2024 第十届中国大学生程序设计大赛 (哈尔滨站) 题解

电子科技大学

2024 年 10 月 20 日

概况

- Easy: C、G、M
- Easy-Medium: B、K
- Medium: A、J、L
- Medium-Hard: D、E、I
- Hard: F、H

C. 在哈尔滨指路

题目大意

- 在网格图上指路，你知道的是绝对位置，比如从这个路口往南走 2 个路口再往东走 1 个路口，你要输出一个相对序列，比如面向南，沿街直走 2 个十字路口左拐再走一个十字路口
- 关键词：模拟

本题有 2 种做法

- 按照相邻两次绝对方位的关系判断左转或右转即可。
- 求出最终位置，然后按最短路径行走。但需要注意可能存在最终位置和出发位置相同的情况。

G. 欢迎加入线上会议！

题目大意

- 给出一张 n 个点的无向图（不一定连通），保证每个点的度都大于等于 1。找出此图的一棵生成树，并满足给定的 k 个节点的度为 1。
  数据范围：$2 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq 5 \cdot 10^5$
- 关键词：图论

Fun fact:

Welcome

to join the conference

- 本题的做法很多，下面介绍其中一种。
- 下文称需要满足节点在生成树中度数为 1 的点为特殊点。对于无向图的每一条边，考虑对特殊点进行缩点操作：
  - 如果一条边所连接的两个点都不是特殊点，则新图中保留这条边；
  - 如果有一个点是特殊点，如果这个特殊点没有被进行缩点操作，则将特殊点缩到非特殊点中，新图中不保留这条边；
  - 如果两个点都是特殊点，则新图中也不保留这条边。
- 对于新图，如果为空或不连通，则无法找到一棵原图中满足要求的生成树，否则某个被缩点的特殊点将是割点，这个点的度将大于 1。如果新图是连通的，找出新图的一棵生成树即可，注意需要在原图中恢复特殊点。
- 时间复杂度：$\mathcal{O}(n+m)$，空间复杂度：$\mathcal{O}(n+m)$。

M. 奇怪的上取整

题目大意

- 给出了一个「上取整」程序 $f(a,b)$，求 $\sum_{i=1}^{n} f(n,i)$ 。
  数据范围：$1 \le n \le 10^9$。
- 关键词：质因数分解

- 可以发现 $f(a,b)$ 中循环变量 i 为小于等于 b 的整数里能整除 a 的最大整数。
- 因此考虑 i 一定是 a 的因子。对于 $f(n,p)$ $(1 \le p < n)$，记 n 的因子从小到大排列为 $d_1, d_2, \ldots, d_m$，如果 $d_i \le p < d_{i+1}$ $(1 \le i < m)$，则求 $f(n,p)$ 中的循环变量将停在 d_i 位置，返回值为 $\frac{n}{d_i}$。考虑 $p = n$，则 $f(n,n) = 1$。
- 综上，答案为：

$$1 + \sum_{i=1}^{m-1} (d_{i+1} - d_i) \frac{n}{d_i}$$

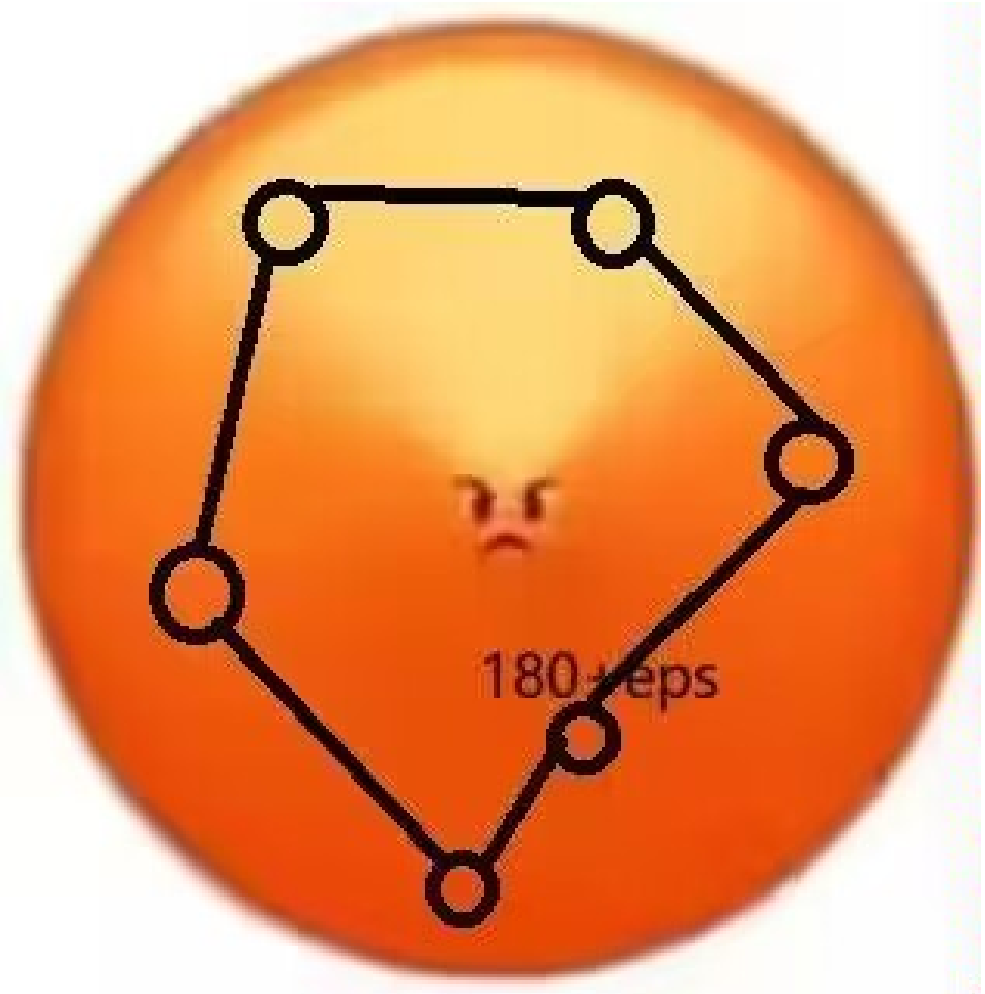
- 对 n 进行因数分解后求值即可。
- 时间复杂度：$\mathcal{O}(\sqrt{n})$，空间复杂度：$\mathcal{O}(d(n))$，其中 $d(n)$ 为 n 的因子个数。

B. 凹包

题目大意

- 二维平面上的点集，要求选择若干个点以及点之间的连接顺序，使得这些点连成一个面积严格 > 0 的简单凹多边形，最大化这个凹多边形的面积，$n \leq 10^5$。
- 关键词：凸包，双指针

Fun fact:



- 首先可以发现，我们一定会选择到凸包上的所有点，如果没选完，那么选上，将凸包上的点按照极角序插入这些点内，一定不会使得凹包变成凸包，并且会使面积增大。
- 考虑现在已经有一个凸包了，我们要选择一些剩下的点插到凸包的点之间。首先是一些退化的情况，如果所有点都在凸包上，那一定不合法。否则任意顺序插入任意一个点都能使得凸包变成凹包，因为要最大化面积，而每插入一个点肯定都会减少面积，所以我们至多只会插入一个点，设为 x。
- 我们也不会改变原来的连接顺序，只会将原来的一条边 (a, b) 变成 $(a, x), (x, b)$，减少的面积就是 $|ab| \cdot \mathrm{dis}(x, ab)$，$\mathrm{dis}$ 即点 x 到 ab 直线的距离，直接枚举所有边和 x 显然会超时。

- 但可以发现，最优的 x 一定是删去凸包之后，剩下的点，组成的凸包上的点（如果不退化），因为我们要尽量最小化 $\mathrm{dis}(x, ab)$。
- 而对所有边找到凸包上距离这个边最近的点，可以直接双指针维护，对于每条外层凸包的边，维护距离最近的点，计算插入这个点后减小的面积取最小值依次移动边之后最近点的变化即可，注意可能需要处理内层凸包退化为一个或两个点的情况。
- 复杂度 $\mathcal{O}(n \log n)$。

K. 农场经营

题目大意

- 有 n 种作物，每天工作恰好 m 个单位时间，对于每种作物，处理一单位时间该种作物收益为 w_i，处理这种作物的工作时长可以是 $[l_i, r_i]$ 中的一个整数。现在可以删除最多一种作物工作时长的限制，也就是处理这种作物的工作时长变为 $[0, +\infty)$ 中的一个整数，但要保证每天仍然恰好工作 m 小时。问最大收益。
- 关键词：线段树

- 考虑先对作物按 w_i 从大到小的顺序排序并按该顺序重新编号，下文中下标均是重新编号后的。
- 对于不删除限制的情况，记 t_i 为处理第 i 种作物的工作时长，初始时 $t_i = r_i$。如果 $\sum t > m$，则按 w_i 从小到大的顺序降低 t_i，直到 $\sum t = m$。注意在这个过程中应保持 $t_i \geq l_i$。
- 这样的情况是最优的，在这种情况下对于分配给 i 作物和 j 作物的一个单位时间，交换它们的收益一定不是正的。
- 之后考虑删除限制的情况。在未删除限制的情况下，对于删除第 i 个作物的限制，有两种决策：
  - 将所有小于 w_i 的作物 j 的处理时间均变为 l_j，将多出的时间全部加给 i。
  - 将作物 i 的处理时间变为 0，处理 i 作物的时间分配给大于 w_i 的作物。如果还有剩余，则不往 w 更小的作物分配。

- 对于第一种决策，由于已经调整完毕，如果 j 存在可分配的时间，则大于 w_j 的作物一定没有接收这部分时间的余量。如果存在余量，可以减少 j 可分配的时间给大于 w_j 的作物，使得初始情况获得更大的收益，因此只能将这部分余量分配给 i。可以考虑统计 $t-l$ 和 $w(t-l)$ 的后缀和，枚举 i 计算小于 w_i 的作物将给答案带来的贡献。
- 对于第二种决策，可以统计 $r-t$ 和 $w(r-t)$ 的前缀和，对于每种作物 i，在 $i-1$ 前缀中二分到第一个 $r-t$ 的前缀和大于 t_i 的位置，然后计算贡献。
- 时间复杂度：$\mathcal{O}(n\log n)$，空间复杂度：$\mathcal{O}(n)$。

A. 造计算机

题目大意

- 给定 L，R，你要构造一个 DAG，满足仅有一个起点和一个终点。每条边有权值 $0/1$。从起点 dfs，把经过的 $0/1$ 边权记录下来，每条到终点的路径会是一个二进制数（不含前导零、不重复），所有二进制数刚好是 $[L, R]$ 内每个数的二进制，要求 DAG 的节点数不超过 100，且每个节点的出度不超过 200。$L \leq R \leq 10^6$。
- 关键词：构造

- 考虑构建一个由 20 个节点构成的子图（以下简称为子图），满足第 i 个节点与第 $i+1$ 个节点之间存在两条权值分别为 0 和 1 的边，最后一个节点连向终点，同样建两条权值为 0 和 1 的边。这样的话，通过连向子图的某个节点，当我们已经构造出 xxx 时，我们可以构造出任意长度的 $xxx00..00$、$xxx00...01$ 、...、$xxx11...11$。
- 对于询问 $[L,R]$ 我们可以把它拆成若干个子询问，分别利用子图来构造。

- 具体来说，我们可以从 L 在二进制表示下的低位枚举到高位，遇到一个 $'0'$ 时，考虑把 0 变成 1，同时右边所有数全部改为 0，得到一个新的数 L'；全部改为 1，得到一个新的数 R'。
- 若 $R' \leq R$，则拆分出一个新的子询问 $[L', R']$，然后枚举下一位；
- 否则的话，我们考虑开始从 R 在二进制表示下的高位枚举到低位（从第二位开始枚举），遇到一个 $'1'$ 时，考虑把 1 变成 0，同时右边所有数全部改为 0，得到一个新的数 L'；全部改为 1，得到一个新的数 R'。此时同样拆分出了一个新的子询问 $[L', R']$，然后继续枚举下一位，直到枚举完所有位。

- 举一个具体的例子：
- L = 00110101, R = 11101110
- 询问 $[L, R]$ 可以被拆成以下若干子询问：
  - L1 = 00110101, R1 = 00110101
  - L2 = 00110110, R2 = 00110111
  - L3 = 00111000, R3 = 00111111
  - L4 = 01000000, R4 = 01111111
  - L5 = 10000000, R5 = 10111111
  - L6 = 11000000, R6 = 11011111
  - L7 = 11100000, R7 = 11100111
  - L8 = 11101000, R8 = 11101011
  - L9 = 11101100, R9 = 11101101
  - L10 = 11101110, R10 = 11101110

- 使用上述拆分方法，可以保证任意两个子询问的交为空，且所有子询问的并等于 $[L, R]$。并且对于每个子询问 $[L_i, R_i]$，都满足 R_i 与 L_i 去掉 LCP 后剩余的后缀分别是 $11...1$ 和 $00...0$，这个可以直接通过向子图连边来构造。因此只需要考虑对 L_i 和 R_i 的 LCP 建字典树即可。
- 不难发现，建出来的字典树实际上和对 L 和 R 建字典树是类似的（对于每对 $[L_i, R_i]$，L_i 与 R_i 的 LCP 一定是 L 或 R 的某个前缀后接 0 或 1）。字典树结点个数不超过 40，子图的节点个数不超过 20，因此满足总节点数不超过 100。
- 子图中可能有些节点没被用到，要记得删掉。

J. 新能源汽车

题目大意

- 含 n 种电瓶的车，每种电瓶上界 a_i，耗 1 单位任意电瓶种的电力前进 1（只能向前），有 m 个充电站，每个充电可以给一个指定的电瓶充电。求初始电瓶满的情况下最远可以行驶多远。
- 关键词：贪心

- 以下 n, m 同阶。
- 考虑一个 n^2 的贪心，我们从一个加油站 j 走向 $j+1$ 的时候，一定是先用能先加到的油，这样模拟就是 $\mathcal{O}(n^2)$。
- 考虑加速这个过程，我们的贪心需要支持不断删去前缀的数，修改第一个数，插入一个数，用堆即可维护，复杂度 $\mathcal{O}(n \log n)$。

L. 树上游戏

题目大意

- 给定一棵 n 个点的树，从 $\frac{n(n-1)}{2}$ 条简单路径中等概率随机选择两条路径（选择的路径可以相同），记两条路径的公共边数量为 X，求 $E(X^2)$，结果对 998244353 取模。
- 关键词：dfs，期望

- 显然总共有 $(\frac{n(n-1)}{2})^2$ 种选法，先计算所有选法的 X^2 之和，然后除以 $(\frac{n(n-1)}{2})^2$ 即可。
- 考虑两条路径的公共边为 $e_1, e_2, ..., e_k$ ，那么对 X^2 之和的贡献为 $(|e_1| + |e_2| + ... + |e_k|)^2 = \sum_{i=1}^{k} |e_i|^2 + \sum_{1 \le i,j \le k} |e_i| \cdot |e_j|$ ，本题中所有边边长 $|e_i|$ 均为 1 。
- 因此一种选法的贡献可以分成两种，一种由公共边 e_i 产生的，一种由公共边的边有序对 (e_i, e_j) 产生。
- 不妨设 1 为树的根，记 $siz[u]$ 为 u 的子树大小，$sum[u]$ 是 u 子树内 siz 的平方和即 $\sum_{v \in subtree_u} (siz[v])^2$ 。

- 对于第一种贡献，考虑每一条边 (u, v) 在多少种选法中会成为公共边即可。不妨设 u 是 v 的父节点，则只需两条路径的端点均分别在 (u, v) 两侧即可，故选法有 $(siz[v])^2 \cdot (n - siz[v])^2$ 种。
- 对于第二种贡献，需要计算每个边有序对 (e_i, e_j) 在多少种选法中有贡献。考虑在 u 处仅计算两条边端点的最近公共祖先为 u 的边有序对的贡献。分两条边位于 u 的同一子树分支和不同子树分支两种情况分别讨论：
  - 若两条边位于 u 的同一子树分支，由于仅考虑边端点的最近公共祖先在 u 的情况，因此存在一条边其中一个端点是 u ，另一个端点是 u 的某个儿子 v 。另一条边则位于 v 的子树中，易得选法有 $\sum_{v \in son_u} 2 \cdot (n - siz[v])^2 \cdot (sum[v] - (siz[v])^2)$ 种。
  - 若两条边位于 u 的不同子树分支，则选法有 $\sum_{v_1, v_2 \in son_u} sum[v_1] \cdot sum[v_2]$ 种。
- 一次深度优先搜索即可实现上述统计。
- 时间复杂度 $O(n)$ ，空间复杂度 $O(n)$ 。

E. 弹珠赛跑

题目大意

- m 个球，速度为 v_i，n 个位置 $x_1, \ldots, x_n$，每个球都会随机选择一个位置，然后向正方向移动，求坐标中位数在原点的期望时间，对 $10^9 + 7$ 取模
- $n, m \leq 500$
- 关键词：DP

- 设 $m = O(n)$，一个显然的想法是可能的时间答案一定是某一个 $t = \frac{x_i}{v_j}$，于是我们考虑枚举 $n * m$ 种可能的时间 t，然后求在这个时间，恰好有 $\lfloor \frac{m}{2} \rfloor$ 个球还没经过原点的概率。
- 那么对于其他的每个球，我们可以枚举所有位置，计算出有多少概率在 t 的时候还没经过原点。那么就是每个球有 p_i 的概率没经过，$1 - p_i$ 的概率经过，求恰好有 $\lfloor \frac{m}{2} \rfloor$ 个球还没经过原点的概率。这个可以直接背包，这样单次复杂度是 $\mathcal{O}(n^2)$ 的，总复杂度 $\mathcal{O}(n^4)$，不能接受。

- 重新观察所有球的状态，总共 n^2 种时间 t，看起来所有的 p_i 的数量是 $\mathcal{O}(n^3)$ 的，但实际上对于每个球，如果不是中位数，每个球状态只有 $\mathcal{O}(n)$ 种，总状态其实只有 $\mathcal{O}(n^2)$。而这只取决于 t 的大小。
- 所以这启发我们将所有的 $\mathcal{O}(n^2)$ 个 t 按照大小排序。从小到大枚举每个 t 考虑。可以发现这时候每次 $t = \frac{x_i}{v_j}$ 变化，只有当前这个 t 对应的球的 p_j 产生了变化，而其他的球并不改变。所以我们可以维护所有球的整个 dp 数组，每次枚举的时候撤销这一个球的影响，然后加上概率变化后的贡献重新计算即可，而这个背包显然是可撤销的。
- 也可以把每个球看成一个多项式 $p_i x + (1 - p_i)$，求的是 $x^{m/2}$ 项的系数，每次修改就是除一个单项式，乘一个单项式，这时候单次计算的复杂度就是 $\mathcal{O}(n)$ 的，总复杂度 $\mathcal{O}(n^3)$ 。可以通过。

I. 一道全新的几何问题

题目大意

- 给你一个正整数的多重集合你每次操作可以选择删除一个数/加入任意一个正整数问你最小的操作次数使得这个集合的数的乘积为 M, 和为 S
- $M, S \leq 10^{10}$ 集合初始大小 $\leq 10^5$ 初始集合每个数 0^{10}
- 关键词：DP, 质因子分解

- 乘积为 M，所以只需要考虑 M 的因数。
- 一个暴力的想法是找出所有 M 的乘积分解，假设是 $2 \le x_1 \le x_2 \le ... \le x_t$，且 $\prod x_i = M$，那么代价就是 $S - \sum x_i$ 个 1 加上所有 x 和原集合求对称差。
- 优化可以考虑设置一个 B，小于 B （除了 1）的因子使用 dp，然后去搜索 $B \le x_1 \le x_2 \le ... \le x_t$，且 $(\prod x_i) | M$ 的所有方案。

- dp 具体可以设 $f_{i,j,s}$ 表示考虑了前 i 个因子，现在的乘积是 j，和是 s，的最小代价，转移就是枚举下一个因子用了多少个即可。
- 令 $\sigma(n)$ 表示 n 的因子个数函数。f 的状态数 $\mathcal{O}(\sigma(M)B^2\log M)$，转移复杂度 $\mathcal{O}((\sigma*\sigma)(M)B\log M)$，$(\sigma*\sigma)$ 是狄利克雷卷积。1 的个数要特殊处理一下。
- 实测 $B=500$ 左右比较优秀。

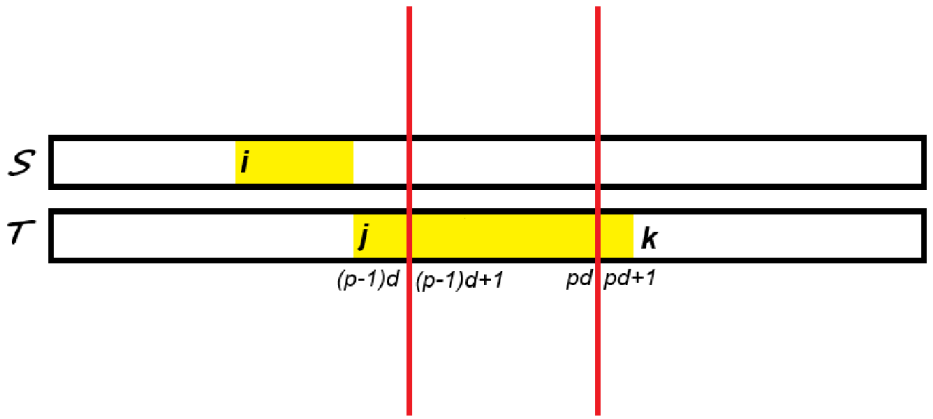
D. 一个朴素的字符串问题

题目大意

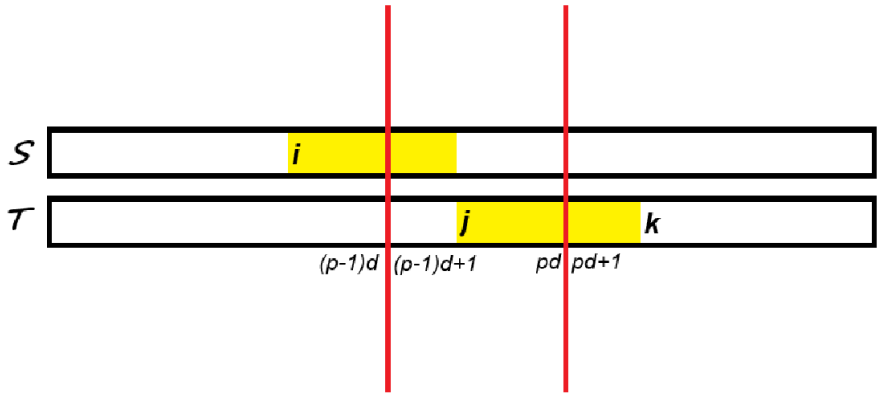
- 给你一个 $2 \times n$ 的字符矩形，你可以从其上任意一个位置出发向右或向下走走出一条路径。问将路径上经过的所有字符串起来后能够得到的最长的平方串的长度是多少。这里平方串指能够写成 $S^2 = SS$ 形式的字符串，如 `abcabc`、`tt`。
- 关键词：字符串，后缀数组

- 方便起见，我们将字符矩阵的两行分别写为 S 和 T。此外为了方便，我们在 S 的末端和 T 的前端分别加上两个新字符 $\$_1$ 和 $\$_2$（并令 $n = |S| = |T|$）。则问题转化为在对于 $1 \le i \le j \le k \le n$，$S[i,j)T[j,k)$ 是平方串的条件下求 $k-i$ 的最大值。
- 首先，别忘了 S 和 T 本身就可能包含平方串，故而显然我们需要对这两个字符串单独求其平方子串。这也意味着能够解决这一问题的算法必然也能求出任意一个字符串的平方串。这提示我们从已有的求平方子串算法进行改造。
- 我们考虑这样一个算法：（从大到小）枚举可能的平方串长度 $2d$，将 $[1,n]$ 分解为 $[1,d],[d+1,2d],\ldots,[pd+1,n]$ 共 $\lceil \frac{n}{d} \rceil$ 个子区间。如果存在一个长为 $2d = k-i$ 的平方串 $S[i,j)T[j,k)$，则区间 $[i,k)$ 必然完全包含这 $\lceil \frac{n}{d} \rceil$ 个子区间中的至少一个。不妨考虑枚举区间 $[(p-1)d+1, pd]$，按照分割点 j 在区间左侧、区间内和区间右侧三种情况进行分类讨论。

- $1.j \le (p-1)d+1$：首先求出 T 中包含 $[(p-1)d+1, pd]$ 且周期为 d 的最长字串 $[j,k)$，再向左贪心拓展即可。具体的
  - $j = (p-1)d+1-\mathrm{lcs}(T[1,(p-1)d], T[1,pd])$
  - $k = pd+1+\mathrm{lcp}(T[(p-1)d+1,n], T[pd+1,n])$
  - $i = j-\mathrm{lcs}(S[1,j-1], T[1,j-1+d])$
  - 判断 $k-i \ge 2d$ 是否成立即可（注意到这里同时也可以判断 T 中是否含有平方串）



- 2. $(p-1)d+1 < j \le pd$：考虑尽可能贪心地让 S 和 T 在中间匹配更多地位置，具体而言可以转化为判断 $\mathrm{lcp}(S[(p-1)d+1,n],S[pd+1,n])+\mathrm{lcs}(S[1,(p-1)d],S[1,pd]) \ge n$ 是否成立即可



- 3. $j > pd$：分割点在区间右侧的情形，和第一种情况类似，不再赘述
- 可以使用 SA+RMQ 实现 $\mathcal{O}(n\log n)-\mathcal{O}(1)$ 的 lcp、lcs 查询。时间复杂度 $\mathcal{O}(n\log n)+\sum_{d=1}^{\lfloor n/2\rfloor}(n/d)\times\mathcal{O}(1)=\mathcal{O}(n\log n)$。使用哈希求 lcp、lcs 会多一个 log。

F. 一维星系

题目大意

- 数轴上 N 个球，初始每个球在 x_i 的位置，重量为 w_i，每一秒每个球统计严格在它左右两边的球的重量和，左边 $>$ 右边就往左一步，左边 $<$ 右边就往右一步，等于就不动，Q 个询问，问你 t_i 秒时第 a_i 个球所在位置
- $1 \le N, Q \le 10^5, -10^9 \le x_i, w_i \le 10^9$ $0 \le t_i \le 10^9$
- 关键词：模拟，数据结构

- 容易发现，任意两个球如果出现在同一位置，那它们就不再会分开，可以被视为一组球；还有一种特殊情况是两个相邻的球不断交换位置，它们也可以被视为一组球。
- 因此整个过程可以分为若干个阶段，每个阶段中，一组球一定处于某种状态：整组球在一个位置，向左/右移动或者不动；整组球在两个相邻位置，不断交换。每次若两组球接触，则进入下一个阶段。可以证明，总共的阶段数为 $O(N)$。
- 因此，只需要模拟整个过程，并离线所有询问，按照时间顺序回答即可。模拟整个过程有多种做法，可以用一个优先队列来储存现存的相邻两组球所需的合并时间，只需要在两组球接触时按照规则判断它们是否合并，合并后的移动方向是什么，并更新优先队列即可。时间复杂度 $O(N \log N)$。

H. 子序列计数

题目大意

- 数列 s, t，s 以 n 个 pair (len, v) 给出，表示从左往右长为 len 个 v 的连续段逐一拼接形成 s，s' 为 $s'_{i*k} = s_i$ 构成的序列，保证 $\gcd(k, len(s)) = 1$，求 s' 有多少个子序列为 t，对 $10^9 + 7$ 取模，
- $len(t) \leq 10, len(s) \leq 10^9$, $n \leq 1000$
- 关键词：类欧几里得，矩阵优化 dp

- 令 $d = k^{-1} \pmod{L}$。
- 考虑将统计子序列的 dp 写成矩阵乘法，那么实际上题目里的打乱就是需要我们将 $\{s\}$ 的对应矩阵按照 $s_0, s_d, s_{2d}, \cdots$ 的顺序乘起来。
- 我们将这个问题写成 $f(L, d, M)$，M 维护我们现在的矩阵段。

- 我们可以支持以下两种转移：
  - 1. 当 $2d > L$，$f(L, d, M) \to f(L, L-d, M')$：这个是平凡的，只需要把 0 处不动，剩下矩阵段翻转即可。
  - 2. 否则，$f(L, d, M) \to f(d, d-(L \bmod d), M')$：这个操作就是我们把整个 M 按 d 一块切开，然后叠成一个最后一行可能有缺的矩形，然后每一列按顺序乘起来。可以用线段树 + 矩阵快速幂实现。并且这个操作不会增加矩阵的段数。
- $L = 1$ 就是终止状态，此时 M 也只有一个矩阵，就是答案。
- 复杂度主要在 2 转移，是 $\mathcal{O}(nm^3 \log n \log L)$，常数极小。

Thank you!