg_u!}{\prod_{i=1}^{k} m_i!} \prod_{i=1}^{k} x_i^{m_i}$。

本质不同的树可以通过树哈希容易求出，由于树的形态比较多，对哈希函数的设计相对严苛。

最终时间复杂度为 $O(n\log n)$。

Thank you!