

CCPC2026 黑龙江省大学生程序设计竞赛题解

吉林大学出题组 拱垵

2026 年 5 月

预估难度与实际难度

预估难度（从易到难）

F | G H M C B | J E A L | I D K

实际难度

题号	标题	通过
F	复读机	234
G	卡卡数	185
H	周二会员日	156
M	六个核桃	179
C	圆桌会议	113
B	最大数码 2	94
J	秋水横渡	69

题号	标题	通过
E	乐乐的杏花	21
A	卡卡杀	18
L	猪猪车	2
I	假想对冲	0
D	格兰法洛	1
K	卡卡拆	1

Problem F. 复读机：题目大意

题目大意

输入一个整数 x ，按 $x = 1$ 、 $x = 0$ 、其他三种情况输出固定字符串。

数据范围与算法目标

$O(1)$ 时间， $O(1)$ 空间。

Problem F. 复读机

判定

$x = 1$ 输出 I love Heilongjiang。

$x = 0$ 输出 I love CCPC。

其他值输出 Error。

输入是单个整数，直接比较即可，不需要额外计算。

输出的大小写和空格要与题面完全一致。

边界

题面只分三类情况，负数或大于 1 的数都归为 Error。

Problem F. 复读机：步骤

步骤

读入整数 x 。

按三类情况输出固定字符串。

Problem G. 卡卡数：题目大意

题目大意

统计 $[0, 10^x - 1]$ 中所有整数写成十进制后，数字 1 出现的总次数，模 998244353。

数据范围与算法目标

x 可到 10^6 ， $O(n)$ 或 $O(\log n)$ 皆可。

Problem G. 卡卡数

补零与对称

把区间内所有数补成恰好 x 位十进制串（允许前导零）。共有 10^x 个串，一共有 $x \cdot 10^x$ 个数码位。

固定任一位置，该位上的 $0 \sim 9$ 出现次数完全相同，每个数字出现 10^{x-1} 次。

所以数字 1 在每一位出现 10^{x-1} 次，总计 $x \cdot 10^{x-1}$ 次。

答案与计算

当 $x = 0$ 时区间只有 0，答案为 0。

当 $x > 0$ 时，答案为 $x \cdot 10^{x-1} \bmod 998244353$ 。

10^{x-1} 用快速幂计算即可。

Problem G. 卡卡数：例子

小例子

$x = 2$ 时枚举 $00 \sim 99$ ，每一位出现 10 个 1，总计 $2 \times 10 = 20$ 。

Problem H. 周二会员日：题目大意

题目大意

给出柠檬误解后的单个汉堡均价 x ，反推原价。答案必须精确到分。

数据范围与算法目标

数据很大，不能用浮点；把金额全部转成“分”做整数运算。

Problem H. 周二会员日

推导

设实际原价为 P ，宣传语中的“折合单堡”是 $(P+1)/2$ 。

柠檬把它当成原价，记作 Q ，于是 $Q = (P+1)/2$ 。

他认为活动后均价为 $x = (Q+1)/2$ ，联立得 $P = 4x - 3$ 。

$$price = 4x - 3.$$

为避免歧义，把所有金额都换算成“分”， x 和 $price$ 都用整数表示。

精度

读入形如 $a.bb$ 的字符串，或用 `scanf("%llu.%llu")` 分别读整数部分和小数部分。

令 $x = a \times 100 + b$ ，避免任何浮点误差。

输出时再除以 100，补齐两位小数。

Problem M. 六个核桃：题目大意

题目大意

构造排列 p ，最大化 $\sum_i \min(|p_i - i|, m)$ 。

数据范围与算法目标

时间复杂度 $O(n)$ 。按 $m < n/2$ 与 $m \geq n/2$ 分类构造，使每个位置尽量达到截断上界。

Problem M. 六个核桃

构造上界

p_i 表示第 i 个位置放的编号。

每一项的贡献最多为 m 。若能让所有位置满足 $|p_i - i| \geq m$ ，就达到上界 $n \cdot m$ 。

若无法全部达到 m ，最优形态可以看作把小的数统一作为减数、大的数统一作为被减数。

拆开绝对值后，让小数尽量出现在右侧、大数尽量出现在左侧，以接近上界。

$2m < n$

做长度为 n 的循环位移：

$$p_i = \begin{cases} i + m, & 1 \leq i \leq n - m, \\ i - (n - m), & n - m < i \leq n. \end{cases}$$

每项都能取到 m ，所以达到 $n \cdot m$ 。

Problem M. 六个核桃：大 m 情况

$$2m \geq n$$

令 $h = \lfloor n/2 \rfloor$ ，把前半段和后半段整体交换：

$$p_i = \begin{cases} h + i, & 1 \leq i \leq n - h, \\ i - (n - h), & n - h < i \leq n. \end{cases}$$

当 n 为偶数时，这就是左右两半互换；当 n 为奇数时，按同样的循环位移处理即可。此时每个位置的位移尽量接近 $n/2$ ，已无法再提高。

证明思路

拆开绝对值后，小编号尽量出现在右侧、大编号尽量出现在左侧。这样可让小数作为减数、大数作为被减数，达到该情况下的构造上界。

Problem M. 六个核桃：上界证明

$2m < n$ 的上界

每一项都不超过 m ，因此总分不超过 nm 。

循环位移构造让每个位置的距离至少为 m ，于是每一项都取满 m 。

$2m \geq n$ 的上界

此时构造中的距离不会超过 m ，截断不再降低这些项。

最大化 $\sum |p_i - i|$ 时，应让小编号尽量配到大位置，大编号尽量配到小位置。

交换前后两段正好实现这个极端配对，因此达到上界。

Problem M. 六个核桃：实现

输出排列

若 $n = 1$ ，直接输出 1。

若 $2m < n$ ，按偏移量 m 输出 $m + 1, m + 2, \dots, n, 1, 2, \dots, m$ 。

否则令 $h = \lfloor n/2 \rfloor$ ，输出 $h + 1, h + 2, \dots, n, 1, 2, \dots, h$ 。

全程只顺序输出，不需要额外数组。

Problem C. 圆桌会议：题目大意

题目大意

构造 1 到 n 的环形排列，使每个连续三人组的中位数种类数尽量少。

数据范围与算法目标

所有 n 总和约为 10^6 ，构造需要线性输出。

Problem C. 圆桌会议

构造

令 $k = \lfloor n/3 \rfloor$ ，把数分为 $[1, k]$ 、 $[k+1, 2k]$ 、 $[2k+1, 3k]$ 和剩余的 0 到 2 个数。
前三组轮流放置，末尾接上剩余组：

$$1, k+1, 2k+1, 2, k+2, 2k+2, \dots, k, 2k, 3k, 3k+1, 3k+2.$$

轮流放置保证相邻三人通常来自“低、中、高”三段各一人。

解释

当三人来自低/中/高三段时，中位数必在中段，因此中位数种类被压缩到 k 个左右。
只在拼接剩余段附近会产生少量额外中位数，数量是常数级。
下一页会从三元窗口数量说明它达到下界。

环形意味着首尾也相邻，因此把剩余段放到末尾能减少新的中位数组合。

Problem C. 圆桌会议：数量说明

构造得到多少种中位数

当 $n = 3k$ 时，所有连续三元组的中位数只会落在 $k + 1 \sim 2k$ ，恰好 k 种。

当 $n = 3k + 1$ 时，末尾多一个数，只会在首尾拼接处多引入 1 种。

当 $n = 3k + 2$ 时，末尾多两个数，只会在尾部多引入 2 种。

所以构造的答案种类数为 $k, k + 1, k + 2$ 。

最优性来源

中位数集合必须命中每个长度为 3 的环形窗口；否则该窗口会产生新的中位数。

按 $n \bmod 3$ 分类检查首尾边界窗口，可得到下界分别为 $k, k + 1, k + 2$ 。

上面的构造恰好达到这个下界，并且可以按公式线性输出。

Problem B. 最大数码 2: 题目大意

题目大意

给定区间 $[l, r]$, 求 $f(x)$ 的最大值, 并在达到最大值的 x 中输出十进制字符串字典序最小者。

数据范围与算法目标

测试组数高, 但 $r \leq 10^9$ 。每组只需要检查常数个候选, 或在长度小于 10 时直接枚举。

Problem B. 最大数码 2

关键观察

当区间长度小于 10，直接枚举所有 x 。

当区间长度至少为 10，个位可以覆盖 0 到 9，所以最大数码一定是 9。

于是问题变成：在 $[l, r]$ 中找字典序最小、且包含数码 9 的 x 。

候选处理

同位数整数的字典序与数值顺序一致，若 l, r 位数相同，只需找区间内最小的含 9 数。

若区间跨越位数，字典序更小的候选可能来自更高位数的形态，如 $100\dots009$ 。需要额外判断它是否落在区间内。

实际实现只需要检查少量候选即可。

常见做法是先看 l 是否含 9，否则构造同位数的最小含 9 候选，再与跨位数候选比较。

Problem B. 最大数码 2: 字典序细节

为什么不能只取数值最小

同位数时，十进制字符串字典序与数值大小一致。

跨位数时二者不同，例如 100000009 的字典序可能小于很多短串。

因此要在所有“含 9 且落在区间内”的数字字符串中取字典序最小。

稳定实现

枚举长度 l ，只考虑与区间 $[l, r]$ 有交的 l 位数。

对固定长度，从高位到低位贪心填最小可行数字，并维护“后缀还能否落在 $[l, r]$ 且最终含 9”。

位数最多 10，也可以直接生成每个位置第一次放 9 的常数个候选。

Problem J. 秋水横渡：题目大意

题目大意

有 n 座周期性开闭关口，每座关口相位可预设，但全局随机偏移未知。求最小期望总耗时。

数据范围与算法目标

把所有后续关口的相位按航行时间递推设定，使第 2 到第 n 座关口无需等待。答案为航行时间总和加 $k/4$ 。

Problem J. 秋水横渡

相位设定

只要 i 号关口的相位 S_i 已经确定, 就令

$$S_{i+1} = (S_i + t_{i+1}) \bmod 2k.$$

这样从第一个关口通过后, 后续到达每座关口时都会落在同一开闭状态偏移上。

因此第 $2 \dots n$ 座关口的等待时间都可以消为 0。

Problem J. 秋水横渡：期望

只剩首关

第一座关口面对未知全局偏移，无法再通过相位选择改变均匀分布。

把全局偏移视为在 $[0, 2k)$ 的均匀随机变量。

单个周期中关闭区间长度为 k ，关闭时等待从 k 线性降到 0。

$$\frac{1}{2k} \int_0^k x dx = \frac{k}{4}.$$

最终答案为 $\sum_{i=2}^n t_i + k/4$ 。

Problem J. 秋水横渡：完整性说明

为什么这是最优

第一座关口一定会遇到未知全局偏移；无论相位怎样设置，看到的仍然是均匀分布。

因此首关的期望等待至少为 $k/4$ 。

前面的递推构造已经把第 2 到第 n 座关口等待降到 0，所以这个下界可以达到。

输出形式

总航行时间是确定量，等待只贡献 $\frac{k}{4}$ 。

题面接受绝对或相对误差，直接用 `long double` 计算并输出足够多位小数即可。

Problem E. 乐乐的杏花：题目大意

题目大意

树上点动态开关，当前位置沿边移动；每个时刻查询离当前位置最近的盛开杏花距离。

数据范围与算法目标

以 1 为根做 DFS 序，用线段树维护所有点到当前位置的距离；点开关和移动后查询全局最小值。

Problem E. 乐乐的杏花

DFS 序

以 1 为根 DFS, 记录 dfn_x 、 $dfnEnd_x$ 和深度 dep_x 。

一个点的子树在 DFS 序上是一段连续区间 $[dfn_x, dfnEnd_x]$ 。

线段树的每个叶子对应一个原树节点。

线段树含义

x_2 表示该节点到当前位置 pre 的距离。初始 $pre = 1$, 所以 $x_2 = dep$ 。

x_1 表示若该点盛开则等于 x_2 , 否则为一个很大的数。

根节点维护所有叶子的 x_1 最小值, 就是当前最近盛开点距离。

Problem E. 乐乐的杏花：查询

向父亲移动

若从 pre 移到父亲 $x = fa_{pre}$ ，对 pre 子树外的点距离都减 1，对子树内的点距离都加 1。
代码中先对全体加 -1 ，再对区间 $[dfn_{pre}, dfnEnd_{pre}]$ 加 2。

向儿子移动

若从 pre 移到儿子 x ，对 x 子树内的点距离都减 1，对子树外的点距离都加 1。
代码中先对全体加 1，再对区间 $[dfn_x, dfnEnd_x]$ 加 -2 。
区间加同时作用于 x_1, x_2 ，懒标记维护即可。

Problem E. 乐乐的杏花：维护细节

开关

若点 x 从未盛开变为盛开，在线段树叶子 dfn_x 令 $x_1 = x_2$ 。

若点 x 从盛开变为未盛开，在线段树叶子 dfn_x 令 $x_1 = INF$ 。

数组 w_x 记录当前开关状态。

输出

每次操作后查看线段树根的 x_1 。

若 x_1 仍然大于所有可能真实距离，说明没有盛开点，输出 2147483648。

否则输出 x_1 。每次操作复杂度 $O(\log n)$ 。

Problem A. 卡卡杀：题目大意

题目大意

树上有不超过 3000 个节点和 3000 名玩家。玩家按最短路逐时刻移动；同点玩家按阵营和冷却规则互相击杀，最后输出存活玩家。

数据范围与算法目标

先预处理每名玩家的路径，再按时刻模拟。每个时刻把玩家按“所在点、阵营”分桶。

Problem A. 卡卡杀

路径预处理

任选 1 为根，DFS 得到父亲和深度。

对玩家 u 的起点 s_u 、终点 t_u ，沿父亲指针求两点路径，存入数组 lj_u 。

时刻 x 若 $x < |lj_u|$ ，玩家位于 $lj_u[x]$ ；若路径已经结束，std 中不再放入本时刻桶。

状态

$zt_u = -1$ 表示死亡；否则表示本时刻冷却是否就绪。

若 $x \bmod tim_u = 0$ ，则本时刻 $zt_u = 1$ ，可以触发击杀。

Problem A. 卡卡杀：实现细节

分桶

在每个时刻 x ，建立 $h[pos][camp]$ ，保存当前位置为 pos 、阵营为 $camp$ 的玩家编号。
std 中阵营编号减一后使用，因此卡卡在 $camp = 0$ ，其他阵营为 1, 2, 3。
死亡玩家和已经走完路径的玩家不进入桶。

非卡卡击杀

枚举每个点，先处理阵营 1, 2, 3 中冷却就绪的玩家。
阵营 1 击杀同点卡卡；阵营 2 击杀同点阵营 1；阵营 3 击杀同点所有其他非卡卡阵营。
被击杀者先加入临时数组，当前点规则处理完后统一标记死亡。

Problem A. 卡卡杀：回合结算

卡卡击杀

若卡卡本人本时刻冷却就绪，则在每个点检查是否有卡卡同点玩家。

std 会把该点所有非卡卡玩家加入死亡列表。

这一段放在非卡卡击杀之后，与代码中的处理顺序一致。

复杂度控制

时刻只需模拟 $1 \sim n$ ，因为树上简单路径长度最多为 n 。

每个时刻遍历玩家入桶，再遍历所有点和阵营执行规则。

最后统计所有 $z_{t_i} \geq 0$ 的玩家并输出其编号。

Problem L. 猪猪车：题目大意

题目大意

交互猜四位数。一次询问返回位置和数码完全相同的位数，要求 12 次询问内猜出答案。

数据范围与算法目标

前 9 次确定数字多重集；std 随后把多重集归一化到几种类型，并用离线生成的固定决策树表完成后三问。

Problem L. 猪猪车

前九问确定多重集

依次询问 0000, 1111, ..., 8888, 返回值就是该数字在答案中的出现次数。

若某次返回 4, 答案已经确定, 直接提交。

未出现的数字位置最后补成 9, 因此可以恢复答案数字的多重集。

归一化

std 把多重集映射成类型, 复用归一化后的决策树, 无需为所有具体数字分别存树。

例如四个互异、两两相同、一组三个相同、一组两个相同等情况, 分别从不同根节点开始。

数组 `ans[0..3]` 保存参与决策树的代表数字。

Problem L. 猪猪车：固定决策树

三张表

$a[now][0..3]$ 表示当前节点要询问的四个代表数字下标。

$b[now][ret]$ 表示询问返回 ret 后跳到哪个节点。

若跳到的编号不少于 25，说明已经到叶子，答案排列在 $c[now-25]$ 中。

树的来源

这棵树可以由 minimax 离线搜索生成，再把结果硬编码到 a, b, c 。

比赛程序在线阶段只查表，不再搜索。

Problem L. 猪猪车：交互流程

在线执行

根据多重集类型设置初始节点 now ，并调整 ans 中代表数字的顺序。

循环询问 $ans[a[now][0..3]]$ ，读入返回值 ret ，令 $now=b[now][ret]$ 。

当 $now \geq 25$ 时，用 $c[now-25]$ 给出的排列提交真实数字。

次数

固定 9 次用于确定多重集，最多 3 次用于区分排列，总询问数不超过 12。

最终的 ! xxxx 不计入询问次数。

std 表中所有叶子都在三层内到达，因此覆盖所有归一化后的情况。

Problem I. 假想对冲：题目大意

题目大意

对每个子树，最大化初始部署无人机数。无人机同步向父亲移动，任一轮开始前节点无人机数不能超过出边容量。

数据范围与算法目标

树形 DP 的朴素复杂度是 $O(n^2)$ 。利用 DP 序列单调、阶梯压缩和长链剖分合并，可以做到总 $O(n)$ 。

Problem I. 假想对冲

树形 DP

定义 $f_{u,d}$: 在以 u 为根的子树中, 只在相对 u 深度为 d 的点上部署无人机时, 最终能通过边 (u, fa_u) 的最大数量。

$$f_{u,d} = \min \left(w_u, \sum_{v \in \text{son}_u} f_{v,d-1} \right), \quad f_{u,0} = w_u.$$

合并子树时, 相当于把子数组整体右移 1 后求和, 再与 w_u 取 \min 。

以 u 为根的答案是 $\sum_{d \geq 1} f_{u,d}$, 因为不能在子树根 u 本身部署无人机。

Problem I. 假想对冲：单调性

单调性

固定 u , $f_{u,d}$ 随 d 增加单调不增。更深处无人机向上汇聚时会经过更多边，瓶颈约束只会增加。

因此 $f_{u,*}$ 是“若干段常值”的阶梯序列。

Problem I. 假想对冲：线性化

阶梯压缩

用长链剖分给每个点分配一段连续下标 $[low_u, high_u]$ 。

下标 $low_u + d$ 存深度 d 的 DP 值，重儿子的数组天然接在 u 后面，可以原地复用。

单调序列压缩成若干段， $R[l]$ 表示当前段的右端开区间， $f[l]$ 表示该段值。

复杂度

长链剖分保证轻链合并总量为 $O(n)$ 。前缀削平每次都会减少段数，整棵树产生的新段总数也是 $O(n)$ ，所以均摊总复杂度为 $O(n)$ 。

Problem I. 假想对冲：实现结构

合并轻儿子

先处理重儿子，令 ans_u 继承重儿子的贡献。

每个轻儿子 v 处理完后，把 v 的段从 low_v 开始加到 $low_u + 1$ 开始的位置。

若两边当前段长度不同，就把较长的段切开，使两段长度相同后再相加。

答案

合并完所有儿子后，当前 ans_u 是未经过 u 这条出边限制的子树权值，先记为 res_u 。

随后把深度至少 1 的前缀中大于 w_u 的段削成 w_u ，并从 ans_u 中减掉被削去的量。

最后设置 $f[low_u] = w_u$ ，加入深度 0 这层，供父亲继续合并。

Problem D. 格兰法洛：题目大意

题目大意

每次操作选择一个功率 w_i ，从左或右清理满足覆盖关系且 $p_j < w_i$ 的区块。求让每个区块至少清理一次的功率方案数。

数据范围与算法目标

$n, W \leq 5000$ ，总 m 很大。std 做容斥和前缀/后缀 DP，复杂度 $O(nW \log m + n^2)$ 。

Problem D. 格兰法洛：先数补集

失败事件

令 A_i 表示第 i 个区块没有被清理。题目要数没有任何 A_i 发生的功率方案。
对失败事件容斥，设 $F(S)$ 表示让 S 中所有区块都保持未清理的方案数：

$$ans = W^m + \sum_{\emptyset \neq S \subseteq \{1, \dots, n\}} (-1)^{|S|} F(S).$$

单次操作的限制

一次操作经过区块 i 时，只有 $p_i < w$ 才会清理它。

如果这次操作经过了若干个 S 中的区块，为了让它们都不被清理，必须有 $w \leq \min p_i$ 。

所以一个操作只关心它覆盖到的已选区块中的最小浓度；没有覆盖到已选区块时有 W 种功率。

Problem D. 格兰法洛：按中心点分类

中心点 k

直接枚举非空集合 S 不可行。对每个 S ，取其中浓度最小的点；如果有多个最小值，取最左边的一个，记为 k 。

这样每个非空 S 都被唯一分到一个 k ，之后只需要枚举这个 k 。

经过 k 的操作

记 a_i 为满足 $op = 0, x = i$ 的操作数， b_i 为满足 $op = 1, x = i$ 的操作数。

$op = 0, x \geq k$ 和 $op = 1, x \leq k$ 的操作都会经过 k ，数量为

$$P_k = \sum_{i=k}^n a_i + \sum_{i=1}^k b_i.$$

因为 $k \in S$ ，这些操作都必须满足 $w \leq p_k$ 。每次有 p_k 种选择，所以贡献因子是 $p_k^{P_k}$ 。

Problem D. 格兰法洛：左右独立

剩余操作的形状

固定 k 后, $op = 0, x < k$ 的操作只会碰到左侧选点, $op = 1, x > k$ 的操作只会碰到右侧选点。两边不再共享操作, 因此可以把 S 拆成左侧部分、中心点 k 、右侧部分分别计数。

左右两边的浓度条件

左侧如果出现浓度 $\leq p_k$ 的选点, 中心点就会变成它, 所以左侧只能选浓度 $> p_k$ 的点。右侧可以选浓度等于 p_k 的点, 因为 k 更靠左, 中心点仍然是 k 。

Problem D. 格兰法洛：前缀 DP 含义

状态

沿用上一页的记号， a_i 为满足 $op = 0, x = i$ 的操作数。

$f_{i,j}$ 汇总所有 $T \subseteq [1, i]$ 的贡献，其中 j 是 T 内选点浓度的最小值， $j = W + 1$ 表示 T 为空。

每个集合 T 自带容斥符号 $(-1)^{|T|}$ ，并且已经计算了所有 $op = 0, x \leq i$ 的功率选择数。

为什么状态只要一个 j

扫描到位置 i 时，所有 $op = 0, x = i$ 的操作会经过当前前缀里被选的点。

这些操作的合法功率数只由当前最小浓度决定，也就是 $\min(j, W)$ 。

Problem D. 格兰法洛: DP 转移

加入位置 $i + 1$

$$f_{i+1,j} += f_{i,j} \min(j, W)^{a_{i+1}},$$

这是不选 $i + 1$, 最小浓度不变。

$$f_{i+1,\min(p_{i+1},j)} -= f_{i,j} \min(p_{i+1},j)^{a_{i+1}}.$$

这是选 $i + 1$, 最小浓度更新, 容斥符号翻转。

后缀 DP

$g_{i,j}$ 从右往左处理 $op = 1$ 操作, 含义完全对称, 初始 $f_{0,W+1} = g_{n+1,W+1} = 1$ 。

Problem D. 格兰法洛：合并公式

枚举中心 k

左侧取最小浓度 $> p_k$ 的状态，右侧取最小浓度 $\geq p_k$ 的状态：

$$c_1(k) = \sum_{j=p_k+1}^{W+1} f_{k-1,j}, \quad c_2(k) = \sum_{j=p_k}^{W+1} g_{k+1,j}.$$

扣掉这一类失败集合

$$ans = W^m - \sum_{k=1}^n c_1(k)c_2(k) \cdot p_k^{\sum_{i=k}^n a_i + \sum_{i=1}^k b_i}.$$

负号来自中心点 k 本身被选入失败集合，左右 DP 只统计了两侧的容斥符号。

Problem D. 格兰法洛：实现细节

预处理

统计每个位置的 a_i, b_i ，并预处理 v^0, v^1, \dots, v^{N-1} 。

quick_pow 把指数按 $N-1$ 分块： $v^e = (v^{N-1})^{\lfloor e/(N-1) \rfloor} \cdot v^{e \bmod (N-1)}$ 。

前缀 DP 从左到右，后缀 DP 从右到左，状态维度都是 $W+1$ 。

复杂度

两个 DP 都是 $O(nW)$ 。枚举 k 时要得到 P_k ，std 直接扫一遍统计，整体为 $O(nW \log m + n^2)$ 。

Problem K. 卡卡拆：题目大意

题目大意

求特殊整数拆分数 $p(0), p(1), \dots, p(N)$ ，每个正整数出现次数只能为 0 或奇数，模 998244353。

数据范围与算法目标

$N \leq 10^6$ ，用生成函数加 NTT 多项式指数，总复杂度 $O(N \log N)$ 。

Problem K. 卡卡拆

生成函数

$$P(x) = \sum_{n \geq 0} p(n)x^n = \prod_{i \geq 1} (1 + x^i + x^{3i} + x^{5i} + \dots) = \prod_{i \geq 1} \frac{1 + x^i - x^{2i}}{1 - x^{2i}}.$$

问题变成求 $P(x)$ 的前 $N+1$ 项系数。

对数展开

令 $A(x) = \ln P(x)$ 。先求 A_1, \dots, A_N ，再做一次多项式 \exp 得到 $P(x)$ 前 $N+1$ 项。
这样把连乘转化为求和，便于按约数筛求系数。

Problem K. 卡卡拆：符号准备

两个记号

$[x^n]Q(x)$ 表示形式幂级数 $Q(x)$ 中 x^n 的系数。

$\sigma(t)$ 表示 t 的所有正约数之和，也就是

$$\sigma(t) = \sum_{d|t} d.$$

Lucas 数列

记 $L_0 = 2, L_1 = 1$ ，并且 $L_m = L_{m-1} + L_{m-2}$ 。

后面用到的恒等式是

$$\ln(1 + y - y^2) = \sum_{m \geq 1} (-1)^{m+1} \frac{L_m}{m} y^m.$$

Problem K. 卡卡拆：偶数分母部分

偶数分母部分

$$-\ln(1 - x^{2i}) = \sum_{k \geq 1} x^{2ik} / k.$$

若要求 x^{2t} 的系数，需要满足 $2ik = 2t$ ，也就是 $ik = t$ 。

枚举约数 $i | t$ ，此时 $k = t/i$ ，所以贡献为

$$\sum_{i|t} \frac{1}{t/i} = \frac{\sigma(t)}{t}.$$

奇数次数没有贡献

这部分只有 x^{2ik} ，所以 n 为奇数时贡献是 0。

Problem K. 卡卡拆: Lucas 部分

单个因子的对数

对分子项使用上一页的恒等式, 代入 $y = x^i$:

$$\ln(1 + x^i - x^{2i}) = \sum_{m \geq 1} (-1)^{m+1} \frac{L_m}{m} x^{im}.$$

取 x^n 的系数

要产生 x^n , 必须有 $im = n$ 。枚举 $m \mid n$, 对应的 $i = n/m$ 唯一。

因此分子部分对 A_n 的贡献为

$$\sum_{m \mid n} (-1)^{m+1} \frac{L_m}{m}.$$

Problem K. 卡卡拆：实现要点

合并系数

令 $A(x) = \ln P(x) = \sum_{n \geq 1} A_n x^n$ 。由分子和分母两部分相加，得到

$$A_n = \sum_{d|n} (-1)^{d+1} \frac{L_d}{d} + \begin{cases} \frac{\sigma(n/2)}{n/2}, & 2 \mid n, \\ 0, & 2 \nmid n. \end{cases}$$

实现要点

用除数筛分别计算 $\sigma(t)$ 与 $\sum_{d|n} (-1)^{d+1} L_d/d$ 。

预处理逆元以处理除法，最后用 NTT+ 牛顿迭代做多项式 exp。

求出 $A_0 = 0, A_1, \dots, A_N$ 后，计算 $P(x) = \exp(A(x))$ ，输出 P 的 0 到 N 次项系数。

致谢

命题人

拱垲：组长，G、K 出题人。
袁英麒：I 出题人。
栗铭远：M 出题人。
赵星宇：B、C、H 出题人。
张健哲、韩子睿：D、J 出题人。
席乐媛：L 出题人。
邱天：A、E 出题人。

验题人

吉林大学：于之航、陈鹏旭、徐梓博。
四川大学：hungry、林宸。
西南财经大学：吴晗、韩宇轩。
吉林大学：钟姝伶、韩洋、庄锦皓。
西北工业大学：邵嘉豪、王彦淞、郭知宇。
四川大学：王鹏超。
杭州电子科技大学：黄光升。

感谢以上命题人，以及参与验题工作的各支验题队伍。