2024 中国大学生程序设计竞赛 全国邀请赛 (山东)

SUA 程序设计竞赛命题组

2024 年 5 月 26 日

概况

- 阶段一（Easy）：I、K、A、F
- 阶段二（Medium）：C、J、D
- 阶段三（Difficult）：H、M、E
- 阶段四（Challenging）：L、B、G

I. 左移

题意

- 给定字符串，问至少左移几次之后，字符串的首尾字符相同。
- 字符串长度 5×10^5。

- 若字符串没有相同的相邻字符则无解。
- 否则枚举左移次数 $0 \le d < n$，判断是否 $s_d = s_{(d+n-1) \bmod n}$ 即可。
- 复杂度 $\mathcal{O}(n)$。

K. 矩阵

题意

- 构造一个 $n \times n$ 的矩阵，元素范围从 1 到 $2n$。要求 1 到 $2n$ 每种元素至少出现一次，且恰有一个子矩阵的四角元素互不相同。
- $n \leq 50$。

K. 矩阵

- 考虑以下方法

$$\begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 & 4 \\ 5 & 6 & 7 & 8 & 9 & 10 \\ 5 & 6 & 7 & 8 & 11 & 12 \end{matrix}$$

- 前 $(n-2)$ 行，第 i 行全填 i，这样子矩阵的上下边界就只能选择最后两行，否则至少两个角会相同。
- 最后两行的前 $(n-2)$ 列，第 i 列全填 $(n-2)+i$，这样子矩阵的左右边界就只能选择最后两列，否则至少两个角会相同。
- 右下角的 2×2 子矩阵填剩下四个不同的数即可。
- 复杂度 $\mathcal{O}(n^2)$。

A. 打印机

题意

- 有 n 台打印机，第 i 台打印机每 t_i 秒打印一份试题，但每次打印 l_i 份试题后要休息 w_i 秒。
- 如果所有打印机同时工作，打印 k 份题至少要多久。
- $n \le 100, t_i, l_i, w_i, k \le 10^9$。

- 二分答案 x，计算每台打印机在 x 秒内能打印多少试题，看加起来是否大于等于 k。
- 最差情况下，$k = t_i = w_i = 10^9$，$l_i = 1$，所以二分上界是 2×10^{18}。
- 然而 $x = 2 \times 10^{18}$ 时，如果打印机每 1 秒打印一份题，$n = 100$ 台打印机一共可以打印 2×10^{20} 份，超出了 long long 的上界。所以二分过程需要判断和已经大于 k 时提前返回 true。
- 复杂度 $\mathcal{O}(n \log X)$。

F. 分割序列

题意

- 给定长度为 n 的序列，把它分成 k 段。设从左到右第 i 段的和是 s_i，最大化 $\sum_{i=1}^{k} i \times s_i$。对所有 $1 \le k \le n$ 求答案。
- $n \le 5 \times 10^5$。

- 设最后 m 段的元素总和为 p_m（其中 $p_k = \sum a_i$），则目标式可以改写为 $\sum_{i=1}^{k} p_i$。
- 所以答案就是从 $(n-1)$ 个后缀和里，选最大的 $(k-1)$ 个加起来，再加上序列的和。预处理后缀和并排序即可。复杂度 $\mathcal{O}(n \log n)$。

C. 多彩的线段 2

题意

- 给定数轴上的 n 条线段，每条线段可以涂上 k 种颜色的一种，要求颜色相同的线段不能相交。求方案数。
- $n \le 5 \times 10^5$，$k \le 10^9$。

- 首先考虑 k 最小是几，答案才能大于 0。
- 这就是经典的 interval partitioning 问题：最少把区间分成几组，才能让组内区间不相交。
- 贪心解法是：所有区间按左端点从小到大排序，维护每一组的右端点 g_j。若存在一组的右端点小于当前区间的左端点 l_i，则将当前区间分入那一组（如果有多组满足要求，随便选一组都可以）；否则新开一组。

C. 多彩的线段 2

- 从这个思路出发。如果我们把 $g_j < l_i$ 的组都看成 $g_j = 0$ 可以发现，无论把区间分到哪一组，排序后的 g 序列都是一样的。因为我们把所有区间按左端点排序了，所以 $g_j = 0$ 将一直保持，直到有个区间分到了那一组里。所以答案就是每次 $g_j = 0$ 的数量乘起来。
- 因为 k 很大，所以用一个堆只维护 $g_j \neq 0$ 的值即可。复杂度 $\mathcal{O}(n \log n)$。
- 如果从弦图的角度解释，本题其实就是弦图的色多项式 $P_G(x) = \prod_{i=1}^{n}(x - d_{G_i}(v_i))$。详见《弦图与区间图》（陈丹琦）定理 2.5。

J. 多彩的生成树

题意

- 给一张完全图，每个点有颜色，共有 n 种颜色，其中第 i 种颜色的点有 a_i 个。两点之间的边权由两点的颜色决定，求该完全图的最小生成树。
- $n \le 10^3$，$a_i \le 10^6$。

- 对每种颜色需要维护：这个颜色的所有点是否在同一连通块里，以及颜色之间的并查集。
- 模仿 Kurskal 算法的过程，按边权从小到大考虑每对颜色 (u, v) 的连边，并分类讨论。

J. 多彩的生成树

情况一： $u = v$

- 若颜色 u 所有点已在同一连通块内则跳过。
- 否则连 $(a_u - 1)$ 条边，标记这个颜色所有点在同一连通块内。

情况二： $u \neq v$

- 若颜色 u 和 v 在同一连通块内则跳过。
- 否则若 u 和 v 所有点都不在同一连通块内，连 $(a_u + a_v - 1)$ 条边。
- 否则若 u 所有点都不在同一连通块内，连 $(a_u - 1)$ 条边。v 同理。
- 否则连 1 条边即可。
- 连边结束后，标记 u 和 v 的所有点在同一连通块内，并连接颜色的并查集。

D. 王国英雄

题意

- 买入 x 件商品需要 $ax+b$ 秒和 px 金币，卖出 x 件商品需要 $cx+d$ 秒但可以赚 qx 金币。有 t 秒以及 m 金币，问 t 秒结束后最多能有多少金币。
- 数值范围 10^9。

- 每次交易额外花费 $(b+d)$ 秒，所以在时间足够的前提下，每次交易一定尽可能多花钱买入商品，以减少交易的次数。
- 注意到如果 $\lfloor \frac{m}{p} \rfloor$ 相同，那么每次交易的结果也是相同的。而每次交易后 m 在不断增加，因此可以通过除法 $\mathcal{O}(1)$ 地把 $\lfloor \frac{m}{p} \rfloor$ 相同的交易合起来处理。设一共要处理 k 种交易。
- 由于买卖一件物品至少需要 2 秒，$(1+2+\cdots+k)\times 2 \le t$ 得出 $k \sim \mathcal{O}(\sqrt{t})$。所以复杂度 $\mathcal{O}(\sqrt{t})$。

H. 阻止城堡

题意

- 棋盘上有城堡（车）和障碍物。处于同一行/列，且中间没有其它城堡或障碍物的一对城堡可以互相攻击。问最少再加几个障碍物可以让所有城堡都不能互相攻击。
- $n, m \leq 200$，坐标范围 $[1, 10^9]$。

- 放一个障碍物最多可以阻止一横一竖两对城堡互相攻击，称这种放障碍物的位置为好位置。
- 把所有处于同一行，且可以互相攻击的一对城堡，看成二分图左边的点；同样地，把所有处于同一列，且可以互相攻击的一对城堡，看成二分图右边的点。这样所有好位置就连接了一个二分图左边的点，和一个二分图右边的点。

H. 阻止城堡

- 为了最小化额外障碍的数量，我们要选尽可能多的好位置。求这张二分图的最大匹配即可。因为可能是完全二分图，所以直接用匈牙利算法，在 $\mathcal{O}(n^3)$ 的复杂度下求最大匹配。
- 添加完二分图对应的好位置之后，剩下的位置最多只能阻止一对城堡互相攻击。这种障碍物直接放在对应城堡的旁边即可。
- 如何维护方案？可以给每一行以及每一列都维护一个 set，保存这一行/列哪些列/行有城堡/障碍物。这样就能快速将障碍物插入到 set 里，以及检查某一行/列是否有两座城堡之间没有障碍物。

M. 回文多边形

题意

- 给一个凸多边形，每个顶点有个权值。选择若干个顶点，使得从某个点开始，逆时针转一圈构成的权值序列是回文序列。求这些顶点的凸包的最大面积。
- 顶点数 500。

- 设 $f(l,r)$ 表示从顶点 l 逆时针转到顶点 r，能构成回文序列的最大凸包面积（$a_l = a_r$）。有转移方程
$$f(l,r) = \max f(l',r') + S_{\Delta ll'r'} + S_{\Delta lr'r}$$
其中 l' 和 r' 位于 l 和 r 之间，且 $a_{l'} = a_{r'}$。
- 直接计算这个转移方程的复杂度是 $\mathcal{O}(n^4)$ 的。注意到对于每种权值，最优情况下，l' 和 r' 其中之一肯定是这个权值在 (l,r) 开区间内最靠近左/右边的顶点。否则额外选择最两边的顶点，面积能变得更大。
- 因此枚举 l' 可以直接确定 r'；枚举 r' 可以直接确定 l'。复杂度降至 $\mathcal{O}(n^3)$。

E. 传感器

题意

- 有一行 n 个红色球以及 m 个监控区间，每次把一个球变成蓝色，问每次变色之后，哪些区间里恰有一个红球。
- 强制在线。
- $n, m \leq 5 \times 10^5$。

- 每个区间都能对应到线段树上的 $2 \log n$ 个节点。如果每次节点内红球的数量变化，就通知节点内的所有区间，最差情况下每次都会通知所有区间。
- 然而我们只想知道恰有一个红球的区间。这说明区间对应的线段树节点中，恰有一个节点里包含一个红球，剩下的节点里包含零个红球。
- 因此使用懒更新的思想，对每个线段树节点维护该节点里还有几个红球。当这个数量变成 1 和 0 的时候，再更新该节点对应的所有区间里还剩几个红球。
- 这样每个区间最多被更新 $4 \log n$ 次，复杂度 $\mathcal{O}(n \log n)$。

L. 路径的交

题意

- 在树上选择 k 条路径，路径的 $2k$ 个端点必须互不相同。最大化被所有路径包含的边权之和。
- 有多次询问，每次询问会临时修改一条边的边权。每次询问的 k 可能不同。
- 节点数、询问数 5×10^5。

- 可能被端点两两不同的 k 条路径包含的边，需要满足把这条边去掉之后，形成的两个连通块大小都大于等于 k。
- 在 k 固定的情况下，这些边形成了一棵树。所以答案就是这棵树的直径。

L. 路径的交

- k 变化的时候怎么办？注意到 $(k+1)$ 的树被 k 的树完全包含，所以可以将所有询问按 k 从小到大排序，依次处理。
- 问题变为：每次询问临时修改一条边的边权，以及在 k 变大时永久删掉若干条边，求树直径。删掉边可以转化为将边权改成 0。
- 这就是经典的动态树直径问题，用线段树在欧拉序上进行维护即可。详见 CF1192B。
- 复杂度 $\mathcal{O}(n \log n + q(\log n + \log q))$。

B. 三角形

题意

- 如果三个字符串中，任意两个串连起来都大于第三个串（两个串连起来有两种方式，存在一种即可），则称这三个字符串构成了三角形。给定 n 个字符串，从中选三个下标不同的串构成三角形，求方案数。
- 单组数据所有字符串总长不超过 3×10^5。

- 记 f_s 表示给定的字符串中，字典序比 s 小的有几个。记 c_s 表示给定的字符串中等于 s 的有几个。假设三个字符串 x，y，z 有 $x \geq y$ 且 $x \geq z$，我们考虑哪些情况可以构成三角形。

情况一：$x = y \geq z$

- 这种情况一定可以构成三角形。根据简单的组合知识，这种情况的答案就是 $\frac{c_x(c_x-1)(c_x-2)}{6} + \frac{c_x(c_x-1)}{2} \times f_x$。

B. 三角形

情况二：$x > y$ 且 $x > z$

- 这种情况的结论是：y 和 z 中至少有一个是 x 的前缀，这里不妨设 y 是前缀。假设 x 去掉前缀 y 之后，剩下的后缀是 s_y，那么 z 必须大于 s_y。

- 结论证明如下：
  - 如果 y 和 z 都不是 x 的前缀，那么由于 $x > y$ 且 $x > z$，所以 $y + z$ 和 $z + y$ 都小于 x，无法构成三角形。
  - 假设 y 是 x 的前缀，如果 $z \le s_y$ 那么 $y + z \le x$；又由于 $x > z$ 那么 $z + y < x$，也无法构成三角形。
- 因此对于这种情况，我们枚举所有字符串的所有前缀。对于每个前缀 y 我们能得到一个初步的答案 $c_y \times \max(0, f_x - f_s - c_{s_y})$。

B. 三角形

- 但是，这里的初步答案包含以下几种重复情况：
  - 如果前缀 y 满足 $x > y > s_y$，那么 $y = z$ 的情况会被重复计算。因此这个情况下答案减去 $\frac{c_y(c_y+1)}{2}$。
  - 如果 y 和 z 都可以作为前缀，而且另一个字符串都大于剩下的后缀，那么三元组 (x, y, z) 就会被算两次。因此对于每个 x，我们还要枚举 $y, z \in \mathrm{prefix}(x)$，若 $z > s_y$ 且 $y > s_z$，答案就要减去 $c_y c_z$。
- 情况 2 其实就是一个二维偏序的计数问题，可以在 $\mathcal{O}(|s| \log |s|)$ 的复杂度内解决。

B. 三角形

- 最后剩下的问题就是如何求 f_s 和 c_s。
- 我们可以用一个小于所有字符的特殊字符（比如 \$）把给定的字符串连起来（例如给定 ab，cde，fg，连起来就变成 \$ab\$cde\$fg），然后求连接后的字符串的后缀数组。我们从小到大枚举所有后缀，并同时维护变量 cnt。若当前后缀的第一个字符恰好位于一个 \$ 后面（也就是说这个字符是某个给定字符串的开头），那么 cnt += 1。此时 cnt 的值就是 f_s 的值（当然要考虑有相同字符串的情况，不过这个处理较为简单）。
- c_s 的值就很好求了，哈希，trie，后缀数组都可以。
- 因此整体的复杂度 $\mathcal{O}(|s| \log |s|)$。

G. 宇宙旅行

题意

- 给定 n 个数 $a_1, a_2, \cdots, a_n$，设 $f(x,k)$ 表示 $a_i \oplus x$ 里第 k 小的数。多次询问，求 $\sum_{x=l}^{r} f(x,k)$。每次询问的 l，r，k 可能都不同。
- 数值范围 2^{60}，询问数 10^5。

- 首先考虑只有一个询问怎么做。令 $g(r,k) = \sum_{x=1}^{r} f(x,k)$，则答案是 $g(r,k) - g(l-1,k)$。
- 考虑怎么求 $g(r,k)$。可以将所有 a_i 建成一棵字典树，用类似于数位 dp 的方式，记忆化字典树每个子树在 full $= 0$ 的情况下的答案：枚举这一位填 0 还是 1，递归求对应子树里关于 k 或者 $k - s$ 的答案，其中 s 是兄弟子树的大小。

G. 宇宙旅行

- 接下来考虑多次询问怎么做。数位 dp 多组询问的经典处理方式是：因为 full $=0$ 的答案对所有询问都是通用的，所以即使处理新询问，也不清空之前的 dp 结果。
- 如果我们对字典树也这样处理，我们将要保存子树大小之和的 dp 结果。
- 注意到字典树所有子树的大小加起来是 $\mathcal{O}(n\log X)$ 的（因为每个叶子只会对它的所有祖先产生贡献），因此可以使用持续记忆化的技巧。仍然沿用只有一个询问时的做法，只是 full $=0$ 的结果要持续保存。
- 复杂度 $\mathcal{O}((n+q)\log X)$。

最后

- 没听明白？没关系。
- 访问 https://sua.ac/wiki/ ，有文字版题解与带注释的参考代码。

Thank you!