第十五届山东省 CCPC 大学生程序设计竞赛

SUA 程序设计竞赛命题组

2025 年 5 月 25 日

赛题设计

- 基础编码能力：D、L；
- 基础思维能力：A、G；
- 经典算法应用：C、E、H、I；
- 进阶思维能力：J、K、M；
- 高级算法应用：B、F。

L. 恒星

题意

- 恒星按从热到冷分成 OBAFGKM 七类，每大类下又从热到冷分成 $0 \sim 9$ 十小类。给两个恒星类别，问两者的冷热关系。

- 按题意模拟，先比较字母关系，再比较数字关系。
- 复杂度 $\mathcal{O}(1)$。

D. 分布式系统

题意

- 某系统有编号从 0 到 $(n-1)$ 的工作节点，接下来要分配 q 个任务，每个任务有编号从 0 到 (a_i-1) 的子任务，且子任务 j 会分配给节点 $(b_i+j) \bmod n$，问最后每个工作节点分到了几个子任务。
- $n, q \le 2 \times 10^5$，$a_i \le 10^9$。

- 对于每个子任务，所有节点至少分配 $\lfloor \frac{a_i}{n} \rfloor$ 个子任务，剩下的 $a_i \bmod n$ 个子任务从节点 b_i 开始，连续分配给接下来的节点。
- 连续分配任务，也就是某个区间的答案加 1。因为只在最后输出答案，可以用差分数组维护。复杂度 $\mathcal{O}(n+q)$。

A. 项目管理

题意

- 有 n 名员工，第 i 名员工的职级为 a_i。现在需要选择尽量多的人，限制条件为，如果选择了员工 i，则被选中的所有人里至多只能有 b_i 个人职级大于 a_i。
- $n \le 2 \times 10^5$。

- 二分答案 x，从小到大考虑职级，考虑从当前职级选 t 个人。
- 假设之前的职级已经选了 s 个人，后面还要选出 $(x - s - t)$ 个人。高职级的人越少，越容易满足限制，因此 s 越大越好，t 也越大越好，也就是每个职级要选出尽量多的人。
- 同一职级的人里，b_i 越大的限制越少，因此考虑当前职级时，按 b_i 从大到小选人即可。复杂度 $\mathcal{O}(n \log n)$。

G. 装配线

题意

- k 名工人加工 n 个工件，第 i 个工件在第 t_i 分钟的开头加入工人 w_i 的收件箱。
- 每分钟，工人从收件箱里拿出一个工件，完成加工后放入下一个工人的收件箱（如果是最后一个工人则加工完成）。问所有工件加工完成需要几分钟。
- $k \le 10^9$，$n \le 2 \times 10^5$。

- 工件 i 原本的结束时间是 $(t_i + k - w_i)$，但每个时间点只能完成一个工件。
- 因此设 a_i 表示从小到大排序后的工件完成时间，依次进行更新 $a_i \leftarrow \max(a_i, a_{i-1} + 1)$ 即可。复杂度 $\mathcal{O}(n \log n)$。

H. 最小生成树

题意

- 给一张带边权的连通无向图，可以最多再添加 k 条边。设添加的边连接节点 u 和 v，则边权为 $|u-v|$，可以出现重边和自环。问加边后，最小生成树最小是多少。
- $n, m, k \leq 2 \times 10^5$。

- 能添加的边，边权至少为 1。先求出原图的最小生成树，如果我们从大到小把边权大于 1 的树边删掉，能都用边权是 1 的新边来补充吗？
- 答案是可以的。把这些边删掉后，我们得到了若干个连通块。从 1 到 $(n-1)$ 检查，若 i 和 $(i+1)$ 不在同一连通块，则连边 $(i, i+1)$。复杂度 $\mathcal{O}(n\alpha + m\log m + k)$，其中 α 是反阿克曼函数。

I. 方阵谜题

题意

- 给一个九宫格，每次可以选一行右移，选一列下移，或整体顺时针旋转 90 度。T 次询问，每次给出初始和目标状态，问最少步数。
- $T \le 2 \times 10^5$。

- 每个格子里的数字其实不重要，只有每个格子最后去哪了才重要。格子里的数字其实就是用来标识某个格子去哪了，具体是几不重要。
- 所以总是可以把初始状态映射成按顺序从 1 到 9（称为状态 S），同时也得到了目标状态的映射。
- BFS 预处理从 S 到其它每个状态的最小步数，每次询问直接用映射后的目标状态查表输出答案即可。复杂度 $\mathcal{O}(n! \times n + Tn)$，其中 $n = 9$ 是格子数量。

E. 最大公因数

题意

- 给定正整数序列 a_i，进行恰好 k 次操作，每次操作选一个元素加一。最大化所有元素的最大公因数。
- $n, \max a_i \le 10^6$，$k \le 10^{12}$。

- 首先，答案一定是 $s = \sum a_i + k$ 的因数。因此对 s 因数分解，然后在因数中枚举答案 x。
- 为了让答案为 x，需要把每个 a_i 增加到最近的 x 的倍数，剩下的操作次数还需要被 x 整除。
- 如果 $x \le \max a_i$，那么每 x 种数都要增加到同一个数，可以一起计算。这种情况的复杂度为调和级数 $\mathcal{O}(\max a_i \log \max a_i)$。
- 如果 $x > \max a_i$，那么所有数都增加到同一个数，直接 $\mathcal{O}(1)$ 计算。
- 整体复杂度 $\mathcal{O}(n + \sqrt{s} + f(s) + \max a_i \log \max a_i)$，其中 $f(s) \approx 6 \times 10^3$ 是因数个数。

C. 括号整数

题意

- 用 $0 \sim 9$ 表示括号的种类，称一个整数 $b_1 b_2 \cdots b_n$ 是括号整数，若存在一个合法括号序列，使得第 i 个括号的种类是 b_i。给一个整数 A，求小于等于它的最大括号整数。
- $11 \le A \le 10^{2\times10^5}$。

- 按位枚举答案，即枚举答案和 A 的最长公共前缀 $P = p_1 p_2 \cdots p_k$，再从 0 到 $a_{k+1} - 1$ 枚举第 $(k+1)$ 位是什么，假设是 d。
- 假设用 Pd 走一遍判断合法括号序列的流程，当前的栈从底到顶的元素为 $s_1 s_2 \cdots s_t$，则最大括号整数一定是 $Pd9999 \cdots s_t s_{t-1} \cdots s_1$。
- 公共前缀越长，答案越大；同时栈里元素越少，答案就越大。为了让栈里元素最少，只要入栈元素和栈顶元素相同，就弹出栈顶元素；否则入栈。
- 若没有前缀能构造出答案，则答案为 $999...999$（偶数个 9，数量小于 A 的长度）。复杂度 $\mathcal{O}(nb)$，其中 $b = 10$ 是进制。

K. 路径规划 2

题意

- 给定 $n \times m$ 的网格，每个格子里有一个数。问所有从左上走到右下，只能往右往下的路径里，经过的整数的 mex 最小是多少。
- $n \times m \leq 10^6$。

- 从小到大枚举答案 x，检查是否有路径能不经过 x 走到终点。
- 错误解法：使用“对偶图”，看是否有 x 的八连通块从左下连到右上。反例：考虑 $x = 0$，如下图

1	1	1	1
1	0	1	1
0	1	1	0
1	1	0	1
1	1	1	1

K. 路径规划 2

- 正确做法：用类似归并排序的方式，维护每行能到达哪些区间。每个区间尽量向右合并，下一行的 x 会把每个区间破开。
- 如果遇到空行，则直接把所有区间合并，右端点改成 m，然后跳到下一个有 x 的行。
- 由于每个 x 只会把一个区间分成两个，且每个区间只会被合并一次，因此复杂度为 $\mathcal{O}(nm)$。

J. 有用的算法

题意

- 给一个二分法的实现：寻找元素 k，如果位置 `mid` 的元素小于 k 则 `head = mid + 1`，否则 `tail = mid`。`head` 和 `tail` 最终将同时指向目标位置。
- 给定 n 和 k，称一个 n 的排列 P 是好的，若直接把 P 和 k 作为二分法的输入，真的能找到 k。求好排列有几个，答案除以 $n!$ 后取模输出。
- $n, k \leq 10^9$，数据组数 10^4。

J. 有用的算法

- 维护 $f(l, x, y)$ 表示二分区间的长度已经缩小到了 l，之前的检查点已经用了 x 个小于 k 个数和 y 个大于等于 k 的数。
- 若 $l > 1$，考虑当前检查点用小于 k 还是大于等于 k 的数，转移方程 $f(l, x, y) = f(\lfloor \frac{l}{2} \rfloor, x + 1, y) + f(\lceil \frac{l}{2} \rceil, x, y + 1)$
- 若 $l = 1$，由于 `mid` 的元素大于等于 k 时，`tail` 会指向 `mid`，所以如果 `tail` 有移动，那么它最后一次移动到的元素必须是 k。因此按 y 分类讨论：
  - 若 $y = 0$，则 $f(1, x, 0) = \binom{k-1}{x} \times x! \times (n - x)!$。
  - 若 $y > 0$，思路为：小于 k 的选 x 个，大于 k 的选 $(y - 1)$ 个，$(x + y)$ 个检查点里哪些是小于 k 的，小于 k 的数内部怎么排顺序，大于 k 的数内部怎么排顺序，即 $f(1, x, y) = \binom{k-1}{x} \times \binom{n-k}{y-1} \times \binom{x+y}{x} \times x! \times (y - 1)! \times (n - x - y)!$。
- 记忆化搜索即可，一共只有 $\mathcal{O}(\log^2 n)$ 个状态。最后答案除以 $n!$ 是为了防止引入对 $n!$ 的计算。
- 如果怕用 `unordered_map` 记状态太慢，可以记 $f(l \bmod 2, x, y)$，请读者思考为什么这样是正确的。

M. 三角剖分

题意

- 把正 n 边形三角剖分，给出剖分后的 $(n-2)$ 个三角形，把它们拼回正 n 边形。
- $n \le 2 \times 10^5$。

- 首先一定有一个三角形经过正 n 边形的中心，把它先随便放好。
- 放好以后，三角形把正 n 边形分成了三块，递归进入这三块。假设某一块对应的三角形边长为 l，则找到任意一个最长边为 l 的三角形放进去，继续递归构造即可。
- 做法的正确性来自于正 n 边形的对称性。复杂度 $\mathcal{O}(n)$。

F. ACE 字符串

题意

- 给长度为 n 的字符串，求最长的子串，使得该子串可以被分成非空的五份，其中第一份等于第三份等于第五份。
- $n \leq 3 \times 10^5$。

- 对原串建立后缀自动机（SAM）得到 parent 树（也就是反串的后缀树），每个结点 u 对应一类子串，其中最长的子串长度是 len_u，所有子串的长度范围在 $(len_{fa_u}, len_i]$（fa_u 是 parent 树上 u 的父亲），在原串里出现位置的右端点的集合都是 $right_u$。这部分复杂度 $O(n\Sigma)$，其中 Σ 是字符集大小。
- 对后缀数组的 `height` 建立笛卡尔树也可以得到一个类似的树形结构，这部分复杂度 $O(n)$。

F. ACE 字符串

- 先从 $right_u$ 里选出 $x < z < y$，分别对应第一份、第三份、第五份的出现位置的右端点，那么答案是 $y - x + l$，这里 l 是第一份（也是第三份和第五份）的长度，显然 x 取最小值，y 取最大值。由于第二份和第四份非空，有

$$l = \min\{z - x - 1, y - z - 1, len_u\}$$

那么 z 应该距离 $\frac{x+y}{2}$ 越近越好。
- 注意这里即使 $l \le len_{fa_u}$ 也没有关系，因为 $right_u \subseteq right_{fa_u}$。
- 启发式合并维护 $right_u$，取出最小值 x 和最大值 y，查询集合里距离 $\frac{x+y}{2}$ 最近的 z 即可，这部分复杂度 $O(n \log^2 n)$。

B. 弹球

题意

- 有 $y = 0$ 和 $y = H$ 两条直线，还有 n 块水平小木板（长度厚度忽略不计）。q 次操作，每次操作要么放入一个木板，要么删掉一块木板，要么问一颗弹球从 (x, y) 呈 45 度角出发，碰到直线或木板会反弹，x 坐标走到 g 时 y 坐标是多少。
- $n, q \leq 10^5$。

- 代码题。
- 把在同一条路径上的木板上下表面连起来，看作一个链表。放入和删除木板就是拆分已有的链表再合并，询问弹球位置就是问某一条路径上 $x = g$ 时的 y 坐标。
- 用平衡树维护有序链表的拆分、合并、查询即可。复杂度 $\mathcal{O}((n+q)\log(n+q))$。

Thank you!