骊山 SPbSU 锦标赛

2024 年 5 月 12 日

问题 A：元素比较

创意 开发： Dmitry Belichenko 德米特里-别利琴科

尼基塔-加沃

伊 编辑　　伊万-博奇科夫

元素比较

- 给定一个置换 p，我们需要找到若干对给定长度的子数组，使得左边的子数组是在元素上比右边的小。

元素比较

- 给定一个置换 p，我们需要找到若干对给定长度的子数组，使得左边的子数组是在元素上比右边的小。
- 考虑布尔矩阵 $C_{\ell,s}$，如果 $p_\ell < p_{\ell+s}$，则 $C_{\ell,s} = 1$。

元素比较

- 给定一个置换 p，我们需要找到若干对给定长度的子数组，使得左边的子数组是在元素上比右边的小。
- 考虑布尔矩阵 $C_{\ell,s}$，如果 $p_\ell < p_{\ell+s}$，则 $C_{\ell,s} = 1$。
- 如何做到这一点？比特集！(或实用程序，它们也有帮助)

元素比较

- 给定一个置换 p，我们需要找到若干对给定长度的子数组，使得左边的子数组是在元素上比右边的小。
- 考虑布尔矩阵 $C_{\ell,s}$，如果 $p_\ell < p_{\ell+s}$，则 $C_{\ell,s} = 1$。
- 如何做到这一点？比特集！(或实用程序，它们也有帮助)

- 我们可以在 $O(n^2/w)$ 的时间内明确构建这个矩阵。
- 我们可以按 m 行选择枢轴元素，然后计算枢轴元素之间的前缀-OR 和后缀-OR，从而计算出答案。这部分也需要 $O(n^2/w)$ 时间。

问题 B：女学生

创意： Mikhail Ivanov 开发：米哈伊尔-伊万诺夫 米哈伊尔-伊万诺

夫 编辑：　米哈伊尔-伊万诺

夫

女学生

- 我们给出了平面上正多边形的顶点

女学生

- 我们给出了平面上正多边形的顶点 我们可以将三角形扩展成
- 平行四边形，为我们的集合增加一个新点

女学生

- 我们给出了平面上正多边形的顶点 我们可以将三角形扩展成
- 平行四边形，为我们的集合增加一个新点
- 经过多次这样的操作后，检查顶点来自我们集合的多边形的规则性

女学生

- 每个顶点可以与一个向量 $v \in R^2$

女学生

- 每个顶点可以与一个向量 $v \in R^2$
- 顶点 A、B、C 的运算结果是 $D = A + C - B$

女学生

- 每个顶点可以与一个向量 $v \in R^2$
- 顶点 A、B、C 的运算结果是 $D = A + C - B$
- 如何检查 $A_1, \ldots, A_n$ 是正则表达式?

女学生

- 每个顶点可以与一个向量 $v \in \mathbb{R}^2$
- 顶点 A、B、C 的运算结果是 $D = A + C - B$
- 如何检查 $A_1, \ldots, A_n$ 是正则表达式?
- 首先，让 $M = \frac{A_1+\ldots+A_n}{n}$, $A_i' = A_i - M$

女学生

- 每个顶点都可以与一个向量 $v \in R^2$
- 顶点 A、B、C 的运算结果是 $D = A + C - B$
- 如何检查 $A_1, \ldots, A_n$ 是正则表达式?
- 首先，让 $M = \frac{A_1 + \ldots + A_n}{n}$，$A'_i = A_i - M$
- 检查 $A'_1, \ldots, A'_n$ 是正则表达式，中心为零

女学生

- 每个顶点可以与一个向量 $v \in \mathbb{R}^2$
- 顶点 A、B、C 的运算结果是 $D = A + C - B$
- 如何检查 $A_1, \ldots, A_n$ 是正则表达式?
- 首先，让 $M = \frac{A_1 + \ldots + A_n}{n}$, $A'_i = A_i - M$
- 检查 $A'_1, \ldots, A'_n$ 是正则表达式，中心为零
  - 将每个矢量旋转 $\frac{2\pi}{n}$ 并检查旋转后的矢量是否在设置

女学生

- 每个顶点可以与一个向量 $v \in R^2$
- 顶点 A、B、C 的运算结果是 $D = A + C - B$
- 如何检查 $A_1, \ldots, A_n$ 是正则表达式?
- 首先，让 $M = \frac{A_1 + \ldots + A_n}{n}$, $A'_i = A_i - M$
- 检查 $A'_1, \ldots, A'_n$ 是正则表达式，中心为零
  - 将每个矢量旋转 $\frac{2\pi}{n}$ 并检查旋转后的矢量是否在设置

而不是

A_i'

女学生

- $\mathbb{R}^2$ 是 $\mathbb{R}$ 上的二维向量空间

女学生

- $\mathbb{R}^2$ 是 $\mathbb{R}$ 上的二维向量空间
- 存储每个点的两个实数坐标，相加并检查相应的正则性

女学生

- R^2 是 R 上的二维向量空间
- 存储每个点的两个实数坐标，相加并检查相应的正则性
- 然而，现代计算机无法存储 R

女学生

- R^2 是 R 上的二维向量空间
- 存储每个点的两个实数坐标，相加并检查相应的正则性
- 然而，现代计算机无法存储 R
- 精度误差

女学生

- $\mathbb{R}^2$ 是 $\mathbb{R}$ 上的二维向量空间
- 存储每个点的两个实数坐标，相加并检查相应的正则性
- 然而，现代计算机无法存储 $\mathbb{R}$
- 精度误差
- 对于任意 $n \neq 4$，我们可以构造一个指数趋向于某一点但永远

不会到达该点的序列

女学生

- $\mathbb{R}^2$ 是 $\mathbb{R}$ 上的二维向量空间
- 存储每个点的两个实数坐标，相加并检查相应的正则性
- 然而，现代计算机无法存储 $\mathbb{R}$
- 精度误差
- 对于任意 $n \neq 4$，我们可以构造一个指数趋向于某一点但永远

不会到达该点的序列

- 因此，任何有限精度都是不够的

女学生

- 相反，让我们在 Q 上的向量空间中工作

女学生

- 相反，让我们在 Q 上的向量空间中工作
- 正则 n 冈生成维数为$\varphi(n)$ 的向量空间

女学生

- 相反，让我们在 Q 上的向量空间中工作
- 正则 n 冈生成维数为$\varphi(n)$ 的向量空间
  - 要更好地理解这个向量空间，请参阅 MemSQL Start[c]UP 3.0 第 1 轮问题 G：数圆

女学生

- 相反，让我们在 Q 上的向量空间中工作
- 正则 n 冈生成维数为 $\varphi(n)$ 的向量空间
  - 要更好地理解这个向量空间，请参阅 MemSQL Start[c]UP 3.0 第 1 轮问题 G：数圆
- 现在我们存储 $\varphi(n)$ 个整数

女学生

- 相反，让我们在 Q 上的向量空间中工作
- 正则 n 冈生成维数为$\varphi(n)$ 的向量空间
  - 要更好地理解这个向量空间，请参阅 MemSQL Start[c]UP 3.0 第 1 轮问题 G：数圆
- 现在我们存储 $\varphi(n)$ 个整数

- 从初始多边形的顶点中选择基点

女学生

- 相反，让我们在 Q 上的向量空间中工作
- 正则 n 冈生成维数为$\varphi(n)$ 的向量空间
  - 要更好地理解这个向量空间，请参阅 MemSQL Start[c]UP 3.0 第 1 轮问题 G：数圆
- 现在我们存储 $\varphi(n)$ 个整数

- 将矢量旋转 $\frac{2\pi}{n}$ 弧度，用

  多边形中下一个顶点的表示方法

女学生

- 为了避免对很长的整数进行运算，可以随机选择一个质数，然后计算这个质数的模数和

女学生

- 为了避免对很长的整数进行运算，可以随机选择一个质数，然后计算这个质数的模数和
- 渐近学

女学生

- 为了避免对很长的整数进行运算，可以随机选择一个质数，然后计算这个质数的模数和
- 渐近学
  - 每个平行四边形 $O(\varphi(n))$

女学生

- 为了避免对很长的整数进行运算，可以随机选择一个质数，然后计算这个质数的模数和
- 渐近学
  - 每个平行四边形 $O(\varphi(n))$
  - $O(\varphi(n)^2$) 每一个 $\frac{2\pi}{n}$ 自转

女学生

- 为了避免对很长的整数进行运算，可以随机选择一个质数，然后计算这个质数的模数和
- 渐近学
  - 每个平行四边形 $O(\varphi(n))$
  - $O(\varphi(n)^2)$ 每一个 $\frac{2\pi}{n}$ 自转
  - 每次多边形检查 $O(n\varphi(n))^2$

女学生

- 为了避免对很长的整数进行运算，可以随机选择一个质数，然后计算这个质数的模数和
- 渐近学
  - 每个平行四边形 $O(\varphi(n))$
  - $O(\varphi(n)^2)$ 每一个 $\frac{2\pi}{n}$ 自转
  - 每次多边形检查 $O(n\varphi(n))^2$

- 速度仍然很慢

女学生

- 对于某个素数 p，让我们将这一结构嵌入 $\mathbb{F}_p$

女学生

- 对于某个素数 p，让我们将这一结构嵌入 $\mathbb{F}_p$
- 为此，n 应除以 $p - 1$

女学生

- 对于某个素数 p，让我们将这一结构嵌入 $\mathbb{F}_p$
- 为此，n 应除以 $p - 1$
- 对 $kn + 1$ 形式的大数进行迭代，检查其原始性，找到其生成根，并取其 k^{th} 的幂 g

女学生

- 对于某个素数 p，让我们将这一结构嵌入 $\mathbb{F}_p$
- 为此，n 应除以 $p - 1$
- 对 $kn + 1$ 形式的大数进行迭代，检查其原始性，找到其生成根，并取其 k^{th} 的幂 g
- 绕零旋转为 $x \mapsto gx$，多边形为

$$g,\ g^2,\ .\ .\ .,\ g^{n-1},\ g^n = 1$$

女学生

- 那么 $m \neq n$ 的多边形呢?

女学生

- 那么 $m \nmid n$ 的*多边形*呢?
- 如果 m 不能整除 $\mathrm{lcm}(n,2)$，那么它肯定不是正则数

女学生

- 那么 $m \not\models n$ 的*多边形*呢?
- 如果 m 不能整除 lcm(n,2)，那么它肯定不是正则数
- 如果 n 为奇数，则可以构造正则 2n 形，因此让我们从 $2n$ 开始

女学生

- 那么 $m \neq n$ 的*多边形*呢?
- 如果 m 不能整除 lcm(n,2)，那么它肯定不是正则数
- 如果 n 为奇数，则可以构造正则 2n 形，因此让我们从 $2n$ 开始

- 如果 m 除以 n，则旋转 $\frac{2\pi}{m}$ 为 $x' \to g x^{n/m}$

问题 C：挑选樱桃

创意 开发：

Anton Maidel 安东-迈德尔

编辑： 米哈伊尔-伊万诺夫

采摘樱桃

- 你下了 n 盘棋
- 您会得到对手的 n 个国际象棋等级分
- 您还可以知道每场比赛的胜负情况
- 求最大值 x，使得在与等级$\geq x$ 的棋手的对局中，有 k 场连胜

采摘樱桃

- 注意，二进制搜索是不可能的：没有单调性

采摘樱桃

- 请注意，二进制搜索是不可能的：没有单调性 有两
- 种解决方案：使用段树状数据结构和 DSU

采摘樱桃

- 数据结构

采摘樱桃

- 数据结构
  - 未选中游戏 = 0
  - 选中胜利 = 1
  - 失败 = -∞

采摘樱桃

- 数据结构
  - 未选中游戏 = 0
  - 选中胜利 = 1
  - 失败 = -∞
- 使子阵列上的总和最大化的标准分而治之法

采摘樱桃

- DSU：

采摘樱桃

- DSU：
- 逐渐增加 x

采摘樱桃

- DSU：
- 逐渐增加 x
- 起初，我们在失败之间有许多分段

采摘樱桃

- DSU：
- 逐渐增加 x
- 起初，我们在失败之间有许多段落，会发生什么
- 呢？
  - 随着失败的消失，两个区段合并

采摘樱桃

- DSU：
- 逐渐增加 x
- 起初，我们在失败之间有许多段落，会发生什么
- 呢？
  - 随着一次失败的消失，两个片段合并
  - 随着一

次胜利的消失，一个价值单位也随之消失

采摘樱桃

- DSU：
- 逐渐增加 x
- 起初，我们在失败之间有许多段落，会发生什么
- 呢？
  - 随着一次失败的消失，两个片段合并
  - 随着一

次胜利的消失，一个价值单位也随之消失

- 我们可以使用 std::整数对集合来代替 DSU
  - (失败的瞬间，它斩断的胜利之链)

问题 D：小矮人的就寝时间

创意： Ivan

Kazmenko Ivan Kazmenko

开发： 伊万-卡兹缅科 编辑

： 伊万-卡兹缅科

小矮人的就寝时间

- 白雪公主和小矮人住在房子里
- 每个小矮人每天连续睡眠 12 小时（周期性），每天也连续清醒 12 小时
- 我们有一天时间提问，从 00:00 到 23:59
- 对于每个小矮人，我们可以交互式地询问他是睡着了还是醒着

，最多 50 次

■为每个小矮人找出他入睡的准确时间■ 扭曲：我们无法回到过去提问

小矮人的就寝时间

- 小矮人是独立的，让我们解决一个小矮人的问题 约束条件允
- 许我们在 00:00 到 23:59 的每一分钟，检查我们是否对每个小矮人都提出了问题

小矮人的就寝时间

给每个侏儒

- 首先，在 00:00 提问
- 如果小矮人是醒着的，他会转为睡着，然后再转为醒着
- 如果小矮人是睡着的，他会转为醒着，然后再转为睡着
- 解法是

对称的：只需与小矮人睡着时的状态进行比较即可。

00:00，如有需要，可在最后增加 12 小时

小矮人的就寝时间

给每个侏儒

- 如果我们能回到过去呢?
- 二进制搜索：12 - 60 = 720 分钟意味着多 10 个问题

小矮人的就寝时间

给每个侏儒

- 我们可以寻找哪些关键时刻?
  - 1. 小矮人从 00:01 到 12:00 改变状态
  - 2. 小矮人从 12:01 到 24:00 会改变状态
- 但我们不能在 24:00 询问，所以请注意最后一分钟
- （如何：

如果我们没有找到答案，那么答案就是 00:00）。

- 思路：近似求出第一时刻，然后精确求出第二时刻

小矮人的就寝时间

给每个侏儒

- 平方根法：将 729 > 720 分钟分成 27 段，每段 27 分钟
- 至 12:00，在每节开始时询问
- 直到 24:00，在相应部分的每一分钟提问
- 1 + 27 + 27 =

55 > 50，有点不够

小矮人的就寝时间

给每个侏儒

- 改进的平方根方法：将 741 > 720 分钟分成 38 段，每段 38、37、36、., 3, 2, 1 分钟
- 至 12:00，在每节开始时询问
- 在 24:00 之前，在相应部分的每一分钟提问■ 如果我们进入

了 k 部分，则有 $39 - k$ 分钟可以提问■ 问题总数将是 $1 +$

$39 = 40 < 50$、

这就够了

问题 E：假硬币和假天平

创意： Ivan

Bochkov Ivan Bochkov 开

发： 伊万-博奇科夫 编

辑：　　　伊万-博奇科夫

F ake Coin and Lying Scales

- 我们有 n 枚硬币和两个平底锅刻度，它们可能最多位于 k 次。一枚硬币是假的，比其他硬币重。

假硬币和说谎的天平

- 我们有 n 枚硬币和两个平底锅刻度，它们可能最多位于 k 次。有一枚硬币是假的，比其他硬币重。我
- 们需要找到那枚假币。

假硬币和说谎的天平

- 我们有 n 枚硬币和两个平底锅刻度，它们可能最多位于 k 次。有一枚硬币是假的，比其他硬币重。我
- 们需要找到那枚假币。
- 我们最多可能猜错 3 千次。

F ake Coin and Lying Scales

- 我们有 n 枚硬币和两个平底锅刻度，它们可能最多位于 k 次。有一枚硬币是假的，比其他硬币重。我
- 们需要找到那枚假币。
- 我们最多可能猜错 3 千次。
- 我们的目标是找到最大可能的 n，使其在一定精度（对数刻度

10）下是可行的。

F ake Coin and Lying Scales

- 假设我们进行了一些称重。对于任何一枚硬币，我们都有哪些信息?

假硬币和说谎的天平

- 假设我们进行了一些称重。对于任何一枚硬币，我们都有哪些信息？
- 最多只有一个数字 k：如果这枚硬币是假的，天平欺骗我们的次数。

假硬币和说谎的天平

- 假设我们进行了一些称重。对于任何一枚硬币，我们都有哪些信息？
- 最多只有一个数字 k：如果这枚硬币是假的，天平欺骗我们的次数。
- 定义潜能 $p(\ell, v)$：如果硬币是假的，我们最多可以称 ℓ 次，最多可以说谎 v 次，则该硬币的潜能。

假硬币和说谎的天平

- 假设我们进行了一些称重。对于任何一枚硬币，我们都有哪些信息？
- 最多只有一个数字 k：如果这枚硬币是假的，天平欺骗我们的次数。
- 定义潜能 $p(\ell, v)$：如果硬币是假的，我们最多可以称重 ℓ 次，最多可以说谎 v 次，则该硬币的潜能。

- p 的取值方式为 $p(0, 0) = 1$ 和 $p(\ell, v) = 2p(\ell - 1, v - 1) + p(\ell - 1, v)$.

F ake Coin and Lying Scales

- 假设我们进行了一些称重。对于任何一枚硬币，我们都有哪些信息?
- 最多只有一个数字 k：如果这枚硬币是假的，天平欺骗我们的次数。
- 定义潜能 $p(\ell, v)$：如果硬币是假的，我们最多可以称 ℓ 次，最多可以说谎 v 次，则该硬币的潜能。

- p 的取值方式为 $p(0, 0) = 1$ 和
  $p(\ell, v) = 2p(\ell - 1, v - 1) + p(\ell - 1, v)$.
- 一个状态的电位被定义为其所有硬币的电位之和。

F ake Coin and Lying Scales

- 有了这个势，我们可以注意到，3 个可能的权衡结果的势之和等于初始状态的势。

假硬币和说谎的天平

- 有了这个势，我们可以注意到，3 个可能的权衡结果的势之和等于初始状态的势。
- 所以 $n \le \frac{(3k+1)3^n}{p(n,k)}$.

假硬币和说谎的天平

- 有了这个势，我们可以注意到，3 个可能的权衡结果的势之和等于初始状态的势。
- 所以 $n \le \frac{(3k+1)3^n}{p(n,k)}$.
- 此外，我们还可以注意到，我们几乎可以在每一步上平分电势，因此这个近似值已经足够好了。

假硬币和说谎的天平

- 有了这个势，我们可以注意到，3 个可能的权衡结果的势之和等于初始状态的势。
- 所以 $n \le \frac{(3k+1)3^n}{p(n,k)}$.
- 此外，我们还可以注意到，我们几乎可以在每一步上平分电势，因此这个近似值已经足够好了。
- $p(n, k) = \sum_{j \le k} C_n^j 2^j$ 。

F ake Coin and Lying Scales

- 有了这个势，我们可以注意到，3 个可能的权衡结果的势之和等于初始状态的势。
- 所以 $n \le \frac{(3k+1)3^n}{p(n,k)}$.
- 此外，我们还可以注意到，我们几乎可以在每一步上平分电势，因此这个近似值已经足够好了。
- $p(n, k) = \sum_{j \le k} C_n^j 2^j$ 。
- 我们只需要对这个总和进行近似计算。这可以用

方法有很多种，最简单的可能是以下几种：

Fake Coin and Lying Scales

- 考虑最大和 m 而不是整个和。

假硬币和说谎的天平

- 考虑最大和 m 而不是整个和。那么 $\frac{1}{4k} p(n, k) \le m \le$
- $\sqrt{\frac{p(n, k)}{2}}$.

假硬币和说谎的天平

- 考虑最大和 m 而不是整个和。那么 $\frac{1}{4k^{\frac{1}{2}}} p(n,k) \le m \le p(n,k)$
- 因此，我们可以将 $m \cdot 2k^{\frac{1}{4}}$ 作为近似值。

假硬币和说谎的天平

- 考虑最大和 m 而不是整个和。那么 $\frac{1}{4k^{\frac{1}{2}}} p(n, k) \le m \le p(n, k)$。
- 因此，我们可以将 $m \cdot 2k^{\frac{1}{4}}$ 作为近似值。
- 精度为 $O(k^{\frac{1}{4}})$，这已经足够好了。

F ake Coin and Lying Scales

- 考虑最大和 m 而不是整个和。那么 $\frac{1}{4k^2} p(n, k) \le m \le p(n, k)$.
- 因此，我们可以将 $\frac{m}{\frac{1}{2k^{4}}}$ 作为近似值。
- 精度为 $O(k^4)$，这已经足够好了。
- 不过，精确度还是可以近似的。

问题 F：整个世界

创意： 米哈伊尔-伊万诺夫 Ivan Bochkov

开发部： 编辑： Ivan Bochkov

整个世界

- 如果多项式在所有整数点上的取值都是整数，那么它就是*整多项式*。

整个世界

- 如果多项式在所有整数点上的取值都是整数，那么它就是*整多项式*。
- 有一些点 (x_i, y_i) 的 $x_i \le 30$。

整个世界

- 如果一个多项式在所有整数点上都取整数值，那么它就是*整多项式*。
- 有一些点 (x_i, y_i) 的 $x_i \le 30$。
- 在这些点上取这些值的*全*多项式的最小阶数是多少?

整个世界

- 全多项式是二项式系数的线性组合。

整个世界

- 全多项式是二项式系数的线性组合。
- 因此，至少存在一个这样的全多项式。

整个世界

- 全多项式是二项式系数的线性组合。
- 因此，至少存在一个这样的全多项式。
- 首先，让我们忘记多项式必须是整数这一条件。

整个世界

- 全多项式是二项式系数的线性组合。
- 因此，至少存在一个这样的全多项式。
- 首先，让我们忘记多项式必须是整数这一条件。
- 那么，我们只需对给定的点进行内插，就可以得到一些多项式。

整个世界

- 如果它是完整的，我们就赢了。这个条件足以检查 1、2、., d，其中 d 是它的度数。

整个世界

- 如果它是完整的，我们就赢了。这个条件足以检查 1、2、., d，其中 d 是它的度数。
- 如果不是，我们可以注意到，所有分母的除数都只来自质数的小幂，直到 29。

整个世界

- 如果它是完整的，我们就赢了。这个条件足以检查 1、2、., d，其中 d 是它的度数。
- 如果不是，我们可以注意到，所有分母的除数都只来自质数的小幂，直到 29。
- 那么只需求解这些素数幂的模，并取最大值即可。

整个世界

- 如果它是完整的，我们就赢了。这个条件足以检查 1、2、., d，其中 d 是它的度数。
- 如果不是，我们可以注意到，所有分母的除数都只来自质数的小幂，直到 29。
- 那么只需求解这些素数幂的模，并取最大值即可。

- 我们可以根据度数进行二元搜索。如何检查给定阶数 d 的全多项式是否存在？

整个世界

- 如果我们取向量 $v_i = (C^i_{x_1}, \dots, C^i_{x_n})$，我们需要检查某个给定数是 v 的线性组合$_i$。

整个世界

- 如果我们取向量 $v_i = (C^i_{x_1}, \dots, C^i_{x_n})$，我们需要检查某个给定数是 v_i 的线性组合。
- 只需检查小素数的小幂模即可。

整个世界

- 如果我们取向量 $v_i = (C^i_{x_1}, \ldots, C^i_{x_n})$，我们需要检查某个给定数是 v 的线性组合$_i$。
- 只需对小素数的小幂进行求解即可。因此，我们需要求解一些线性系统的素数幂模，这可以通过接近高斯消元法的对角化过程来实现。
-

整个世界

- 如果我们取向量 $v_i = (C^i_{x_1}, \ldots, C^i_{x_n})$，我们需要检查某个给定数是 v_i 的线性组合。
- 只需对小素数的小幂进行求解即可。因此，我们需要求解一些线性系统的素数幂模，这可以通过接近高斯消元法的对角化过程来实现。
- 顺便说一句，你可以证明这里不需要第一部分的插值和检验

多项式。只需求解系统的素幂模即可。

整个世界

- 如果我们取向量 $v_i = (C^i_{x_1}, \ldots, C^i_{x_n})$，我们需要检查某个给定数是 v 的线性组合$_i$。
- 只需对小素数的小幂进行求解即可。因此，我们需要求解一
- 些线性系统的素数幂模，这可以通过接近高斯消元法的对角化过程来实现。
- 顺便说一句，你可以证明这里不需要第一部分的插值和检验

多项式。只需求解系统的素幂模即可。

- 奖励。用 $x_i \leq 10^9$ 求解。

问题 G：异常情况

构思： Sergey

Kopeliovich 谢尔盖-科佩利奥维

奇 开发： 谢尔盖-科佩利奥维

奇

编辑：　　米哈伊尔-伊万诺夫

不寻常的案例

- 给您一个随机无向图，图中有 n 个顶点和边缘
- 在给定图形中找出 k 条不相交的哈密顿路径
- $n = 10\,000$，$m = 200\,000$，$k = 8$

不寻常的案例

- 如何找到一条道路?

不寻常的案例

- 如何找到一条路径？

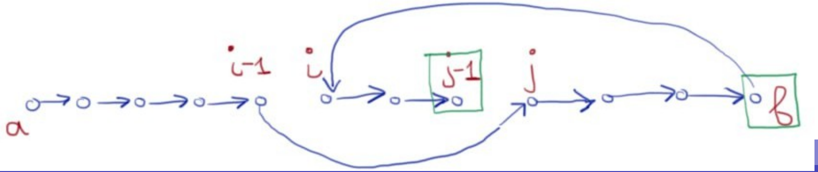
贪婪随机行走

不寻常的案例

- 如何找到一条路径？
-

贪婪随机行走

- 如果无处可去，就按图中的方法重建：



不寻常的案例

- 找到 k 条路径后，用同样的方法开始寻找路径 $k+1$

不寻常的案例

- 找到 k 条路径后，以同样的方法开始寻找路径 $k+1$ 如果
- 我们没有成功找到 8 条路径，则重新开始

不寻常的案例

- 2021 年，研究证明在 $O(n)$ 分钟内可以找到一条路径

不寻常的案例

- 2021 年，有研究证明，在 $O(n)$ 内可以找到一条路径 在移除
- 几条随机的汉密尔顿路径后，图仍然是相当随机的

问题 H：vdome.com 上的页面

创意：Mikhail Ivanov 开发：Anastasia Grigorieva 阿纳斯塔西娅-格里戈里耶娃

编辑： Anastasia Grigorieva 阿纳斯塔西娅-格里戈里耶娃

P age on vdome.com

- 写下从 1 到 N 的所有数字，每个数字倒着写。去掉所有前导
- 零。
- 找出最小排除数 (MEX)。

P age on vdome.com

- 几乎所有 N 的答案都是 10。

vdome.com 上的页面

- 几乎所有 N 的答案都是 10。
- 因为不存在在 "id "和第一个有效数字之间有 0 的页面地址。

vdome.com 上的页面

- 几乎所有 N 的答案都是 10。
- 因为不存在在 "id "和第一个有效数字之间放置 0 的页面地址。
- 因此，10 不可能存在于结果数集中。而 10 将是所有 $N \geqslant 10$ 的 MEX。

P age on vdome.com

- 几乎所有 N *的*答案都是 10。
- 因为不存在在 "id "和第一个有效数字之间放置 0 的页面地址。
- 因此，10 不可能存在于结果数集中。而 10 将是所有 $N \geqslant 10$ *的* MEX。
- $N < 10$ *的*答案是 $N + 1$。

问题 I：旋转和转动！

创意： Mikhail Ivanov　开发：米哈伊尔-伊万诺夫　米哈伊尔-伊万诺

夫 编辑： 米哈伊尔-伊万诺

夫

旋转

- 考虑两根绳子 AB 和 CD 的缠结

旋转

- 考虑两根绳子 AB 和 CD 的缠结
- 两个操作

旋转

- 考虑两根绳子 AB 和 CD 的缠结
- 两个操作
  - S - *旋转*：将正方形 $ABCD$ 旋转 90° ccw
  - R - *旋转*：交换 A 端和 D 端，绕对方顺时针旋转

旋转

- 考虑两根绳子 *AB* 和 *CD* 的缠结
- 两个操作
  - S - *旋转*：将正方形 *ABCD* 旋转 90° ccw
  - R - *旋转*：交换 *A* 端和 *D* 端，绕对方顺时针旋转
- 给你一些初始操作序列

旋转

- 考虑两根绳子 *AB* 和 *CD* 的缠结
- 两个操作
  - S - *旋转*：将正方形 *ABCD* 旋转 90° ccw
  - R - *旋转*：交换 *A* 端和 *D* 端，绕对方顺时针旋转
- 为您提供一些初始操作序列 执行更多操作以解开绳索
-

旋转

- 问题基于一个已知的情节

旋转

- 问题基于一个已知的情节 Conway's
- Rational Tangles

旋转

- 问题基于一个已知的情节 Conway's
- Rational Tangles
- 请随意搜索并观看一些人们用两根绳子玩耍的视频！

旋转

- 重新定义操作 S

旋转

- 重新定义操作 S
- 让我们想象一下 Ka-BAN 向下一侧旋转，而不是将所有东西旋转 90° ccw。

旋转

- 重新定义操作 S
- 让我们想象一下 Ka-BAN 向下一侧旋转，而不是将所有东西旋转 90° ccw。
- 现在 R 的旋转方向不是 A 和 D，而是沿着 Ka-BAN 目前靠近

的一侧

旋转

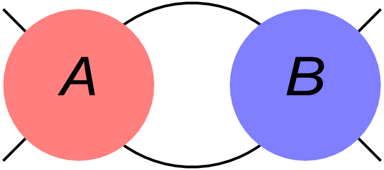
- 定义运算 ÷ - *水平和*

旋转

- 定义运算 $\div$ - *水平和*
- $A \div B$ 是将 B 附加到 A 的右边得到的纠结线

旋转

- 定义运算 $\div$ - *水平和*
- $A \div B$ 是将 B 附加到 A 的右边得到的纠结线



旋转

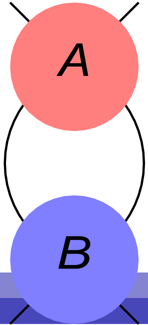
■定义操作  — *纵向总和*

旋转

- 定义操作 $\cdot|\cdot$ — *纵向总和*
- $\cdot|\cdot$ AB 是将 B 连接到 A 的底部后得到的纠结线

旋转

- 定义操作 $\cdot|\cdot$ — *纵向总和*
- $\cdot|\cdot$ AB 是将 B 连接到 A 的底部后得到的纠结线



旋转

- 两个基本纠结：

旋转

- 两个基本纠结：
  - 水平单元 H

旋转

- 两个基本纠结：
  - 水平单元 H

旋转

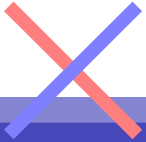
- 两个基本纠结：
  - 水平单元 H

- 垂直单位 V

旋转

- 两个基本纠结：
  - 水平单元 H

- 垂直单位 V



旋转

- 因此，有四种可能的 R 应用于理性纠结 T :

旋转

- 因此，有四种可能的 R 应用于理性纠结 T :
  - $T' \to T \mathbin{\cdot|\cdot} V$
  - $T' \to T \div H$
  - $T' \to V \mathbin{\cdot|\cdot} T$
  - $T' \to H \div T$

旋转

- 如果可以通过 R 和 S 的序列从初始纠结 0 到达该纠结，我们就称该纠结为*合理*纠结。

旋转

- 如果可以通过 R 和 S 的序列从初始纠结 0 到达该纠结，我们就称该纠结为*合理*纠结。
- 如果两个有理切点可以通过高于正方形的平滑变形相互到达，则它们是*等价*的

旋转

定理

任何合理的纠结都等同于水平/垂直翻转的 纠结。

旋转

定理

任何合理的纠结都等同于水平/垂直翻转的 纠结。

证明。

通过感应。 □

旋转

定理

任何合理的纠结都等同于水平/垂直翻转的 纠结。

证明。

通过感应。

推论

$\div$ *和* $\cdot|\cdot$ *是交换式：* $A \div B = B \div A$, $A \cdot|\cdot B = B \cdot|\cdot A$.

旋转

- $T \div H = H \div T$

旋转

- $T \div H = H \div T$
- $T \mathbin{\cdot\!|\!\cdot} V = V \mathbin{\cdot\!|\!\cdot} T$

旋转

- $T \div H = H \div T$
- $T \mathbin{\rotatebox{90}{$\div$}} V = V \mathbin{\rotatebox{90}{$\div$}} T$
- 因此，只有两种可能的 R：

旋转

- $T \div H = H \div T$
- $T \cdot|\cdot V = V \cdot|\cdot T$
- 因此，只有两种可能的 R：
  - $T' \to T \div H$
  - $T' \to T \cdot|\cdot V$

旋转

- $T \div H = H \div T$
- $T \mathbin{\cdot|\cdot} V = V \mathbin{\cdot|\cdot} T$
- 因此，只有两种可能的 R：
  - $T' \to T \div H$
  - $T' \to T \mathbin{\cdot|\cdot} V$
- 此外，S

旋转

- $T \div H = H \div T$
- $T \cdot\!|\!\cdot V = V \cdot\!|\!\cdot T$
- 因此，只有两种可能的 R：
  - $T' \to T \div H$
  - $T' \to T \cdot\!|\!\cdot V$
- 此外，S 撤销 S S^{-1}
- $\sim$ S

旋转

- 如何撤销 R?

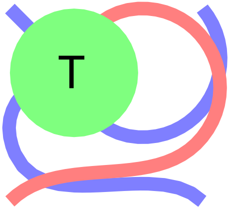
旋转

- 如何撤销 R?
- 例如，如何转换 $T \div H' \to T$?

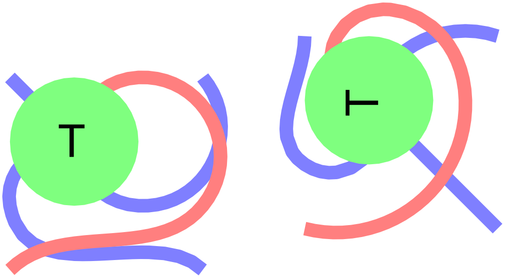
旋转

- 如何撤销 R?
- 例如，如何转换 $T \div H' \to T$?
- 让我们试着增加一个垂直单位：$(T \div H)\ V$

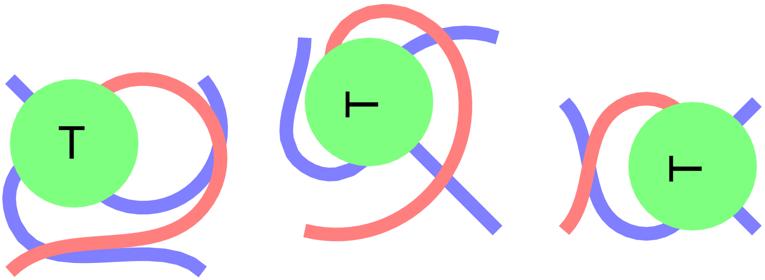
旋转



旋转



旋转



旋转

所以 $((T \div H) \quad V) \div H$ 实际上只是一个旋转的 T

旋转

- 所以（$(T \div H)$ V）$\div$ H 实际上只是一个旋转的 T
- 三个 S 之后，机器人也会改变方向

旋转

所以 $((T \div H) \quad V) \div H$ 实际上只是一个旋转的 T

- 因此，RSRSRS $\sim$ id
-

旋转

- 所以 $((T \div H) \quad V) \div H$ 实际上只是一个旋转的 T
  - 因此，RSRSRS $\sim$ id
  -
  - $R^{-1} \sim$ SRSRS

旋转

- 因此，我们可以通过*某种方式*撤销任何序列

旋转

- 因此，我们可以*以某种方式*撤销任何序列 逆序列
-

  ，用 SRSRS 替换每个 R

旋转

- 因此，我们可以*以某种方式*撤销任何序列 逆向序
-
- 列，用 SRSRS 替换每个 R，但这是最短的序列吗

？

旋转

- 因此，我们可以*以某种方式*撤销任何序列 逆向序
-
- 列，用 SRSRS 替换每个 R，但这是最短的序列吗

  ？
- 可能没有：

旋转

- 因此，我们可以*以某种方式*撤销任何序列 逆向序
-
- 列，用 SRSRS 替换每个 R，但这是最短的序列吗

  ？
- 可能没有：
  - 可删除的 SS 子字符串

旋转

- 因此，我们可以*以某种方式*撤销任何序列 逆向序
-
- 列，用 SRSRS 替换每个 R，但这是最短的序列吗

  ？
- 可能没有：
  - 可删除的 SS 子字符串

■可删除子串 RSRSRS

旋转

- 因此，我们可以*以某种方式*撤销任何序列 逆向序
-
- 列，用 SRSRS 替换每个 R，但这是最短的序列吗

  ?
- 可能没有：
  - 可删除的 SS 子字符串

- 可删除子串 RSRSRS
- 后缀 RS 可用 S 代替（如果我们以零纠结结束）

旋转

- 因此，我们可以*以某种方式*撤销任何序列 逆向序
-
- 列，用 SRSRS 替换每个 R，但这是最短的序列吗

  ？
- 可能没有：
  - 可删除的 SS 子字符串

- 可删除子串 RSRSRS
- 后缀 RS 可用 S 代替（如果我们以零纠结结束）

- 其实，这些就足够了！

旋转

- 让我们给每个有理纠缠赋予一个有理数（或 ∞

旋转

- 让我们为每个有理纠缠赋予一个有理数（或 ∞） 初始纠缠
-

  为 0

旋转

- 让我们为每个有理纠缠赋予一个有理数（或 ∞） 初始纠缠
- 为 0
- $x \xrightarrow{S} -\frac{1}{x}$

旋转

- 让我们为每个有理纠缠赋予一个有理数（或 ∞） 初始纠缠
- 为 0
- $x \xrightarrow{S} -\frac{1}{x}$
- $x \xrightarrow{R} x + 1$

旋转

- 让我们为每个有理纠缠赋予一个有理数（或 ∞） 初始纠缠
- 为 0
- $x \xrightarrow{S} -\frac{1}{x}$
- $x \xrightarrow{R} x + 1$
- $\frac{1}{0} = \infty$, $-\frac{1}{\infty} = 0$, $\infty + 1 = \infty$

旋转

定理

当且仅当两个有理数相等时，两个有理纠缠才等价。

定理

如果从 0 开始得到 x 的 R 和 S 序列不包含子串 SS、$RSRSRS$ 或前缀 SR，则无法缩短。

如果从 x 得到0 的 R 和 S 序列不包含子串 SS、$SRSRSR$ 或后缀 RS

，则无法缩短。

问题 J：第一个十亿

构思： Sergey Kopeliovich 谢尔盖-科佩利奥维奇
开发： 谢尔盖-科佩利奥维奇

奇

编辑：　　米哈伊尔-伊万诺夫

第一个十亿

- 我们生成了两组正整数，每组大小为 n，和为 10^9

第一个十亿

- 我们生成了两组正整数，每组大小为 n，和为 10^9
- 它们被合并并洗牌成一个大小为 $N = 2n$ 的集合

第一个十亿

- 我们生成了两组正整数，每组大小为 n，和为 10^9
- 将它们合并并洗牌成一个大小为 $N = 2n$ 的集合 还原一个
- 任意大小的子集，其总和为 10^9

第一个十亿

- 如果有 $N \le 18$ 个元素，我们可以在 $O(2^N)$ 内求解

第一个十亿

- 如果有 $N \le 18$ 个元素，我们可以在 $O(2^N)$ 内求解
- 如果有 $N \le 36$ 个元素，我们可以使用中间会商方法在 $O(N - 2^N)$ 内求解。

第一个十亿

- 如果有 $N \leq 18$ 个元素，我们可以在 $O(2^N)$ 内求解
- 如果有 $N \leq 36$ 个元素，我们可以使用中间会商方法在 $O(N \cdot 2^N)$ 内求解。
- 如果 $N > 36$ 呢?

第一个十亿

- 将数字贪婪地分配到 $B = 36$ 个桶中

第一个十亿

- 将数字贪婪地分配给 $B = 36$ 个桶 在 $O^*(2^{B/2})$ 时间内求解

第一个十亿

- 将数字贪婪地分配给 $B = 36$ 个桶 在 $O^*(2^{B/2})$ 时间内求解
-
- 由于 2^B 比 10^9 大得多，因此存在解

问题 K：任务和错误

创意　尼古拉-杜布楚克

开发：　尼古拉-杜布楚克 编

辑：　尼古拉-杜布楚克

T 询问和虫子

- 有一个错误列表，每个错误都有一个任务列表

T 询问和虫子

- 有一个错误列表，每个错误都有一个任务列表 创建一个任务
-
  列表，每个任务都有一个错误列表

T 询问和虫子

- 想法：创建任务地图，添加错误

T 询问和虫子

- 想法：创建任务地图，添加错误
- 小心输出结果，按数字顺序而不是按词典顺序排序

问题 L：糖果

创意： Ivan Bochkov Ivan Bochkov 开发： 伊万-博奇科夫 编

辑：　　　　伊万-博奇科夫

糖果

- 我们有三个整数 x_1, x_2, x_3，最初都是零。

糖果

- 我们有三个整数 x_1, x_2, x_3，最初都是零。
- 在一个步骤中，我们将其中一个增加 1，但 x_1 应该是过程中的最大值。

糖果

- 我们有三个整数 x_1, x_2, x_3，最初都是零。
- 在一个步骤中，我们将其中一个增加 1，但 x_1 应该是过程中的最大值。
- 计算得到 $x_1 = a$、$x_2 = b$、$x_3 = c$ 的方法数。

糖果

- 对于 a、b、$c < 500$，我们可以使用动态编程解决方案在 $O(abc)$ 时间内生成答案。

糖果

- 对于 a、b、$c < 500$，我们可以使用动态编程解决方案在 $O(abc)$ 时间内生成答案。
- 原来，$a = b$（以及对称的 $a = c$）的答案可以用一个简单的公式来描述。

糖果

- 对于 a、b、$c < 500$，我们可以使用动态编程解决方案在 $O(abc)$ 时间内生成答案。
- 原来，$a = b$（以及对称的 $a = c$）的答案可以用一个简单的公式来描述。
- 也就是说，如果问题的答案是 $f(a, b, c)$，那么

$f(a, a, 0) = \frac{(2n)!}{n!(n+1)!}$，加泰罗尼亚数。

糖果

- 对于 a、b、$c < 500$，我们可以使用动态编程解决方案在 $O(abc)$ 时间内生成答案。
- 原来，$a = b$（以及对称的 a = c）的答案可以用一个简单的公式来描述。
- 也就是说，如果问题的答案是 $f(a, b, c)$，那么

$n+1)\ !$

- 此外，$f(a,a,k) = \frac{(2a+k)!}{a!\,(a+1)!}\,k!\ -\ Q_a$ $m{=}a{-}k{+}1$ $\frac{2m}{2m+1}$

糖果

- 如何证明？这是一个具有超几何系数的等式，可以用多项式递推技术来证明。

糖果

- 如何证明？这是一个具有超几何系数的等式，可以用多项式递推技术来证明。
- 例如，您可以在多伦-蔡尔伯格（Doron Zeilberger）所著的《A=B》一书中了解到这一点。

糖果

- 如何证明？这是一个具有超几何系数的等式，可以用多项式递推技术来证明。
- 例如，您可以在多伦-蔡尔伯格（Doron Zeilberger）所著的《A=B》一书中了解到这一点。
- 我不知道这个公式的组合含义。如果有人知道，请与我分享！

糖果

- 如何处理一般（a、b、c）？

糖果

- 如何处理一般（a、b、c）？
- 如果我们放弃条件，考虑得到（a、b、c）的所有方法 $x_1 \geq x_2, x_3$。

糖果

- 如何处理一般情况（a、b、c）？
- 如果我们放弃条件，考虑得到（a、b、c）的所有方法 $x_1 \geq x_2, x_3$。
- 这还会算上一些额外的方法。它们看起来像什么？

糖果

- 如何处理一般情况（a、b、c）？
- 如果我们放弃条件，考虑得到（a、b、c）的所有方法 $x_1 \geq x_2, x_3$。
- 这还会算上一些额外的方法。它们看起来像什么?
- 对于某个 x、y，我们到达点（x，x，y）或（x，y，x），再走一步到达（x，$x+1$，y）或 x，y，$x+1$，然后以某种

方式到达（a，b，c）。

糖果

- 如何处理一般情况（a、b、c）？
- 如果我们放弃条件，考虑得到（a、b、c）的所有方法 $x_1 \geq x_2, x_3$。
- 这还会算上一些额外的方法。它们看起来像什么?
- 对于某个 x、y，我们到达点（x，x，y）或（x，y，x），再走一步到达（x，$x+1$，y）或 x，y，$x+1$，然后以某种

方式到达（a，b，c）。

- 如果我们固定 x、y，这个数字可以用 $f(x, x, y)$ 计算出来。

糖果

- 如何处理一般情况（a、b、c）？
- 如果我们放弃条件，考虑得到（a、b、c）的所有方法 $x_1 \geq x_2, x_3$。
- 这还会算上一些额外的方法。它们看起来像什么?
- 对于某个 x、y，我们到达点（x，x，y）或（x，y，x），再走一步到达（x，$x+1$，y）或 x，y，$x+1$，然后以某种

方式到达（a，b，c）。

- 如果我们固定 x、y，这个数字可以用 $f(x, x, y)$ 来计算。
- 我们可以检查所有 x, y 对。渐近公式为 $O(a^2)$.

糖果

有组合意义吗?

问题 M：厕所

创意	列昂尼德-迪亚奇 科夫-尼基塔-加 沃伊
开发	尼基塔-加沃

伊 编辑　　伊万-博奇科

夫

oilets

- 考虑在圆形办公室内设置卫生间。
- 员工们在办公室内向两个可能的方向之一移动，寻找空马桶。
- 员工不会理会被占用的厕所，当他们找到一个空置的厕所时，他们会占用一定的时间，每个员工都有自己的时间规定。
- 我们需要为每位员工确定他们将在何时占用哪个卫生间。

- 当两名员工争抢一个厕所时，以步行时间或员工指数来打破平局。

oilets

- 我们想模拟这个过程。
- 我们需要处理三种可能的情况：
  1. 一名员工发现了一个免费厕所。
  2. 厕所可用。
  3. 新员工开始工作。

oilets

- 我们想模拟这个过程。
- 我们需要处理三种可能的情况:
  1. 一名员工发现了一个免费厕所。
  2. 厕所可用。
  3. 新员工开始工作。
- 我们的所有活动基本上都是厕所和员工的增加和减少，因此

，如果我们能在这些询问下保持最近的未来活动，我们就赢了。

优化活动数量

- 第一个想法是将所有此类事件保存在一个堆中。
- 然而，它们有 $\Theta(n^2)$ 个，所以我们不能直接这样做。

优化活动数量

- 第一个想法是将所有此类事件保存在一个堆中。
- 然而，它们有 $\Theta(n^2)$ 个，所以我们不能直接这样做。
- 我们只对每个员工最近的厕所以及每个厕所最近的两个（每个方向一个）员工感兴趣。
- 我们可以使用 std::set 以 $O(\log(n+m))$ 的时间找到它们。

优化活动数量

- 第一个想法是将所有此类事件保存在一个堆中。
- 然而，它们有 $\Theta(n^2)$ 个，所以我们不能直接这样做。
- 我们只对每个员工最近的厕所以及每个厕所最近的两个（每个方向一个）员工感兴趣。
- 我们可以使用 std::set 以 $O(\log(n+m))$ 的时间找到它们。

- 其余的观察结果是，每次更改后，我们只能更新最近的厕所和员工，每次查询只需增加一定数量的事件。
- 时间复杂度为 $O(n \log(n + m))$。

问题 N：（未）标注的图形

创意： Mikhail Ivanov 开发：米哈伊尔-伊万诺夫 米哈伊尔-伊万诺

夫 编辑： 米哈伊尔-伊万诺

夫

(未）标注的图形

- 给您一个带标签的图形 G

(未) 标注的图形

- 给您一个带标签的图形 G
- 用无标记图 H 进行编码

(未）标注的图形

- 给您一个带标签的图形 G
- 用无标记图 H 进行编码
- 在解码之前，应先对 H 的顶点进行洗牌

(未）标注的图形

- 想法：复制初始图 G，用二进制写出每个顶点的编号

(未）标注的图形

- 想法：复制初始图 G，用二进制写出每个顶点的编号
- 创建 $\ell = \lceil \log_2 n \rceil$ 辅助顶点 $B_0, \ldots, B_{\ell-1}$，对这些数字进行编码

(未）标注的图形

- 想法：复制初始图 G，用二进制写出每个顶点的编号
- 创建 $\ell = \lceil \log_2 n \rceil$ 辅助顶点 $B_0, \ldots, B_{\ell-1}$，对这些数字进行编码
- 如何区分主顶点和辅助顶点?

(未) 标注的图形

- 再添加两个顶点 T_0, T_1，并将它们与所有主顶点连接起来

(未）标注的图形

- 再添加两个顶点 T_0, T_1，并将它们与所有主顶点连接起来
- 现在，T_0 和 T_1 是唯一有重合邻域的顶点

(未）标注的图形

- 再添加两个顶点 T_0, T_1，并将它们与所有主顶点连接起来
- 现在，T_0 和 T_1 是唯一有重合邻域的顶点
- 我们可以找到主要顶点，只需枚举它们即可

(未）标注的图形

- 如何找到辅助顶点上的阶数?

(未）标注的图形

- 如何找到辅助顶点的顺序？添加新顶点 B_ℓ，添
-

加路径 $B\ B_{01}\ .\ .B_\ell$

(未）标注的图形

- 如何找到辅助顶点的顺序？添加新顶点 B_ℓ，
-
- 添加一条路径 $B\ B_{01}\ .\ .B_\ell$ 现在 B_ℓ 是唯一的辅助叶

(未）标注的图形

- 如何找到辅助顶点的顺序？添加新顶点 B_ℓ，
-
- 添加一条路径 $B\ B_{01}\ .\ .B_\ell$ 现在 B_ℓ 是唯一的辅助叶
- $n + \lceil \log_2\ n \rceil\ +$ 共 3 个顶点

问题 O：神秘序列

创意 尼古拉-杜布楚克

开发： 尼古拉-杜布楚克 编

辑： 尼古拉-杜布楚克

神秘序列

- 有一个公式：

$$x_{i+2} = a \cdot x_{i+1} + b \cdot x_i$$

神秘序列

- 有一个公式：

$$x_{i+2} = a \cdot x_{i+1} + b \cdot x_i$$

- 任务是在只知道第一个和最后一个数字的情况下重建序

列的所有元素：X_1 和 X_N

神秘序列

- 使用二进制搜索，找到 x_2，达到所需的精度 x_N

神秘序列

- 使用二进制搜索，找到 x_2，达到所需的精度 x_N
- 或数学解决方案：计算矩阵 A B 的幂后 A B 后，用
x 和 x 计算 x 2 1 N
1 0