构造题选讲

虞皓翔

IIS, Tsinghua University

January 25, 2022

中国计算机学会
China Computer Federation

1 网格类
- Parks
- 挑战哈密顿 2
- Pattern Lock (改编版)

2 序列类
- Secret of Tianqiu Valley
- 开关
- Quick sort
- Balance the Cards

3 图论类
- 最大生成掌
- Evil
- Jeopardy
- GCD Land
- City
- 大鱼划水

4 杂题
- 仓颉造数
- 同构判定海蜇

中国计算机学会 China Computer Federation

题目名称 题目来源

给定平面上 n 个互不相同且坐标形如 $(2x_i, 2y_i)$ $(x_i, y_i \in \mathbb{Z})$ 的点，每对距离为 2 的点之间连有一条边，保证所得的图 G 连通。

你需要构造 G 的一棵生成树 $T=(V,E)$，同时构造 $n-1$ 个不同的形如 $(2x_i+1, 2y_i+1)$ $(x_i, y_i \in \mathbb{Z})$ 的点，使得存在这些点到 E 的一一映射 φ 使得 p 和 $\varphi(p)$ 的两端点构成等腰直角三角形。

$1 \le n \le 2 \times 10^5$。

题目名称 题目来源

给定平面上 n 个互不相同且坐标形如 $(2x_i, 2y_i)$ $(x_i, y_i \in \mathbb{Z})$ 的点，每对距离为 2 的点之间连有一条边，保证所得的图 G 连通。

你需要构造 G 的一棵生成树 $T = (V, E)$，同时构造 $n-1$ 个不同的形如 $(2x_i+1, 2y_i+1)$ $(x_i, y_i \in \mathbb{Z})$ 的点，使得存在这些点到 E 的一一映射 φ 使得 p 和 $\varphi(p)$ 的两端点构成等腰直角三角形。

$1 \le n \le 2 \times 10^5$。

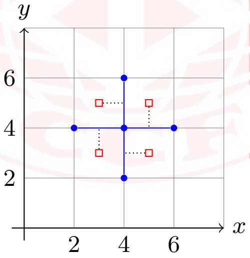
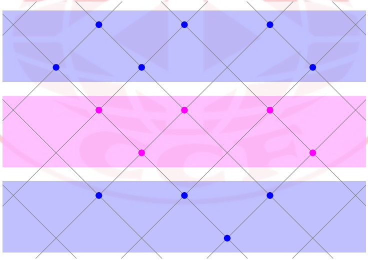
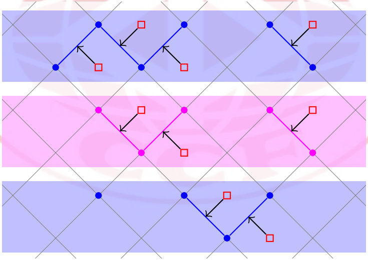


图 1

考虑将坐标系旋转 45°，所有点可以根据新的 y 坐标两两配对。



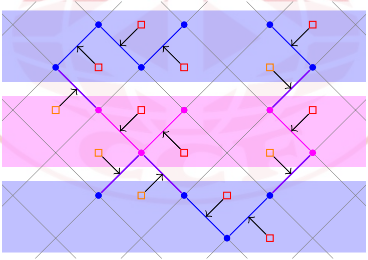
考虑将坐标系旋转 45°，所有点可以根据新的 y 坐标两两配对。对于同一层的点，可以将其连接并选择其右侧的点与之对应。



考虑将坐标系旋转 45°，所有点可以根据新的 y 坐标两两配对。

对于同一层的点，可以将其连接并选择其右侧的点与之对应。

对于不同层的点，考虑连通其的边，对于固定的两个连通块，我们考虑连接其的边中最靠左的那条，并选择其左侧的点与之对应。



中国计算机学会 China Computer Federation

给定无向简单图 $G=(V,E)$，其中 $V=\{(i,j)\mid 0\le i\le C-1, 0\le j\le R-1$ 且 i,j 为整数$\}$，E 为所有距离为 1 或 $\sqrt{2}$ 的点对构成的二元组。你需要找到 G 的一个 Hamilton 圈，满足：

- 该圈的任意连续两条边不平行 (等价地,任意连续三点不共线)；
- (视每条边为直线段) 任意两条边不在非顶点处相交。

或说明其不存在。$R\times C\le 10^6$。

给定无向简单图 $G=(V,E)$，其中 $V=\{(i,j)\mid 0\le i\le C-1,$ $0\le j\le R-1$ 且 i,j 为整数$\}$，E 为所有距离为 1 或 $\sqrt{2}$ 的点对构成的二元组。你需要找到 G 的一个 Hamilton 圈，满足：

- 该圈的任意连续两条边不平行 (等价地,任意连续三点不共线);
- (视每条边为直线段) 任意两条边不在非顶点处相交。

或说明其不存在。$R\times C\le 10^6$。

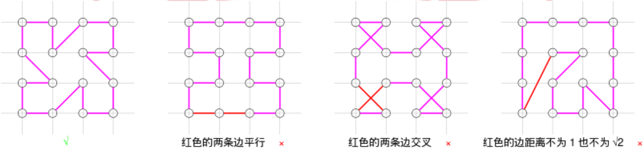


图 2

不妨设 $R \le C$。容易证明当 $R \le 3$ 时除 $(R,C) = (2,2)$ 有平凡解外其余情况都无解，暴搜知 $(R,C) = (5,5)$ 时无解。

不妨设 $R \leq C$。容易证明当 $R \leq 3$ 时除 $(R, C) = (2, 2)$ 有平凡解外其余情况都无解，暴搜知 $(R, C) = (5, 5)$ 时无解。

其余情况均可构造解。我们增强条件，希望从 $(0, 0)$ 开始的前两步分别是右、上。根据下图可知我们可以从 (R, C) 的答案归纳得到 $(R + 4, C + 4)$ 的答案。

不妨设 $R \le C$。容易证明当 $R \le 3$ 时除 $(R, C) = (2, 2)$ 有平凡解外其余情况都无解，暴搜知 $(R, C) = (5, 5)$ 时无解。

其余情况均可构造解。我们增强条件，希望从 $(0, 0)$ 开始的前两步分别是右、上。根据下图可知我们可以从 (R, C) 的答案归纳得到 $(R + 4, C + 4)$ 的答案。

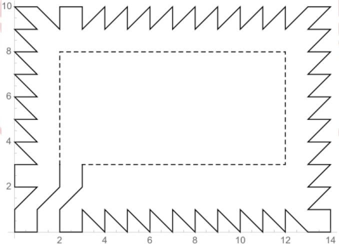


图 3

于是只需考虑 $4 \le R \le 7$ 的情形以及 $(R, C) = (9, 9)$。这些情况可以通过手玩、搜索等方法来完成。

给定完全图 $G = (V, E)$，其中 $V = \{(i, j) \mid 0 \le i \le C - 1, 0 \le j \le R - 1$ 且 i, j 为整数$\}$。你需要找到 G 的一个 Hamilton 圈，满足视每条边为直线段后：

- 对任意点 $v \in V$，它关联的两条边构成一个锐角。
- 每一条边不能经过除端点外的其它顶点。

或说明其不存在。$R \times C \le 10^6$。

给定完全图 $G=(V,E)$，其中 $V=\{(i,j)\mid 0\le i\le C-1, 0\le j\le R-1$ 且 i,j 为整数$\}$。你需要找到 G 的一个 Hamilton 圈，满足视每条边为直线段后：

- 对任意点 $v\in V$，它关联的两条边构成一个锐角。
- 每一条边不能经过除端点外的其它顶点。

或说明其不存在。$R\times C\le 10^6$。

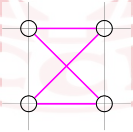
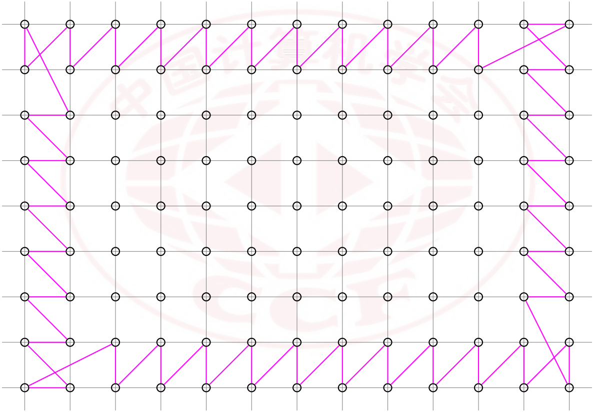


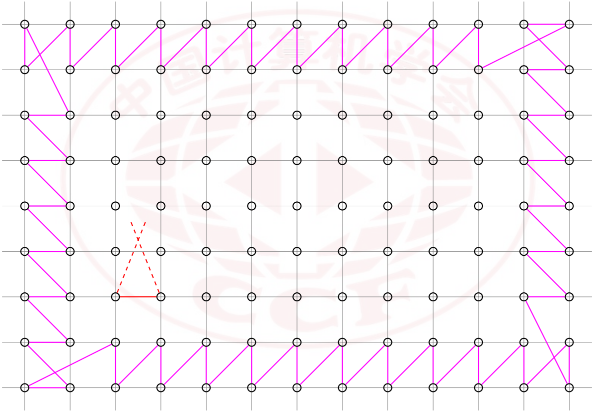
图 4

故技重施。仍然考虑通过 (R, C) 的解得到 $(R+4, C+4)$ 的解。

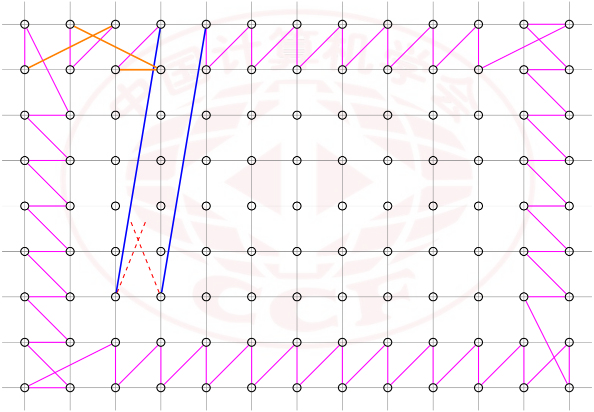
故技重施。仍然考虑通过 (R, C) 的解得到 $(R+4, C+4)$ 的解。



故技重施。仍然考虑通过 (R, C) 的解得到 $(R+4, C+4)$ 的解。



故技重施。仍然考虑通过 (R, C) 的解得到 $(R+4, C+4)$ 的解。



中国计算机学会 China Computer Federation

有 n 盏灯排成一圈，标号为 $0, 1, \ldots, n-1$。给定每盏灯的初始状态 (开或关)，你可以进行若干次如下操作：

- 选择一盏关闭的灯 (设标号为 i)，切换标号为 $i-1, i, i+1$ 的灯的状态 (下标在 $\bmod n$ 意义下)。

你需要在 $2n$ 次操作以内 (含) 将所有灯点亮或说明无法做到。$3 \le n \le 10^5$。

忽略“只能选择关闭的灯”的限制，容易发现这是一个 $\mathbb{F}_2$ 下线性方程组问题：

用 1 表示关闭，0 表示开启。设 $\boldsymbol{s}$ 为状态向量，$\boldsymbol{t}$ 为操作向量，$\mathbf{A} = \begin{pmatrix} 1 & 1 & & & 1 \\ 1 & 1 & 1 & & \\ & 1 & 1 & \ddots & \\ & & \ddots & \ddots & 1 \\ 1 & & & 1 & 1 \end{pmatrix}$，则

$$\boldsymbol{s} = \mathbf{A}\boldsymbol{t}. \tag{1}$$

中国计算机学会 China Computer Federation

忽略“只能选择关闭的灯”的限制，容易发现这是一个 $\mathbb{F}_2$ 下线性方程组问题：

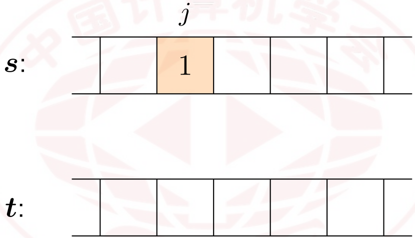
用 1 表示关闭，0 表示开启。设 $\boldsymbol{s}$ 为状态向量，$\boldsymbol{t}$ 为操作向量，$\mathbf{A} = \begin{pmatrix} 1 & 1 & & & 1 \\ 1 & 1 & 1 & & \\ & 1 & 1 & \ddots & \\ & & \ddots & \ddots & 1 \\ 1 & & & 1 & 1 \end{pmatrix}$，则

$$\boldsymbol{s} = \mathbf{A}\boldsymbol{t}. \tag{1}$$

解方程 (1) 后，不妨设 $\boldsymbol{t}$ 为一组解，我们说明，只要解存在，我们一定可以在 $2\,|\boldsymbol{t}|$ 步内操作完成，其中 $|\boldsymbol{t}|$ 表示 $\boldsymbol{t}$ 中 1 的数量。

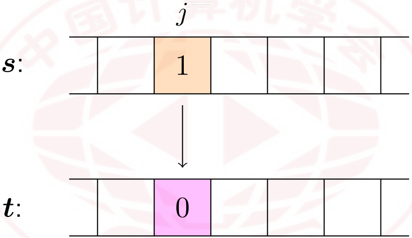
中国计算机学会 China Computer Federation

不妨设不存在 i 使得 $\boldsymbol{s}_i = \boldsymbol{t}_i = 1$，且存在 j 使得 $\boldsymbol{s}_j = 1$。

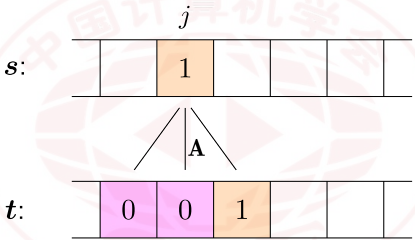


China Computer Federation

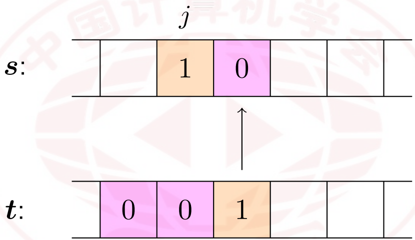
不妨设不存在 i 使得 $\boldsymbol{s}_i = \boldsymbol{t}_i = 1$，且存在 j 使得 $\boldsymbol{s}_j = 1$。



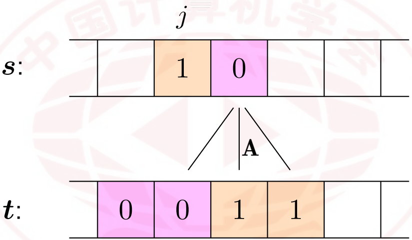
不妨设不存在 i 使得 $\boldsymbol{s}_i = \boldsymbol{t}_i = 1$，且存在 j 使得 $\boldsymbol{s}_j = 1$。



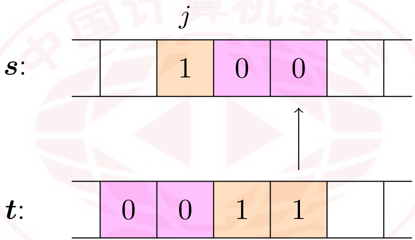
不妨设不存在 i 使得 $s_i = t_i = 1$，且存在 j 使得 $s_j = 1$。



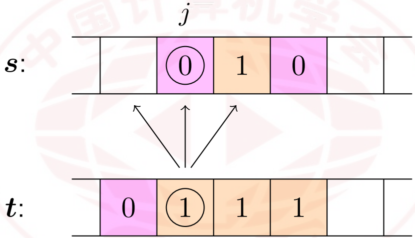
不妨设不存在 i 使得 $s_i = t_i = 1$，且存在 j 使得 $s_j = 1$。



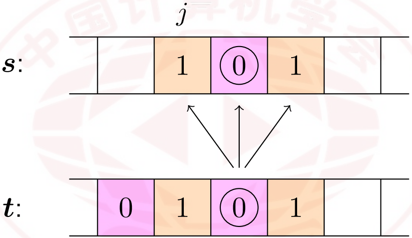
不妨设不存在 i 使得 $\boldsymbol{s}_i = \boldsymbol{t}_i = 1$，且存在 j 使得 $\boldsymbol{s}_j = 1$。



不妨设不存在 i 使得 $\boldsymbol{s}_i=\boldsymbol{t}_i=1$，且存在 j 使得 $\boldsymbol{s}_j=1$。

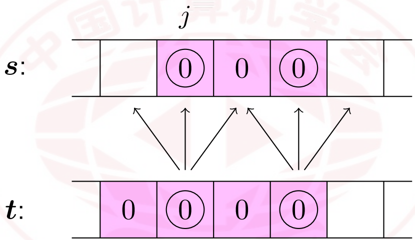


不妨设不存在 i 使得 $s_i = t_i = 1$，且存在 j 使得 $s_j = 1$。



中国计算机学会 China Computer Federation

不妨设不存在 i 使得 $s_i = t_i = 1$，且存在 j 使得 $s_j = 1$。



中国计算机学会 China Computer Federation

有 n 盏灯排成一列，标号为 $1, 2, \ldots, n$，给定初始状态 s 和目标状态 t，你可以进行若干次如下操作：

- 选择一段区间 l, r $(1 \le l \le r \le n)$，你需要满足区间内点亮的灯数等于关闭的灯数，然后切换该区间内所有灯的状态。

你需要在满足 $\sum(r - l + 1) \le 6\,666\,666$ 的条件下将 s 变成 t 或说明无法做到。$1 \le n \le 10^5$。

不难发现操作不改变亮着的灯的总数，所以当 $|s| \neq |t|$ 时无解。

中国计算机学会
China Computer Federation

不难发现操作不改变亮着的灯的总数，所以当 $|s| \neq |t|$ 时无解。

用 $0, 1$ 表示两种状态，注意到操作可逆，故不妨设 t 有序(所有 0 都在 1 的左边)。

中国计算机学会 China Computer Federation

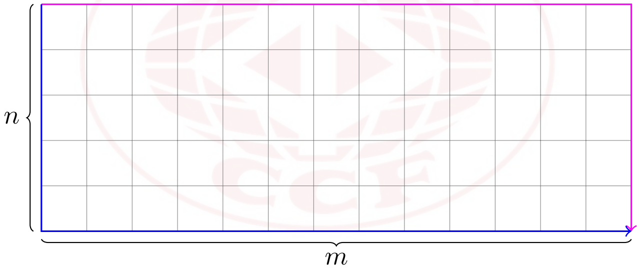
不难发现操作不改变亮着的灯的总数，所以当 $|\boldsymbol{s}| \neq |\boldsymbol{t}|$ 时无解。

用 $0,1$ 表示两种状态，注意到操作可逆，故不妨设 $\boldsymbol{t}$ 有序(所有 0 都在 1 的左边)。

先考虑如何将 $\underbrace{11\ldots1}_{n}\underbrace{00\ldots0}_{m}$ 变成 $\underbrace{00\ldots0}_{m}\underbrace{11\ldots1}_{n}$。

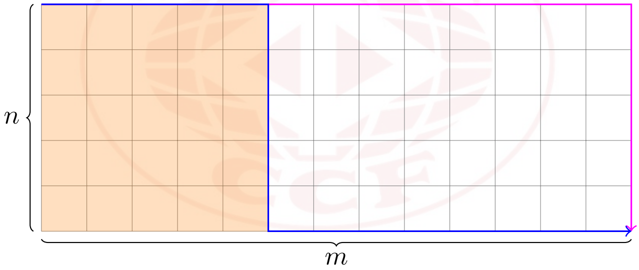
先考虑如何将 $\underbrace{11\ldots1}_{n}\underbrace{00\ldots0}_{m}$ 变成 $\underbrace{00\ldots0}_{m}\underbrace{11\ldots1}_{n}$。

将 0 视作 $\rightarrow$，1 视作 $\downarrow$，则一个序列可以看成一个 n 行 m 列的 (从左上角到右下角的) 格路：



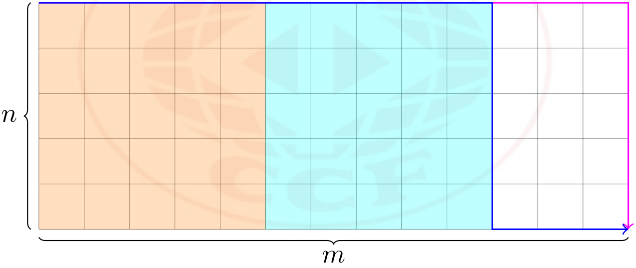
先考虑如何将 $\underbrace{11\ldots1}_{n}\underbrace{00\ldots0}_{m}$ 变成 $\underbrace{00\ldots0}_{m}\underbrace{11\ldots1}_{n}$。

将 0 视作 $\rightarrow$，1 视作 $\downarrow$，则一个序列可以看成一个 n 行 m 列的 (从左上角到右下角的) 格路：



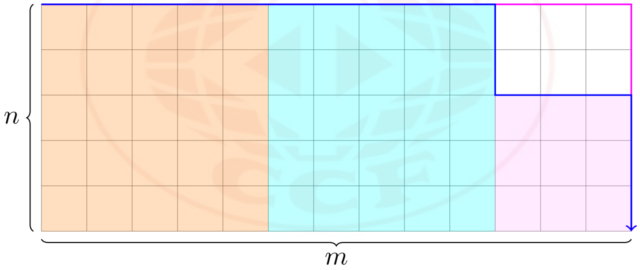
先考虑如何将 $\underbrace{11\ldots1}_{n}\underbrace{00\ldots0}_{m}$ 变成 $\underbrace{00\ldots0}_{m}\underbrace{11\ldots1}_{n}$。

将 0 视作 $\rightarrow$，1 视作 $\downarrow$，则一个序列可以看成一个 n 行 m 列的 (从左上角到右下角的) 格路：



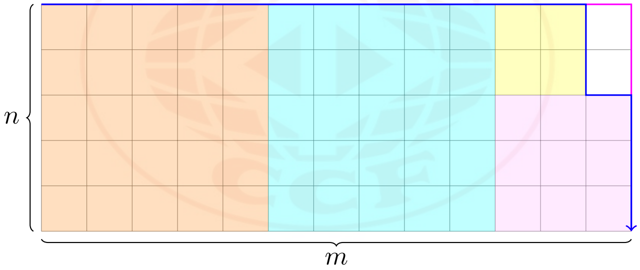
先考虑如何将 $\underbrace{11\ldots1}_{n}\underbrace{00\ldots0}_{m}$ 变成 $\underbrace{00\ldots0}_{m}\underbrace{11\ldots1}_{n}$。

将 0 视作 $\rightarrow$，1 视作 $\downarrow$，则一个序列可以看成一个 n 行 m 列的 (从左上角到右下角的) 格路：



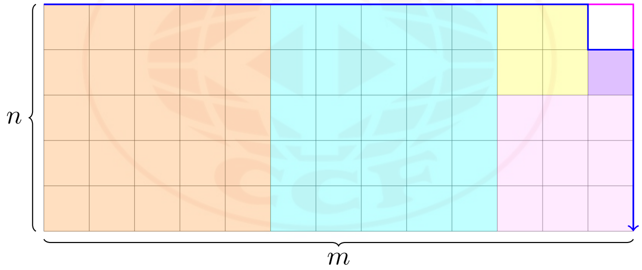
先考虑如何将 $\underbrace{11\ldots1}_{n}\underbrace{00\ldots0}_{m}$ 变成 $\underbrace{00\ldots0}_{m}\underbrace{11\ldots1}_{n}$。

将 0 视作 $\rightarrow$，1 视作 $\downarrow$，则一个序列可以看成一个 n 行 m 列的 (从左上角到右下角的) 格路：



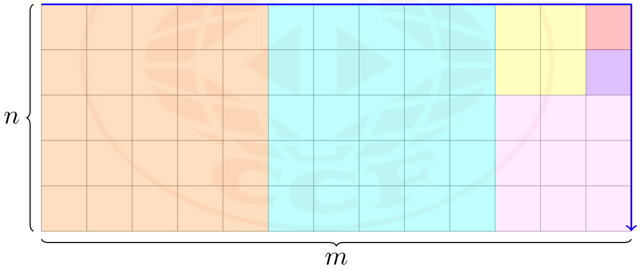
先考虑如何将 $\underbrace{11\ldots1}_{n}\underbrace{00\ldots0}_{m}$ 变成 $\underbrace{00\ldots0}_{m}\underbrace{11\ldots1}_{n}$。

将 0 视作 $\rightarrow$，1 视作 $\downarrow$，则一个序列可以看成一个 n 行 m 列的 (从左上角到右下角的) 格路：



先考虑如何将 $\underbrace{11\ldots1}_{n}\underbrace{00\ldots0}_{m}$ 变成 $\underbrace{00\ldots0}_{m}\underbrace{11\ldots1}_{n}$。

将 0 视作 $\rightarrow$，1 视作 $\downarrow$，则一个序列可以看成一个 n 行 m 列的 (从左上角到右下角的) 格路：



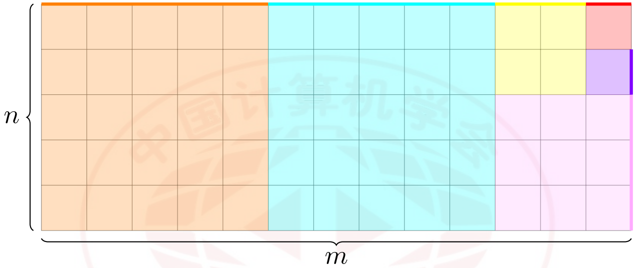
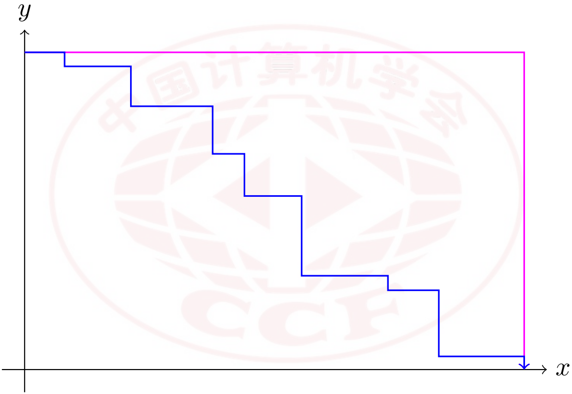


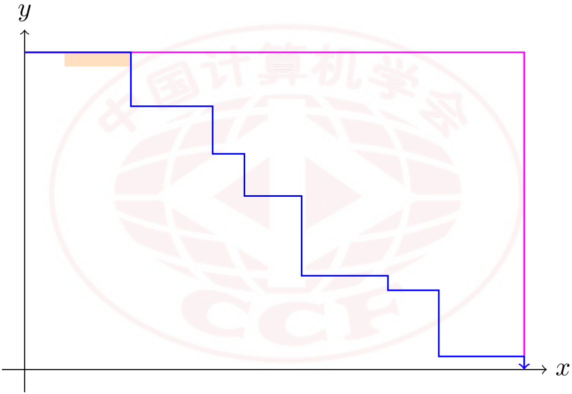
图 5

这样操作的费用为图中所有正方形的边长之和的 2 倍，这个值不超过 $2(n+m-1)$。

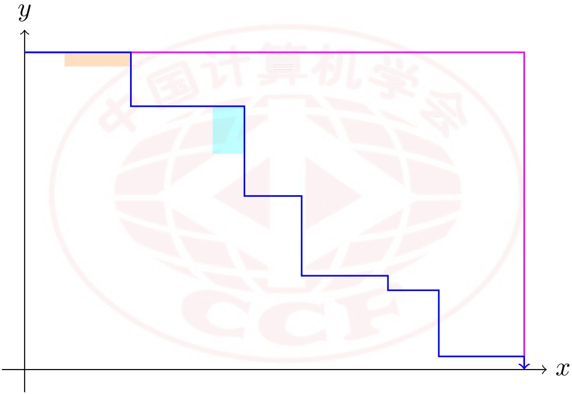
对于一般的情形，找到类似的 1/0 连续段，然后分治处理。



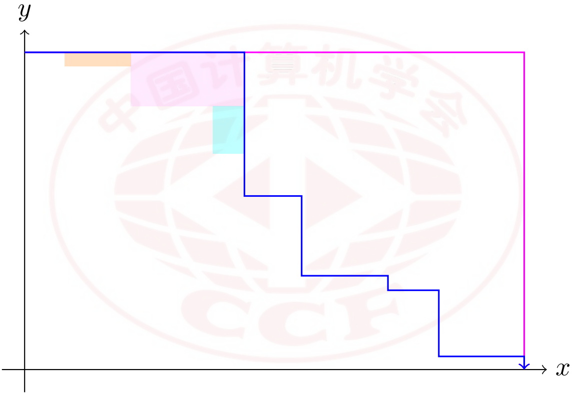
对于一般的情形，找到类似的 1/0 连续段，然后分治处理。



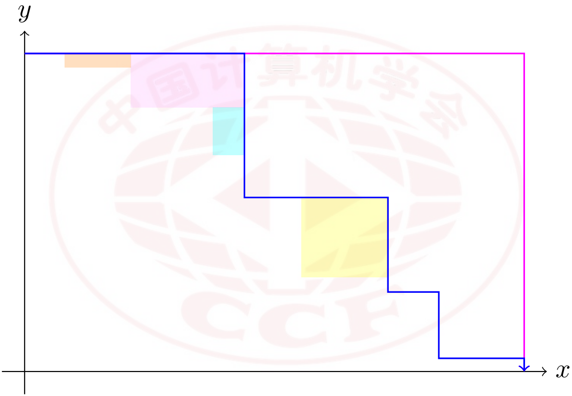
对于一般的情形，找到类似的 1/0 连续段，然后分治处理。



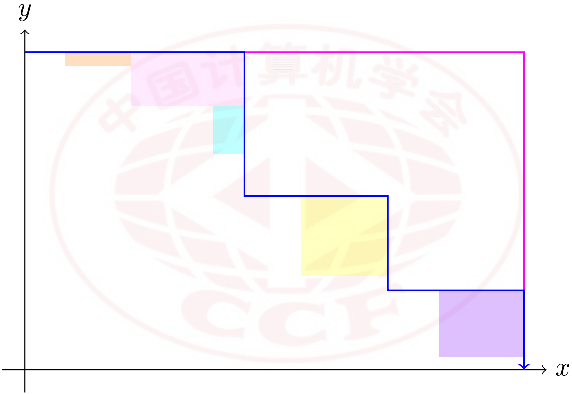
对于一般的情形，找到类似的 1/0 连续段，然后分治处理。



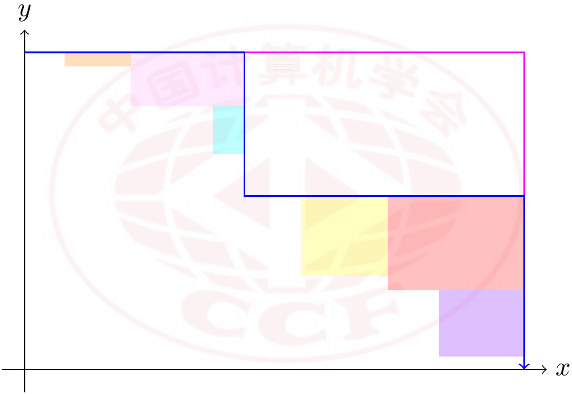
对于一般的情形，找到类似的 1/0 连续段，然后分治处理。



对于一般的情形，找到类似的 1/0 连续段，然后分治处理。

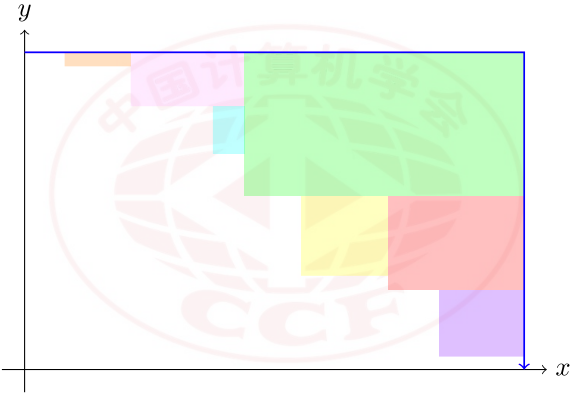


对于一般的情形，找到类似的 1/0 连续段，然后分治处理。

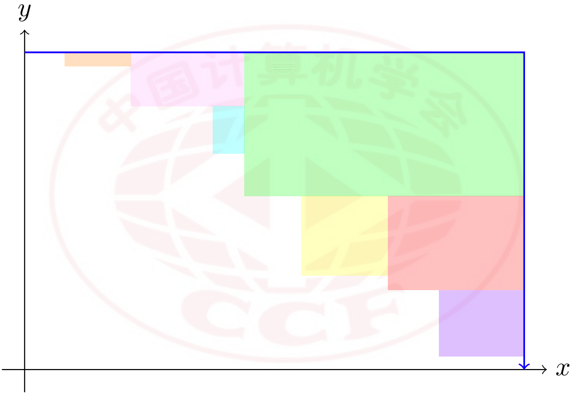


中国计算机学会 China Computer Federation

对于一般的情形，找到类似的 1/0 连续段，然后分治处理。



对于一般的情形，找到类似的 1/0 连续段，然后分治处理。



不难发现，$\sum(r-l)=O(n\log n)$。

给定一个排列 $\boldsymbol{p} = [p_1, p_2, \ldots, p_n]$。对奇偶性不同的整数 l, r $(1 \le l < r \le n)$，定义操作 $S(l, r)$ 为：

- 将 $p_l, p_{l+1}, \ldots, p_r$ 重排为 $p_{l+1}, p_{l+3}, \ldots, p_r, p_l, p_{l+2}, \ldots, p_{r-1}$。

你需要构造一种操作序列将 $\boldsymbol{p}$ 排序。$1 \le n \le 3000$，操作次数不超过 $15\,000$。

给定一个排列 $\boldsymbol{p} = [p_1, p_2, \ldots, p_n]$。对奇偶性不同的整数 l, r $(1 \le l < r \le n)$，定义操作 $S(l, r)$ 为：

- 将 $p_l, p_{l+1}, \ldots, p_r$ 重排为 $p_{l+1}, p_{l+3}, \ldots, p_r, p_l, p_{l+2}, \ldots, p_{r-1}$。

你需要构造一种操作序列将 $\boldsymbol{p}$ 排序。$1 \le n \le 3000$，操作次数不超过 $15\,000$。

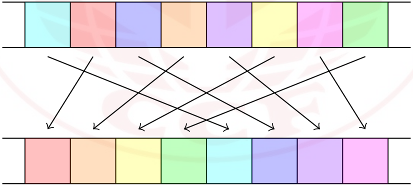


图 6

考虑按照 $1 \sim n$ 的顺序将对应数复位。设我们需要将 v 从位置 p 移到位置 1。

考虑按照 $1 \sim n$ 的顺序将对应数复位。设我们需要将 v 从位置 p 移到位置 1。

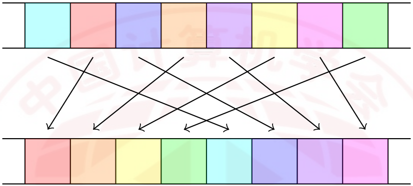
可以通过一次操作将其从位置移到 $\lceil \frac{p}{2} \rceil$。

考虑按照 $1 \sim n$ 的顺序将对应数复位。设我们需要将 v 从位置 p 移到位置 1。

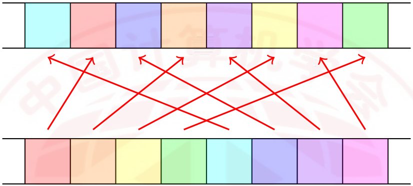
可以通过一次操作将其从位置移到 $\lceil \frac{p}{2} \rceil$。

于是移动第 i 个数的期望步数为 $\frac{1}{n-i+1} \sum_{k=1}^{n-i+1} \log k = O\left(\log(n-i+1)\right)$，总步数为 $\frac{1}{n} \sum_{i=1}^{n} \log i = O(n \log n)$，无法通过。

操作为置换，而置换构成群。

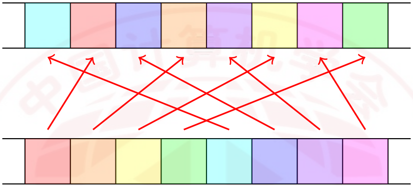


操作为置换，而置换构成群。



中国计算机学会
China Computer Federation

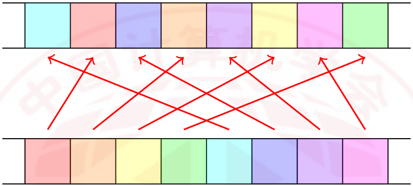
操作为置换，而置换构成群。



从 p 到 1 的步数：$\log p \longrightarrow \log n - \log(n-p)$

中国计算机学会 China Computer Federation

操作为置换，而置换构成群。



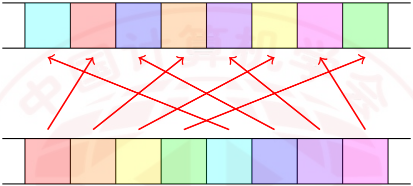
从 p 到 1 的步数：$\log p \longrightarrow \log n - \log(n-p)$

移动第 i 个数的期望步数：

$$\frac{1}{n-i+1}\sum_{k=1}^{n-i+1}\left(\log(n-i+1)-\log(n-i+1-k)\right)=O(1).$$

中国计算机学会 China Computer Federation

操作为置换，而置换构成群。



从 p 到 1 的步数：$\log p \longrightarrow \log n - \log(n-p)$

移动第 i 个数的期望步数：

$$\frac{1}{n-i+1}\sum_{k=1}^{n-i+1}\left(\log(n-i+1)-\log(n-i+1-k)\right)=O(1).$$

通过随机操作随机化初始排列，总步数期望为 $2n+O(1)$。

中国计算机学会 China Computer Federation

称一个由非零整数构成的序列 $[a_1, a_2, \ldots, a_n]$ 是“平衡的”，当且仅当其满足如下三者之一：

- $n = 0$;
- 存在 $1 \le k < n$ 使得 $[a_1, a_2, \ldots, a_k]$ 和 $[a_{k+1}, a_{k+2}, \ldots, a_n]$ 都是“平衡的”;
- $a_1 = -a_n > 0$ 且 $[a_2, a_3, \ldots, a_{n-1}]$ 是“平衡的”。

给定 $2n$ 个二元组 (u_i, v_i)，保证 $\{u_1, u_2, \ldots, u_{2n}\} = \{v_1, v_2, \ldots, v_{2n}\} = \{1, -1, 2, -2, \ldots, n, -n\}$，你需要找到一个大小为 $2n$ 的排列 $p_1, p_2, \ldots, p_{2n}$，使得下列两个条件同时成立：

- $[u_{p_1}, u_{p_2}, \ldots, u_{p_{2n}}]$ 是“平衡的”;
- $[v_{p_1}, v_{p_2}, \ldots, v_{p_{2n}}]$ 是“平衡的”。

给出一组构造，或说明无解。$1 \le n \le 2 \times 10^5$。

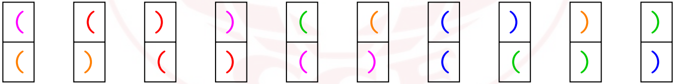
"平衡" 序列的定义非常像括号序列。

"平衡" 序列的定义非常像括号序列。

将 $x > 0$ 看作一种颜色的（，$-x$ 看成对应颜色的）。

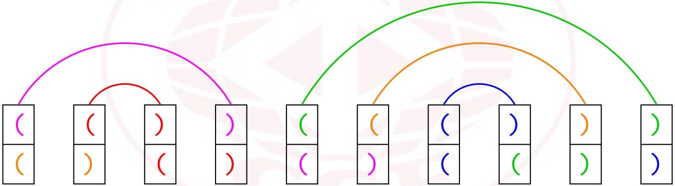
“平衡” 序列的定义非常像括号序列。

将 $x > 0$ 看作一种颜色的（，$-x$ 看成对应颜色的 ）。



"平衡" 序列的定义非常像括号序列。

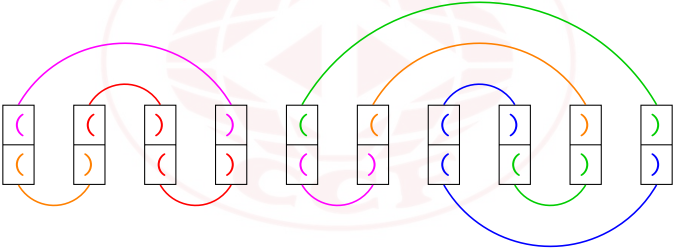
将 $x > 0$ 看作一种颜色的（，$-x$ 看成对应颜色的 ）。



中国计算机学会 China Computer Federation

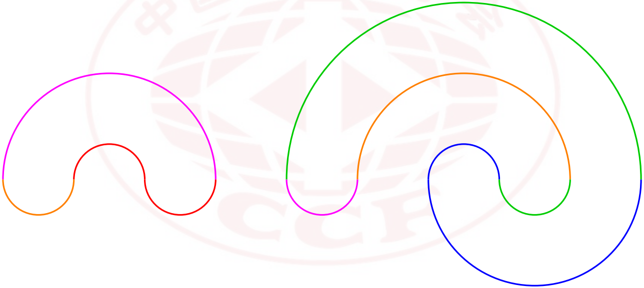
"平衡" 序列的定义非常像括号序列。

将 $x > 0$ 看作一种颜色的（，$-x$ 看成对应颜色的）。



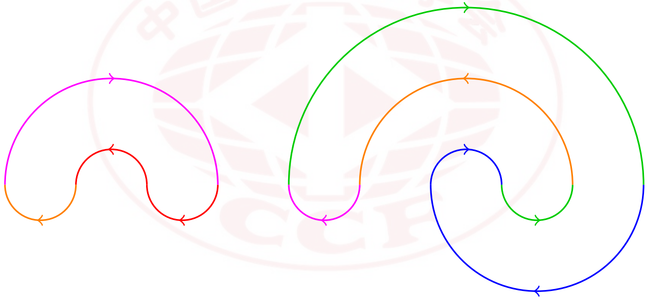
“平衡” 序列的定义非常像括号序列。

将 $x > 0$ 看作一种颜色的（，$-x$ 看成对应颜色的 ）。



“平衡” 序列的定义非常像括号序列。

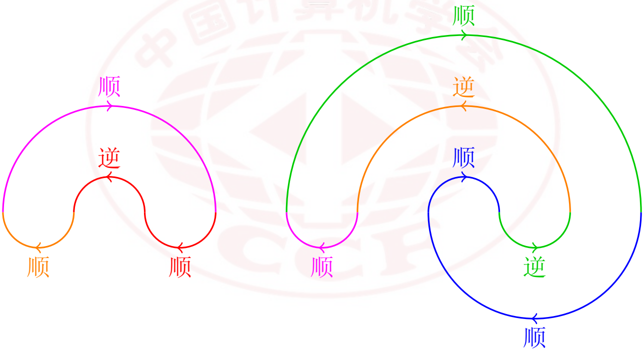
将 $x > 0$ 看作一种颜色的（，$-x$ 看成对应颜色的 ）。



中国计算机学会 China Computer Federation

"平衡" 序列的定义非常像括号序列。

将 $x > 0$ 看作一种颜色的（，$-x$ 看成对应颜色的）。



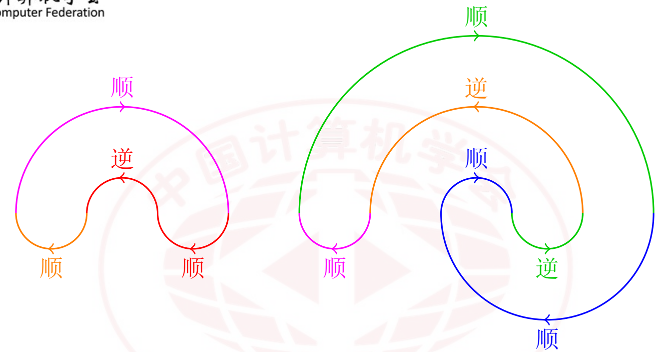


图 7

- 该图的连通分量由输入唯一确定。

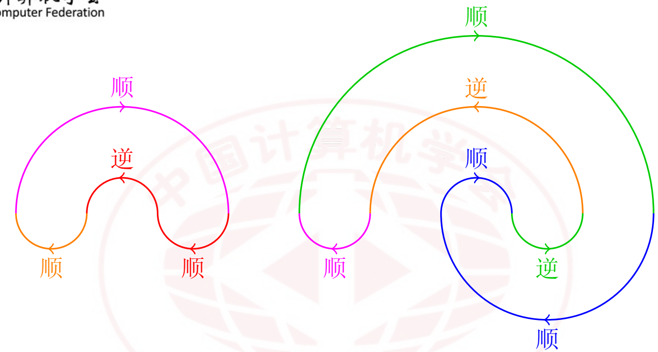


图 7

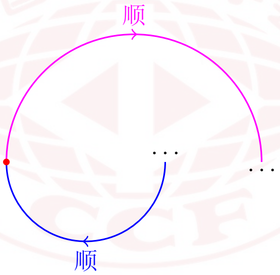
- 该图的连通分量由输入唯一确定。
- 在解中，每个连通分量构成简单闭曲线，故每个圈中顺时针弧的数量比逆时针弧的数量恰好多 2。

对于一个给定的“顺”“逆”环状序列，其中“顺”的数量比“逆”多 2，能否一定完成构造呢？

中国计算机学会 China Computer Federation

对于一个给定的“顺”“逆”环状序列，其中“顺”的数量比“逆”多 2，能否一定完成构造呢？

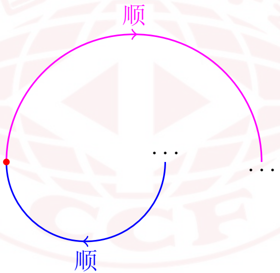
- 必然存在两个相邻的“顺”，令其经过最左侧的点。



中国计算机学会 China Computer Federation

对于一个给定的“顺”“逆”环状序列，其中“顺”的数量比“逆”多 2，能否一定完成构造呢？

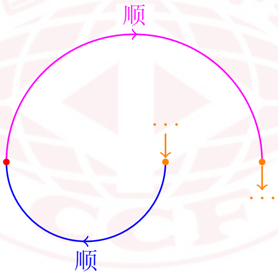
- 必然存在两个相邻的“顺”，令其经过最左侧的点。



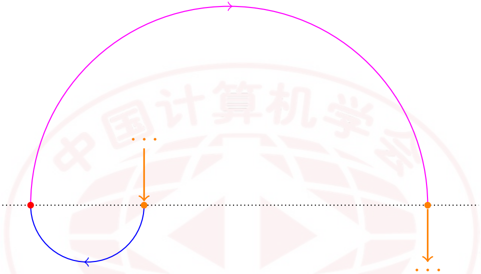
- 剩下的部分中，“顺”的数量与“逆”的数量恰好相同。

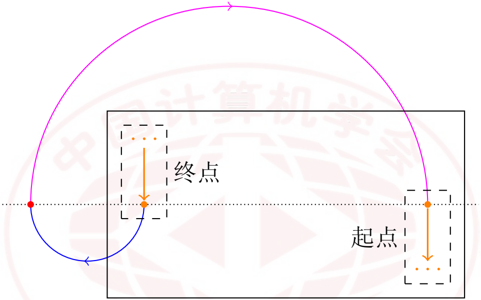
对于一个给定的“顺”“逆”环状序列，其中“顺”的数量比“逆”多 2，能否一定完成构造呢？

- 必然存在两个相邻的“顺”，令其经过最左侧的点。



- 剩下的部分中，“顺”的数量与“逆”的数量恰好相同。





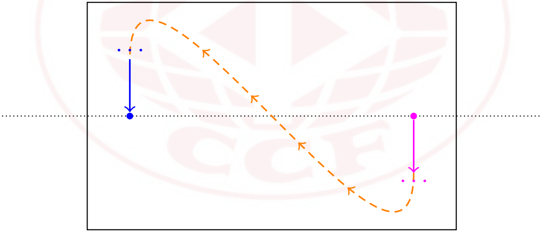
对于一个“顺”“逆”数量相同的序列，可以看成上图中从起点出发向下最后向下回到终点的曲线 (橙色部分)，且起点正上方，终点正下方不能有弧经过。

中国计算机学会 China Computer Federation

记 “顺” 为 $+$，“逆” 为 $-$。令 S 为所有可构造的 $+/-$ 序列的集合。显然 $[] \in S$。

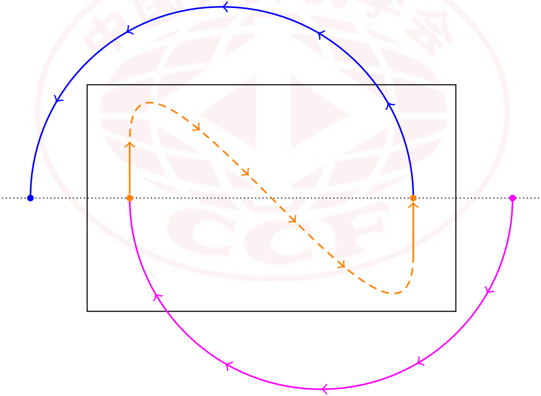
记“顺”为 $+$，“逆”为 $-$。令 S 为所有可构造的 $+/-$ 序列的集合。显然 $[] \in S$。

1. 若 $s \in S$，则 $+s- \in S$。

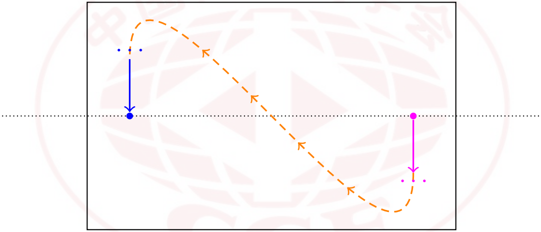


记 “顺” 为 $+$，“逆” 为 $-$。令 S 为所有可构造的 $+/-$ 序列的集合。显然 $[] \in S$。

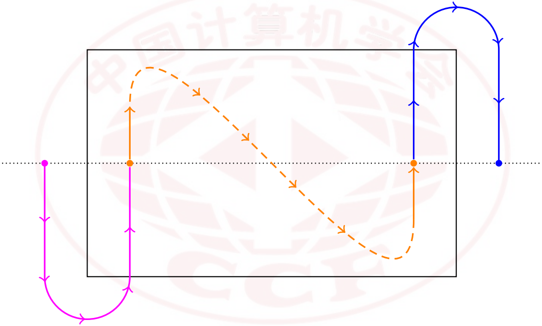
1. 若 $s \in S$，则 $+s- \in S$。



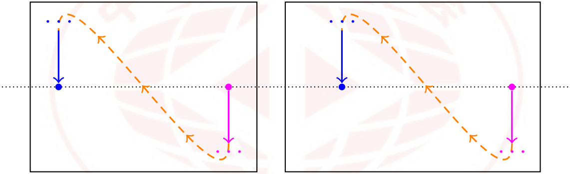
2 若 $s \in S$，则 $-s+ \in S$。



2 若 $s \in S$，则 $-s+ \in S$。

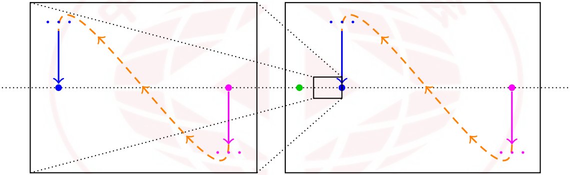


3 若 $s, t \in S$，则 $st \in S$。

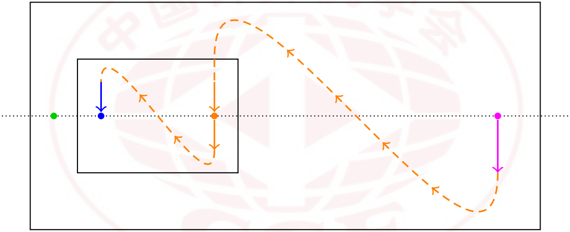


中国计算机学会 China Computer Federation

3 若 $s, t \in S$，则 $st \in S$。

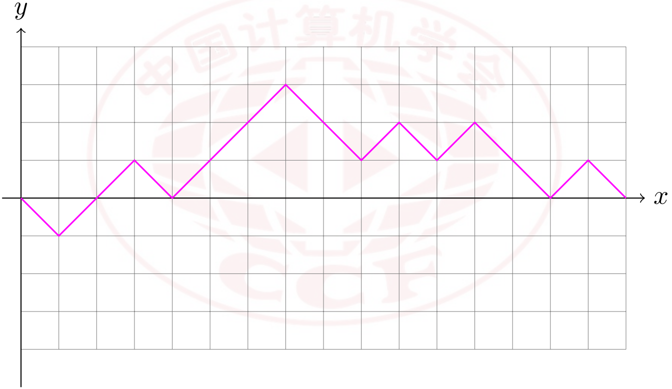


3 若 $s, t \in S$，则 $st \in S$。



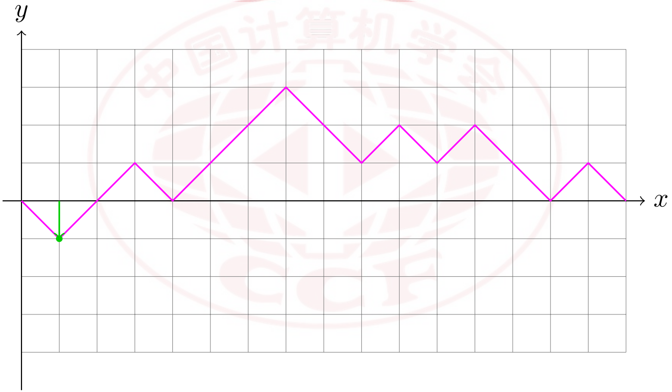
因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

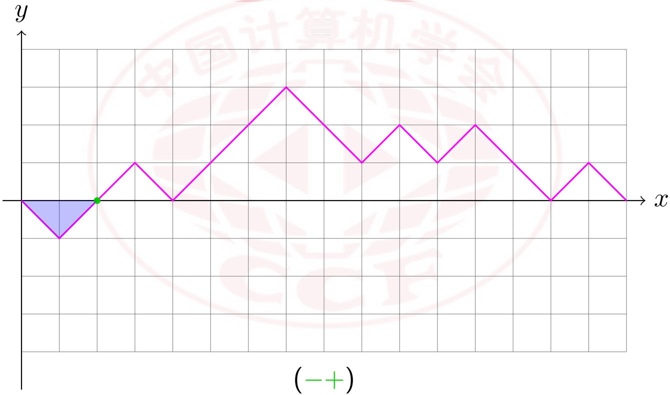


中国计算机学会 China Computer Federation

因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

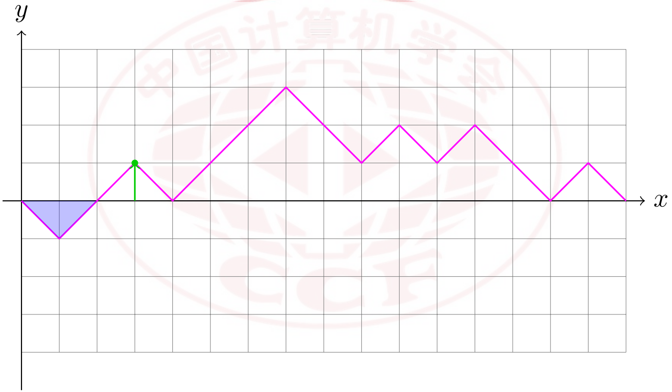


因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

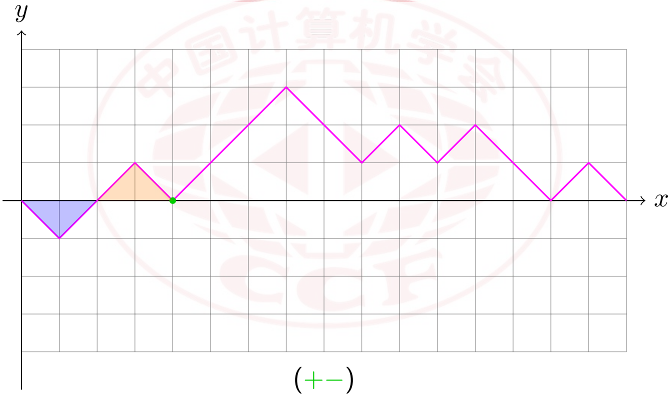


$(-+)$

因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

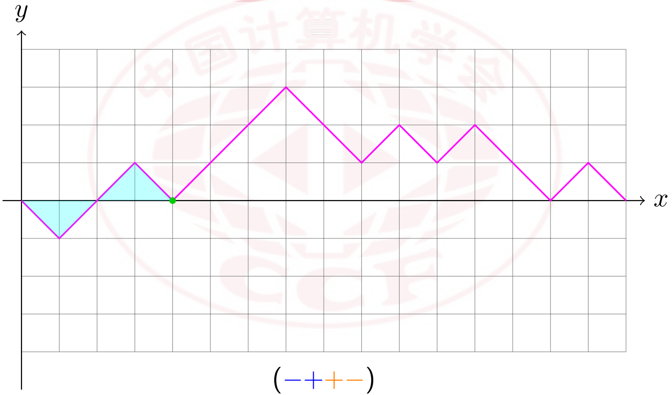


因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。



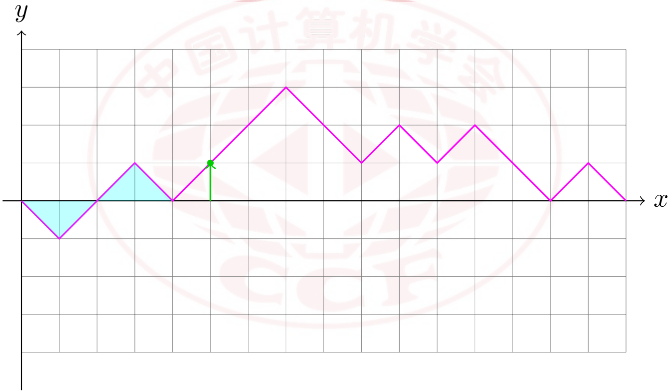
中国计算机学会 China Computer Federation

因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。



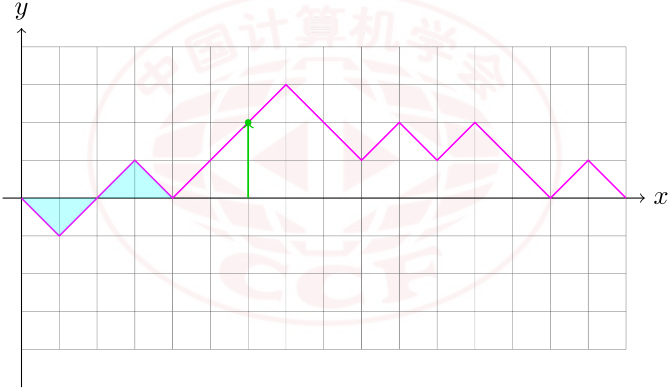
$(-++-)$

因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

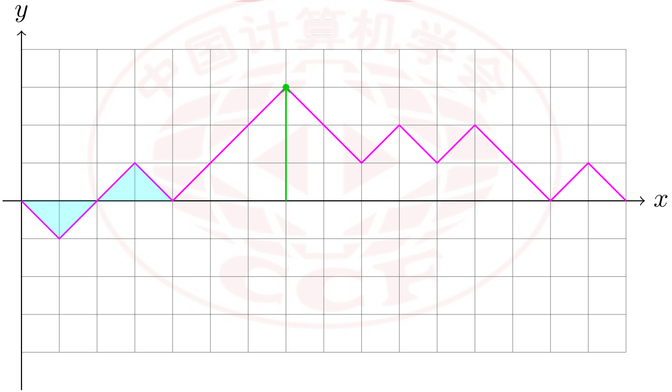


中国计算机学会 China Computer Federation

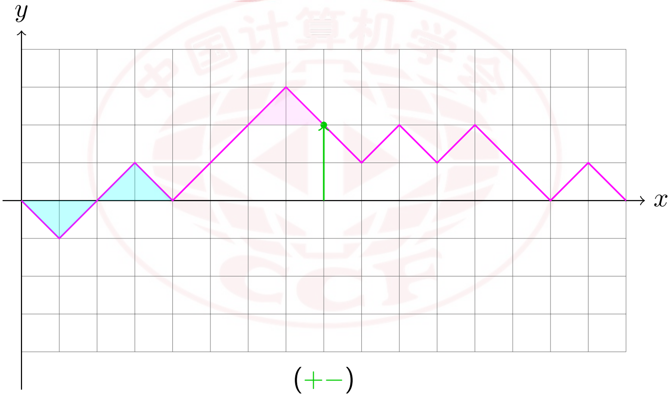
因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。



因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

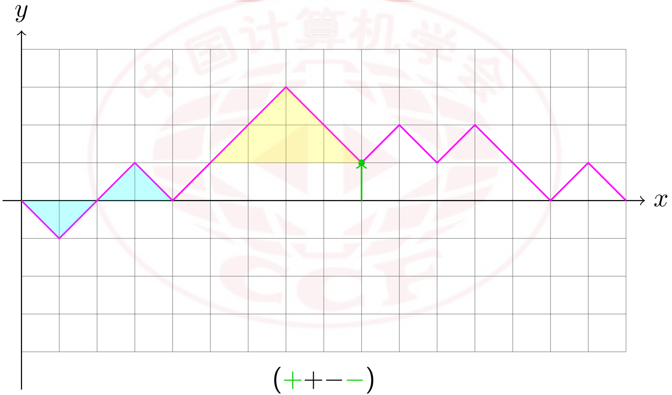


因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

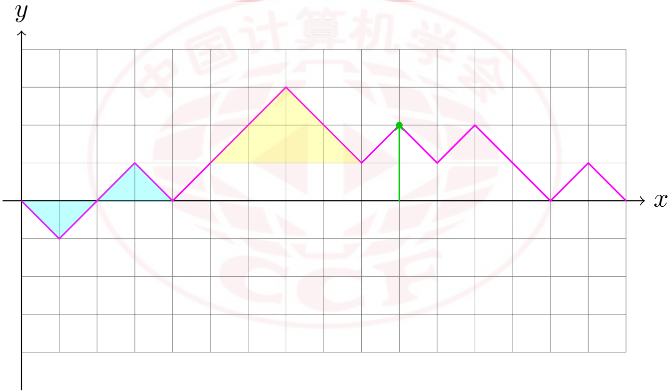


$(+-)$

因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

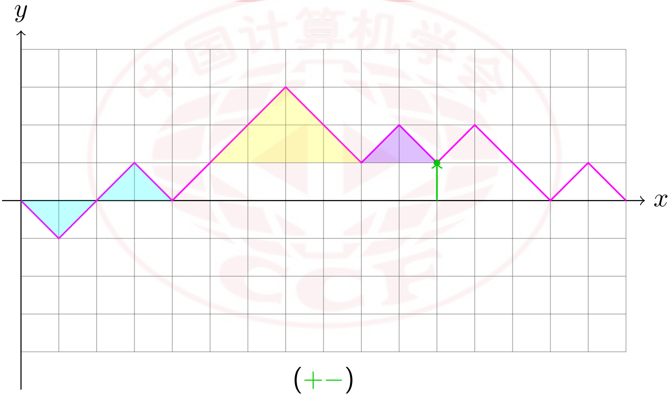


因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。



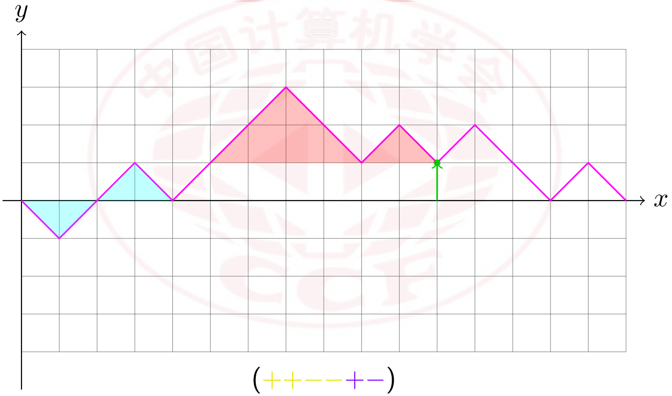
中国计算机学会 China Computer Federation

因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

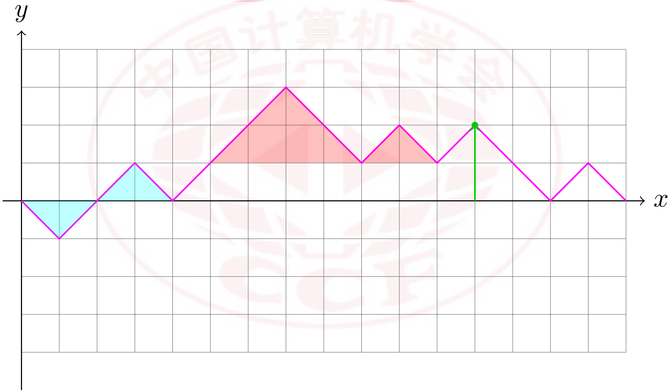


$(+-)$

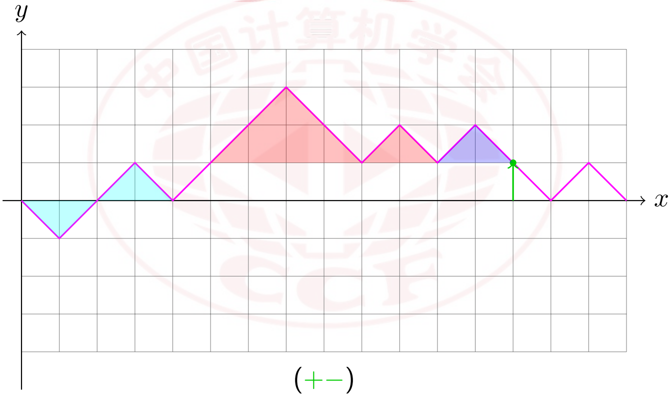
因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。



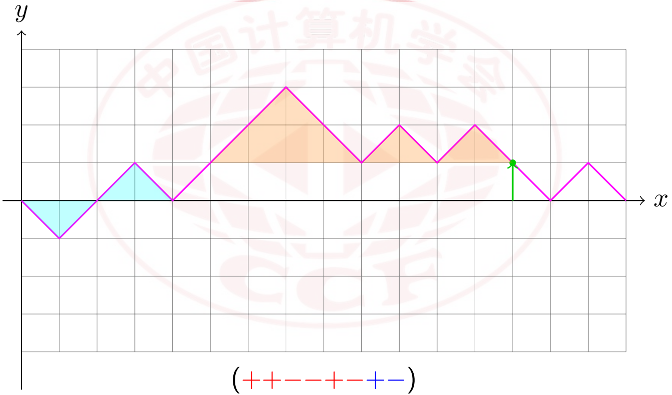
因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。



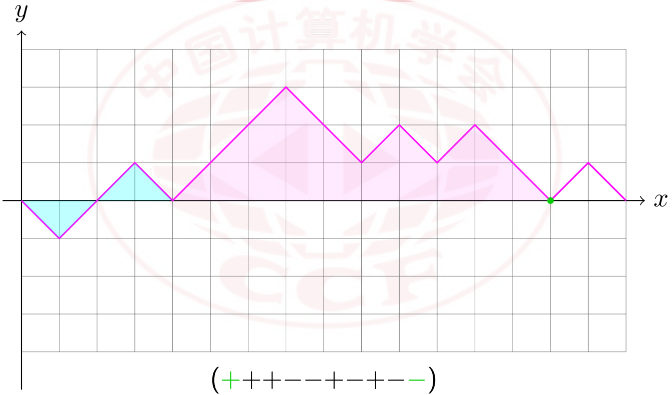
因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。



因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

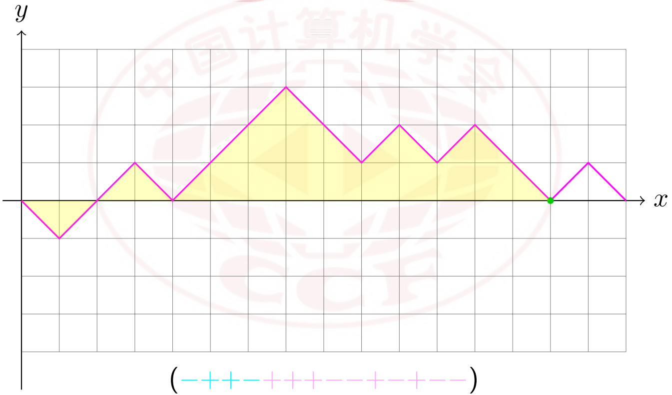


因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

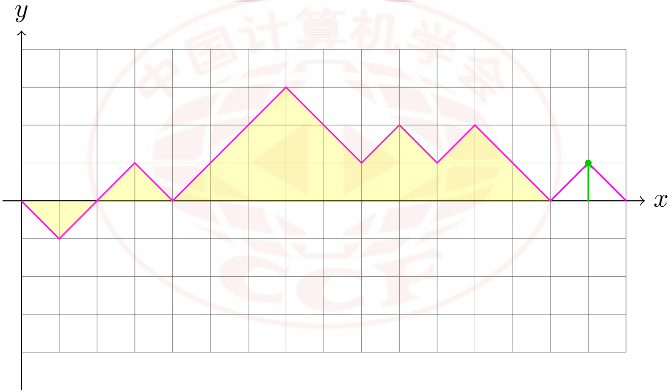


$(+ + + - - + - + - -)$

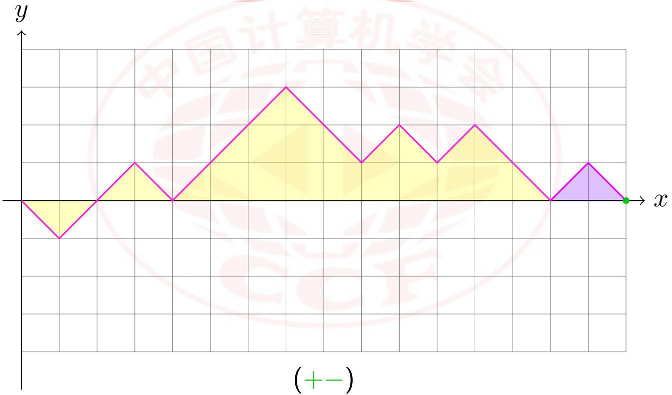
因此 S 为所有 + 和 − 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。



因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。

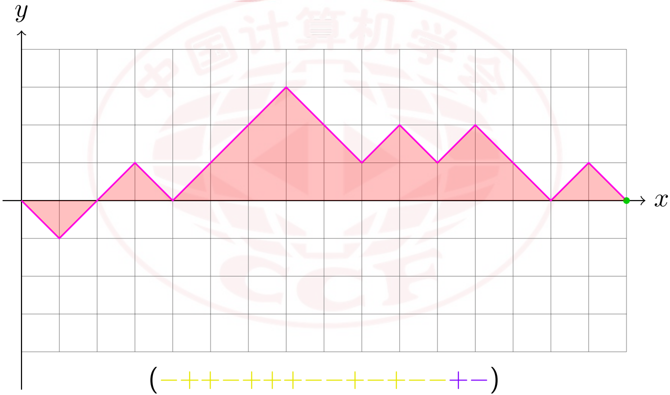


因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。



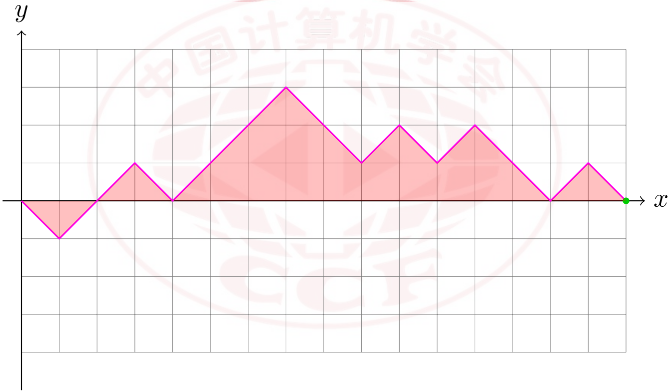
$(+-)$

因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造。



$(-++-+++--+-+--+-)$

因此 S 为所有 $+$ 和 $-$ 数量相同的序列集合。事实上，我们可以通过栈和链表来辅助完成这个构造，时间复杂度 $O(n)$。



中国计算机学会
China Computer Federation

给定带权完全图 $G=(V,E)$，其中 $V=\{1,2,\ldots,n\}$，边 (i,j) 的权值为 $|i-j|$，求 G 的最大生成 (简单) 仙人掌的边权和，并给出一组构造。

$1 \le n \le 5 \times 10^5$。

中国计算机学会
China Computer Federation

Theorem 3.1

存在一组最优解，满足对于每条桥边 (u,v)，均有 $u=1$ 或 $v=n$。

中国计算机学会 China Computer Federation

Theorem 3.1

存在一组最优解，满足对于每条桥边 (u, v)，均有 $u = 1$ 或 $v = n$。

Theorem 3.2

存在一组最优解，不包含长度 ≥ 5 的圈。

中国计算机学会 China Computer Federation

NOI

Theorem 3.1

存在一组最优解，满足对于每条桥边 (u, v)，均有 $u = 1$ 或 $v = n$。

Theorem 3.2

存在一组最优解，不包含长度 ≥ 5 的圈。

Theorem 3.3

存在一组最优解，每个圈都包含点 1 或点 n。

现在允许的构建“材料”：桥边、*3* 元环，*4* 元环。

现在允许的构建“材料”：桥边、*3* 元环，*4* 元环。

- 桥边：桥边只可能形如 $(1, x)$ 或 (x, n)，且当 $x \leq n/2$ 时为后者，否则为前者。

现在允许的构建“材料”：桥边、3 元环，4 元环。

- 桥边：桥边只可能形如 $(1, x)$ 或 (x, n)，且当 $x \le n/2$ 时为后者，否则为前者。
- 3 元环：3 元环 a b c 对边权和提供 $2(c-a)$ 的贡献，和 b 无关。

现在允许的构建“材料”：桥边、3 元环，4 元环。

- 桥边：桥边只可能形如 $(1, x)$ 或 (x, n)，且当 $x \le n/2$ 时为后者，否则为前者。
- 3 元环：3 元环 a b c 对边权和提供 $2(c-a)$ 的贡献，和 b 无关。
- 4 元环：4 元环的摆放形式一定如 a b c d，对边权和可以提供 $2(c+d-a-b)$ 的贡献。

考虑固定使用的圈的个数 C。容易发现桥边的优先级是最低的，因此我们先加入圈。这些圈一定可以按照 $(1,n),(1,n-1),(2,n),(1,n-2),(3,n),\ldots$ 的顺序添加。

考虑固定使用的圈的个数 C。容易发现桥边的优先级是最低的，因此我们先加入圈。这些圈一定可以按照 $(1, n), (1, n-1), (2, n), (1, n-2), (3, n), \ldots$ 的顺序添加。

然后剩下的点先加入到对应的 4 元环提供 $2(r-l)$ 的贡献，然后加入桥边，最后的点留给 3 元环当 b ("工具人")。

考虑固定使用的圈的个数 C。容易发现桥边的优先级是最低的，因此我们先加入圈。这些圈一定可以按照 $(1,n),(1,n-1),(2,n),(1,n-2),(3,n),\ldots$ 的顺序添加。

然后剩下的点先加入到对应的 4 元环提供 $2(r-l)$ 的贡献，然后加入桥边，最后的点留给 3 元环当 b (“工具人”)。

枚举圈的个数 C 即可，时间复杂度 $O(n)$。

考虑固定使用的圈的个数 C。容易发现桥边的优先级是最低的，因此我们先加入圈。这些圈一定可以按照 $(1, n), (1, n-1), (2, n), (1, n-2), (3, n), \ldots$ 的顺序添加。

然后剩下的点先加入到对应的 4 元环提供 $2(r-l)$ 的贡献，然后加入桥边，最后的点留给 3 元环当 b ("工具人")。

枚举圈的个数 C 即可，时间复杂度 $O(n)$。

~~当然如果你打表技术高超可以发现答案及构造以 9 为周期循环的 (~~

中国计算机学会
China Computer Federation

坐标平面上有 n 个互不相同的点 $p_i = (x_i, y_i)$，定义带权完全图 $G = (V, E)$，其中 $V = \{p_1, p_2, \ldots, p_n\}$，两点之间的权值为其 Manhattan 距离。求 G 的最长 *Hamilton* 圈的长度。

$3 \le n \le 10^5$。

先考虑一维的情形，设点的坐标分别为 $x_1 < x_2 < \cdots < x_n$。

中国计算机学会 China Computer Federation

先考虑一维的情形，设点的坐标分别为 $x_1 < x_2 < \cdots < x_n$。

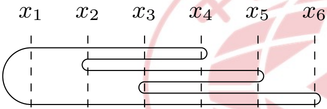


图 8: n 是偶数

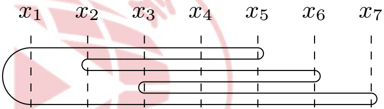


图 9: n 是奇数

先考虑一维的情形，设点的坐标分别为 $x_1 < x_2 < \cdots < x_n$。

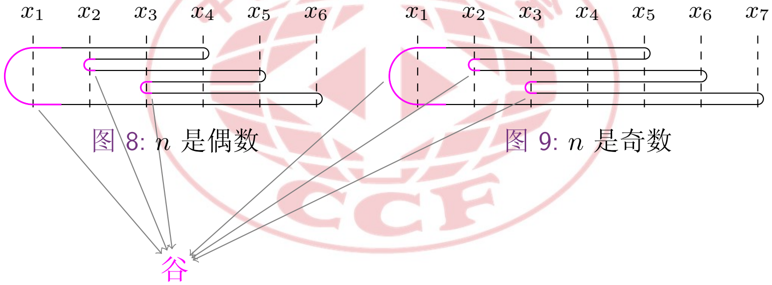


图 8: n 是偶数

图 9: n 是奇数

先考虑一维的情形，设点的坐标分别为 $x_1 < x_2 < \cdots < x_n$。

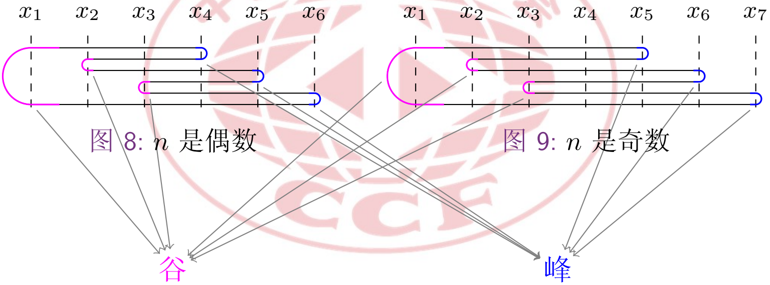


图 8: n 是偶数

图 9: n 是奇数

中国计算机学会 China Computer Federation

先考虑一维的情形，设点的坐标分别为 $x_1 < x_2 < \cdots < x_n$。

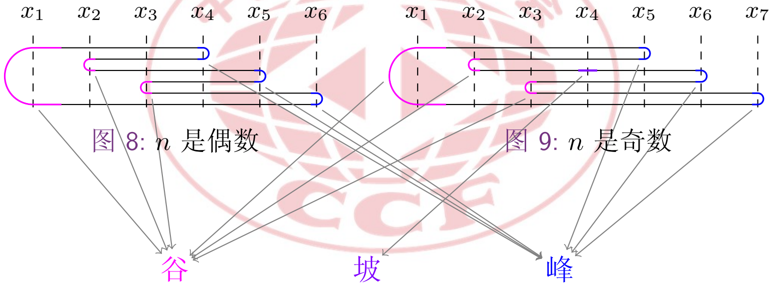


图 8: n 是偶数

图 9: n 是奇数

中国计算机学会 China Computer Federation

先考虑一维的情形，设点的坐标分别为 $x_1 < x_2 < \cdots < x_n$。

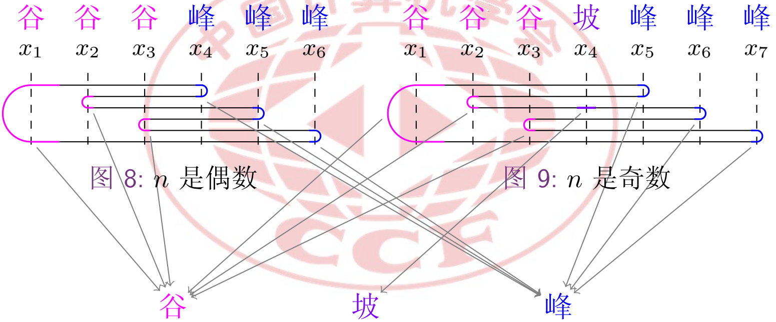


图 8: n 是偶数

图 9: n 是奇数

Theorem 3.4

谷的数量等于峰的数量。

中国计算机学会
China Computer Federation

Theorem 3.4

谷的数量等于峰的数量。

Theorem 3.5

谷峰坡序列可以确定答案。具体地，谷的贡献为 -2，峰的贡献为 2，坡的贡献为 0。

中国计算机学会 China Computer Federation

Theorem 3.6 (最大值的条件)

在一维情形下，最大值取到当且仅当前 $\lfloor \frac{n}{2} \rfloor$ 个元素均为谷，后 $\lfloor \frac{n}{2} \rfloor$ 个元素均为峰。

中国计算机学会 China Computer Federation

Theorem 3.6 (最大值的条件)

在一维情形下，最大值取到当且仅当前 $\lfloor \frac{n}{2} \rfloor$ 个元素均为谷，后 $\lfloor \frac{n}{2} \rfloor$ 个元素均为峰。

Theorem 3.7 (谷峰坡序列的可行性)

一个谷峰坡序列是可行的，当且仅当 1 是谷，n 是峰，且去掉所有的坡后，$2 \sim n-1$ 构成一个合法括号序列，其中视谷为(，视峰为)。

什么时候取次大值?

中国计算机学会 China Computer Federation

什么时候取次大值？

- $n = 2k$ 为偶数：$\underbrace{\text{谷谷}\ldots\text{谷}}_{k-1}$ 坡坡 $\underbrace{\text{峰峰}\ldots\text{峰}}_{k-1}$。

中国计算机学会 China Computer Federation

什么时候取次大值?

- $n=2k$ 为偶数: $\underbrace{\text{谷谷}\ldots\text{谷}}_{k-1}$ 坡坡 $\underbrace{\text{峰峰}\ldots\text{峰}}_{k-1}$。
- $n=2k+1$ 为奇数: 此时有两种情形:

$\underbrace{\text{谷谷}\ldots\text{谷}}_{k}$ 峰坡 $\underbrace{\text{峰峰}\ldots\text{峰}}_{k-1}$ 和 $\underbrace{\text{谷谷}\ldots\text{谷}}_{k-1}$ 坡谷 $\underbrace{\text{峰峰}\ldots\text{峰}}_{k}$。

中国计算机学会 China Computer Federation

什么时候取次大值?

- $n = 2k$ 为偶数:$\underbrace{\text{谷谷}\ldots\text{谷}}_{k-1}$ 坡坡 $\underbrace{\text{峰峰}\ldots\text{峰}}_{k-1}$。
- $n = 2k+1$ 为奇数:此时有两种情形:

$\underbrace{\text{谷谷}\ldots\text{谷}}_{k}$ 峰坡 $\underbrace{\text{峰峰}\ldots\text{峰}}_{k-1}$ 和 $\underbrace{\text{谷谷}\ldots\text{谷}}_{k-1}$ 坡谷 $\underbrace{\text{峰峰}\ldots\text{峰}}_{k}$。

那这个条件有多宽呢?

中国计算机学会 China Computer Federation

Theorem 3.8 (n 为偶数时的次大值可行定理)

设 $n = 2k$ 为偶数。若我们给 $x_1, x_2, \ldots, x_n$ 任意黑白染色 (黑白点各 k 个)，并要求排列必须是黑白交替的形式。不妨设 x_1 是白点，则

- 当所有白点都在黑点左边时，可以取到最大值。
- 否则，一定能取到次大值。

中国计算机学会
China Computer Federation

Theorem 3.9 (n 为奇数时的次大值可行定理)

设 $n = 2k + 1$ 为奇数。若我们给 $x_1, x_2, \ldots, x_n$ 中的其中 $2k$ 个点黑白染色 (黑白点各 k 个)，并要求同色点不能相邻。不妨设 x_1 是白点，则

- 若中间点 x_{k+1} 未被染色，且所有白点都在黑点左边时，可以取到最大值。
- 中间点 x_{k+1} 未被染色的其余情况均可取到任意一种次大值。
- 若中间点 x_{k+1} 被染色，则一定能取到最大值。

中国计算机学会 China Computer Federation

Theorem 3.9 (n 为奇数时的次大值可行定理)

设 $n = 2k + 1$ 为奇数。若我们给 $x_1, x_2, \dots, x_n$ 中的其中 $2k$ 个点黑白染色 (黑白点各 k 个)，并要求同色点不能相邻。不妨设 x_1 是白点，则

- 若中间点 x_{k+1} 未被染色，且所有白点都在黑点左边时，可以取到最大值。
- 中间点 x_{k+1} 未被染色的其余情况均可取到任意一种次大值。
- 若中间点 x_{k+1} 被染色，则一定能取到最大值。

综上，最优方案只可能是下列三者之一：(x 最大, y 最大)、(x 最大, y 次大)、(x 次大, y 最大)。时间复杂度为排序后线性。

考虑完全图 $K_{16} = (V, E)$，其中 $V = \{1, 2, \ldots, 16\}$，你需要将 E 划分成 20 个大小为 6 的子集，使得每个子集是某个 K_4 的 6 条边 (即 K_{16} 的边 K_4 划分)。

进一步，其中某些子集是预先给出的，即你的划分中必须完整地包含这些子集。请给出一组构造或说明无解。

这其实是一个 Steiner 系 $S(2,4,16)$ 问题。先考虑没有预先给出子集时如何直接构造。

中国计算机学会 China Computer Federation

这其实是一个 Steiner 系 $S(2,4,16)$ 问题。先考虑没有预先给出子集时如何直接构造。

考虑有限域 $\mathbb{F}_4=\{0,1,\mathrm{i},\mathrm{i}+1\}$ 上的所有直线 $ax+by+c=0$。由于两点确定一条直线，两条直线不会有多于一个交点。

中国计算机学会
China Computer Federation

这其实是一个 Steiner 系 S$(2,4,16)$ 问题。先考虑没有预先给出子集时如何直接构造。

考虑有限域 $\mathbb{F}_4 = \{0, 1, \mathrm{i}, \mathrm{i}+1\}$ 上的所有直线 $ax+by+c=0$。由于两点确定一条直线，两条直线不会有多于一个交点。

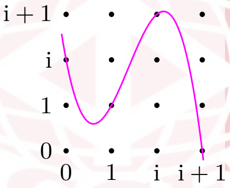


图 10: 直线 $x + \mathrm{i}y + (\mathrm{i}+1) = 0$

这其实是一个 Steiner 系 $\mathrm{S}(2,4,16)$ 问题。先考虑没有预先给出子集时如何直接构造。

考虑有限域 $\mathbb{F}_4=\{0,1,\mathrm{i},\mathrm{i}+1\}$ 上的所有直线 $ax+by+c=0$。由于两点确定一条直线，两条直线不会有多于一个交点。

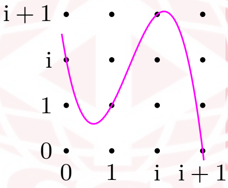


图 10: 直线 $x+\mathrm{i}y+(\mathrm{i}+1)=0$

直线斜率一共有 $0,1,\mathrm{i},\mathrm{i}+1,\infty$ 共 5 种，每种直线有 4 条，这样就是一个可行的构造。

中国计算机学会 China Computer Federation

Theorem 3.10

所有的 *Steiner* 系 $S(2,4,16)$ 都是同构的。

中国计算机学会 China Computer Federation

Theorem 3.10

所有的 Steiner 系 $S(2,4,16)$ 都是同构的。

于是任意一组解 (如果存在的话)，一定可以通过仿射变换变成标准解。于是任意找两个有公共点的边子集，规定其中一条为 x 轴 $(y=0)$，其中一条为 y 轴 $(x=0)$，剩余情况枚举排列验证。

中国计算机学会 China Computer Federation

Theorem 3.10

所有的 *Steiner* 系 $\mathsf{S}(2,4,16)$ 都是同构的。

于是任意一组解 (如果存在的话)，一定可以通过仿射变换变成标准解。于是任意找两个有公共点的边子集，规定其中一条为 x 轴 $(y=0)$，其中一条为 y 轴 $(x=0)$，剩余情况枚举排列验证。

时间复杂度 $3!\cdot 9!\cdot$ 验证时间。

中国计算机学会 China Computer Federation

给定图 $G=(V,E)$，其中 $V=\mathbb{N}, E=\{(i,j) \mid i,j\in\mathbb{N}$ 且 $\gcd(i,j)>1\}$。给定 N，求是否存在 $X\in\mathbb{N}$ 满足 $S_X=\{X,X+1,\ldots,X+N-1\}$ 导出的子图 $G[S_X]$ 连通，若是，给出满足条件的 X。

$1\le N\le 10^5$。

不妨设 N 足够大。若取 $M = \prod_{p \le n, p \in \mathbb{P}} p$，则对于任意 $x \in [-n, n] \setminus \{\pm 1\}$，$M$ 和 $M + x$ 之间连有边。

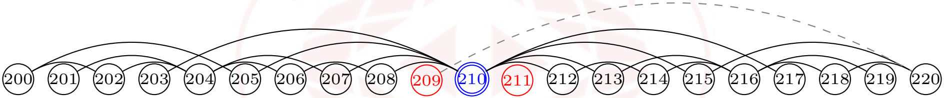


图 11

不妨设 N 足够大。若取 $M = \prod_{p\le n, p\in\mathbb{P}} p$，则对于任意 $x \in [-n, n] \setminus \{\pm 1\}$，$M$ 和 $M + x$ 之间连有边。

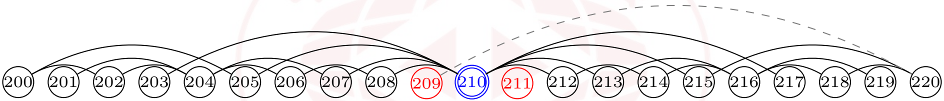
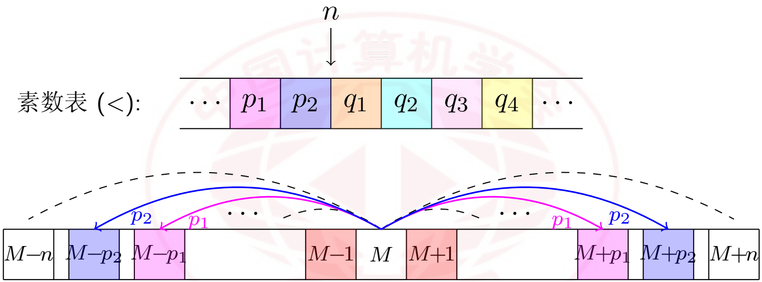


图 11

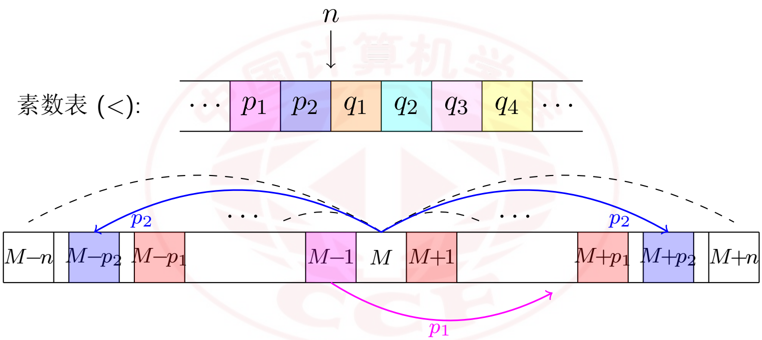
也就是说，这 $2n + 1$ 个点几乎是连通的——除了 $M \pm 1$。

如何将 $M \pm 1$ 连上？“借用”小素数。



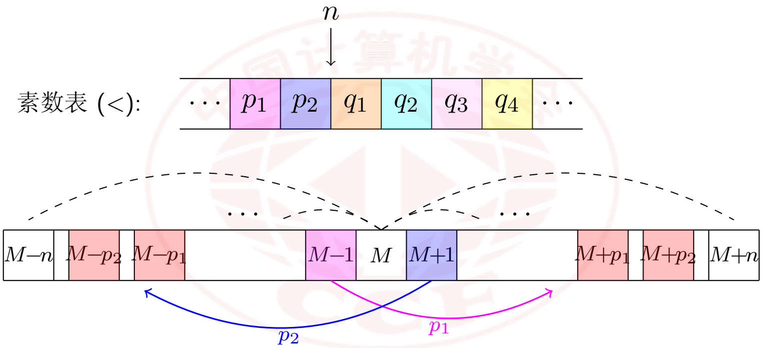
中国计算机学会 China Computer Federation

如何将 $M \pm 1$ 连上？“借用”小素数。



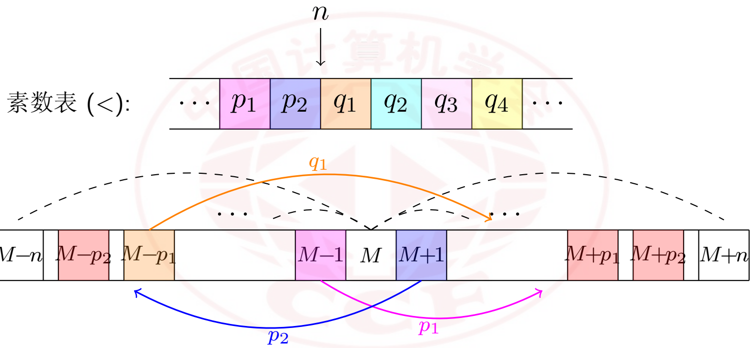
中国计算机学会 China Computer Federation

如何将 $M \pm 1$ 连上？“借用”小素数。



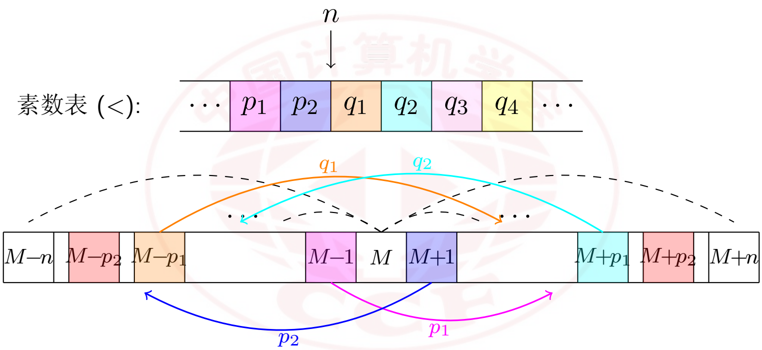
中国计算机学会 China Computer Federation

如何将 $M \pm 1$ 连上？“借用”小素数。



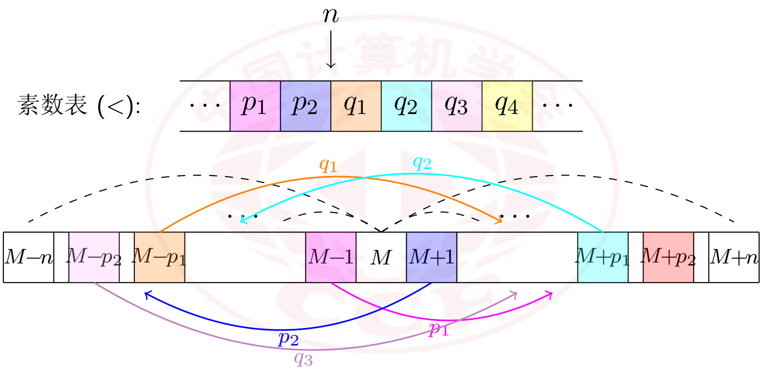
中国计算机学会 China Computer Federation

如何将 $M \pm 1$ 连上？“借用”小素数。

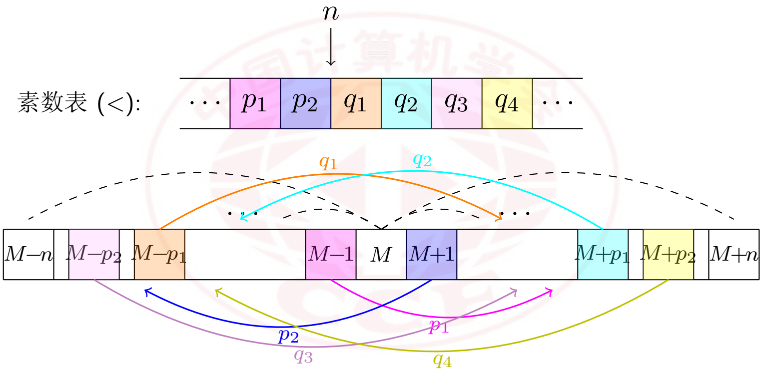


中国计算机学会 China Computer Federation

如何将 $M \pm 1$ 连上？“借用”小素数。

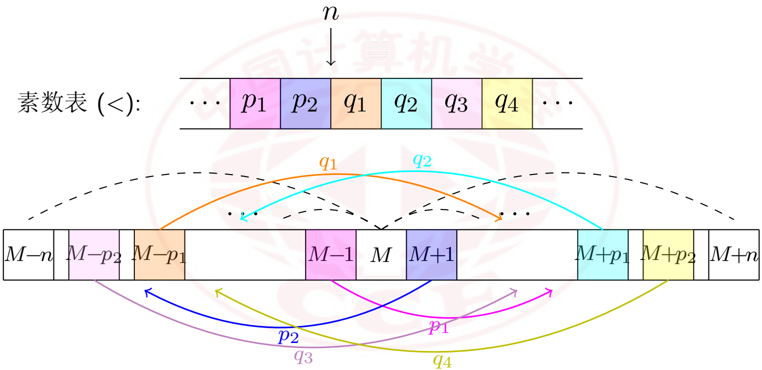


如何将 $M \pm 1$ 连上？“借用”小素数。



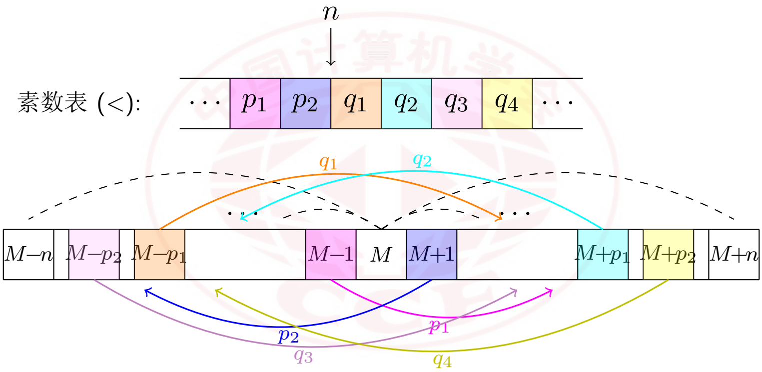
中国计算机学会 China Computer Federation

如何将 $M \pm 1$ 连上？“借用”小素数。



只需 $q_2 \le p_1 + n$，$q_4 \le p_2 + n$。

如何将 $M \pm 1$ 连上？“借用”小素数。



只需 $q_2 \le p_1 + n$，$q_4 \le p_2 + n \Leftarrow n \ge 19$。

当 $N \geq 39$ 时，

$$\begin{cases} M \equiv 1 & \pmod{p_1} \\ M \equiv -1 & \pmod{p_2} \\ M \equiv p_1 & \pmod{q_1} \\ M \equiv -p_1 & \pmod{q_2} \\ M \equiv p_2 & \pmod{q_3} \\ M \equiv -p_2 & \pmod{q_4} \end{cases}$$

中国计算机学会 China Computer Federation

当 $N \geq 39$ 时，

$$\begin{cases} M \equiv 1 & \pmod{p_1} \\ M \equiv -1 & \pmod{p_2} \\ M \equiv p_1 & \pmod{q_1} \\ M \equiv -p_1 & \pmod{q_2} \\ M \equiv p_2 & \pmod{q_3} \\ M \equiv -p_2 & \pmod{q_4} \end{cases}$$

对小于 p_1 的素数 p，我们要求 $p \mid M$，使用 CRT 即可找到符合要求的 M。

中国计算机学会 China Computer Federation

当 $N \geq 39$ 时，

$$\begin{cases} M \equiv 1 & \pmod{p_1} \\ M \equiv -1 & \pmod{p_2} \\ M \equiv p_1 & \pmod{q_1} \\ M \equiv -p_1 & \pmod{q_2} \\ M \equiv p_2 & \pmod{q_3} \\ M \equiv -p_2 & \pmod{q_4} \end{cases}$$

对小于 p_1 的素数 p，我们要求 $p \mid M$，使用 CRT 即可找到符合要求的 M。

当 $17 \leq N < 39$ 时，暴力枚举，最小解不超过 10^7。否则无解。

这是一道通信题。

给定一棵 N 个顶点的有根树，保证所有顶点的深度 (到根的距离) 不超过 18。Alice 需要给每个顶点 v 进行编码 c_v，需要满足 $0 \le c_v \le 2^{28} - 1$。

Bob 不知道关于这棵树的任何信息，而 Bob 需要回答若干次询问：每次询问 Bob 将收到顶点 x, y $(x \ne y)$ 的编码 c_x, c_y，然后 Bob 需要判断以下三种情况哪一种成立：

0 y 是 x 的祖先。
1 x 是 y 的祖先。
2 x, y 不存在祖先关系。

$2 \le N \le 2.5 \times 10^5$。

中国计算机学会 China Computer Federation

如何判定祖先关系？dfs 序。

如何判定祖先关系？dfs 序。

设顶点 v 的 dfs 时间戳为 i_v (即 v 是 dfs 序中的第 i_v 个顶点)，以 v 为根的子树中时间戳最大者为 e_v (即搜完这棵子树时的总点数)，则

$$u \text{ 是 } v \text{ 的祖先} \iff i_u \le i_v \le e_u.$$

中国计算机学会 China Computer Federation

如何判定祖先关系？dfs 序。

设顶点 v 的 dfs 时间戳为 i_v(即 v 是 dfs 序中的第 i_v 个顶点)，以 v 为根的子树中时间戳最大者为 e_v (即搜完这棵子树时的总点数)，则

$$u \text{ 是 } v \text{ 的祖先} \iff i_u \le i_v \le e_u.$$

令 $c_v = (i_v, e_v)$ 即可完成，但是传输信息量为 $2\log_2 N \approx 36\ \mathrm{bit} > 28\ \mathrm{bit}$。

$$i_u \le i_v \le e_u \iff 0 \le i_v - i_u \le e_u - i_u。$$

$$i_u \le i_v \le e_u \iff 0 \le i_v - i_u \le e_u - i_u。$$

$$\Downarrow$$

$$size_u - 1$$

中国计算机学会 China Computer Federation

$$i_u \le i_v \le e_u \iff 0 \le i_v - i_u \le e_u - i_u 。$$

$$\Downarrow$$

$$0 \le i_v - i_u \le size_u - 1$$

中国计算机学会
China Computer Federation

$$i_u \le i_v \le e_u \iff 0 \le i_v - i_u \le e_u - i_u。$$

$$\Downarrow$$

$$0 \le i_v - i_u \le size_u - 1$$

令 $c_v = (i_v, size_v)$，同样可以完成目的，但最坏情况下传输信息量仍为 $2\log_2 N$。

$$i_u \le i_v \le e_u \iff 0 \le i_v - i_u \le e_u - i_u\text{。}$$
$$\Downarrow$$
$$0 \le i_v - i_u \le size_u - 1$$

令 $c_v = (i_v, size_v)$，同样可以完成目的，但最坏情况下传输信息量仍为 $2\log_2 N$。

保证所有顶点的深度不超过 18。

中国计算机学会 China Computer Federation

$$i_u \le i_v \le e_u \iff 0 \le i_v - i_u \le e_u - i_u。$$

$$\Downarrow$$

$$0 \le i_v - i_u \le size_u - 1$$

令 $c_v = (i_v, size_v)$，同样可以完成目的，但最坏情况下传输信息量仍为 $2\log_2 N$。

保证所有顶点的深度不超过 18。$\sum_v dep_v \le 18N$。

$$i_u \le i_v \le e_u \iff 0 \le i_v - i_u \le e_u - i_u。$$

$$\Downarrow$$

$$0 \le i_v - i_u \le size_u - 1$$

令 $c_v = (i_v, size_v)$，同样可以完成目的，但最坏情况下传输信息量仍为 $2\log_2 N$。

保证所有顶点的深度不超过 18。$\sum_v dep_v \le 18N$。

$$\sum_v size_v = \sum_v (dep_v + 1)$$

$$i_u \le i_v \le e_u \iff 0 \le i_v - i_u \le e_u - i_u。$$

$$\Downarrow$$

$$0 \le i_v - i_u \le size_u - 1$$

令 $c_v = (i_v, size_v)$，同样可以完成目的，但最坏情况下传输信息量仍为 $2\log_2 N$。

保证所有顶点的深度不超过 18。$\sum_v dep_v \le 18N$。

$$\sum_v size_v = \sum_v (dep_v + 1) \le 19N = O(N)！$$

中国计算机学会 China Computer Federation

$$i_u \le i_v \le e_u \iff 0 \le i_v - i_u \le e_u - i_u。$$

$$\Downarrow$$

$$0 \le i_v - i_u \le size_u - 1$$

令 $c_v = (i_v, size_v)$，同样可以完成目的，但最坏情况下传输信息量仍为 $2\log_2 N$。

保证所有顶点的深度不超过 18。$\sum_v dep_v \le 18N$。

$$\sum_v size_v = \sum_v (dep_v + 1) \le 19N = O(N)！$$

$size_v$ 的种数不会很多。然而 Bob 无法得知是哪些值。

中国计算机学会 China Computer Federation

$$i_u \le i_v \le e_u \iff 0 \le i_v - i_u \le e_u - i_u \text{。}$$

$$\Downarrow$$

$$0 \le i_v - i_u \le size_u - 1$$

令 $c_v = (i_v, size_v)$，同样可以完成目的，但最坏情况下传输信息量仍为 $2\log_2 N$。

保证所有顶点的深度不超过 18。$\sum_v dep_v \le 18N$。

$$\sum_v size_v = \sum_v (dep_v + 1) \le 19N = O(N)\ !$$

$size_v$ 的种数不会很多。然而 Bob 无法得知是哪些值。

通过加入虚点固定 $size_v$ 的可能取值！

常见的技巧：向等比数列拟合。

常见的技巧：向等比数列拟合。

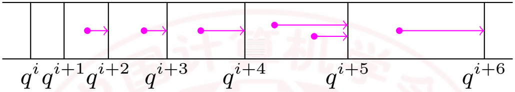


图 12

考虑等比数列 $\{q^n\} = \{1, q, q^2, q^3, \ldots\}$ $(q > 1)$，然后将每个 $size_v$ 取到不小于 $size_v$ 的最小 q^k。

常见的技巧：向等比数列拟合。

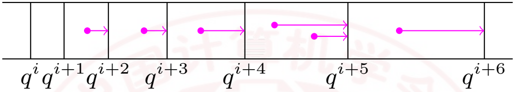


图 12

考虑等比数列 $\{q^n\} = \{1, q, q^2, q^3, \ldots\}$ $(q > 1)$，然后将每个 $size_v$ 取到不小于 $size_v$ 的最小 q^k。

分析可知，最终的总点数不会超过

$$\sum_v q^{dep_v} \leq q^{18} N.$$

经如此量化后，不同的 $size_v$ 的可能取值约为 $\log_q(q^{18}N)$ 种。

经如此量化后，不同的 $size_v$ 的可能取值约为 $\log_q(q^{18}N)$ 种。
最终的传输信息量不超过

$$\left\lceil \log_2\left(\log_q\left(q^{18}N\right)\right)\right\rceil + \left\lceil \log_2\left(q^{18}N\right)\right\rceil \ \text{ bit},$$

经如此量化后，不同的 $size_v$ 的可能取值约为 $\log_q(q^{18}N)$ 种。
最终的传输信息量不超过

$$\left\lceil \log_2 \left(\log_q \left(q^{18} N \right) \right) \right\rceil + \left\lceil \log_2 \left(q^{18} N \right) \right\rceil \ \text{bit},$$

取 $q = 1.06$ 可知其不超过 28。

坐标平面上有 n 个互不相同的点 (x_i, y_i)，你需要在每个点上写一个 $1 \sim n$ 之间的整数，并保证这些整数互不相同。

定义一条直线为 "好的"，如果它满足如下三个条件：

- 平行于坐标轴 (x 轴或 y 轴)；
- 经过至少一个 (给定的) 点；
- 经过的所有点上所写的数的最大公约数为 1。

求 "好的" 直线的数量的最大值，并给出一组构造。

$1 \leq n \leq 2 \times 10^5$。

建立二分图模型，离散化坐标后，对每一条 (经过点的) 水平线 ($x = x_0$) 及竖直线 ($y = y_0$) 建立一个顶点，然后对于题目中的点 (x_i, y_i)，我们可以将其看成连接 x_i 和 y_i 的一条权值为所写的数的边，从而得到二分图 $G = (V, E)$，其中 $|E| = n$。

建立二分图模型，离散化坐标后，对每一条 (经过点的) 水平线 $(x = x_0)$ 及竖直线 $(y = y_0)$ 建立一个顶点，然后对于题目中的点 (x_i, y_i)，我们可以将其看成连接 x_i 和 y_i 的一条权值为所写的数的边，从而得到二分图 $G = (V, E)$，其中 $|E| = n$。

一条直线是 “好的”，当且仅当其在 G 中对应的点 v，所关联的所有边的权值的 gcd 为 1。称这样的点为*互素*的。

建立二分图模型，离散化坐标后，对每一条 (经过点的) 水平线 ($x = x_0$) 及竖直线 ($y = y_0$) 建立一个顶点，然后对于题目中的点 (x_i, y_i)，我们可以将其看成连接 x_i 和 y_i 的一条权值为所写的数的边，从而得到二分图 $G = (V, E)$，其中 $|E| = n$。

一条直线是 "好的"，当且仅当其在 G 中对应的点 v，所关联的所有边的权值的 gcd 为 1。称这样的点为*互素*的。

问题转化为最大化*互素*顶点的个数。

考虑答案的上界。

考虑答案的上界。

如果一个顶点 v 的度数 $d(v)=1$，那么 v 是互素的当且仅当与之关联的唯一边权值为 1。

中国计算机学会 China Computer Federation

考虑答案的上界。

如果一个顶点 v 的度数 $d(v)=1$，那么 v 是互素的当且仅当与之关联的唯一边权值为 1。

因此，设 G 中有 k 个 1 度点，则：

中国计算机学会 China Computer Federation

考虑答案的上界。

如果一个顶点 v 的度数 $d(v) = 1$，那么 v 是互素的当且仅当与之关联的唯一边权值为 1。

因此，设 G 中有 k 个 1 度点，则：

- 若存在两个 1 度点之间有边相连，则答案不超过 $|V| - k + 2$。

考虑答案的上界。

如果一个顶点 v 的度数 $d(v)=1$，那么 v 是互素的当且仅当与之关联的唯一边权值为 1。

因此，设 G 中有 k 个 1 度点，则：

- 若存在两个 1 度点之间有边相连，则答案不超过 $|V|-k+2$。
- 否则，答案不超过 $|V|-\max\{|k|-1,0\}$。

考虑答案的上界。

如果一个顶点 v 的度数 $d(v)=1$，那么 v 是互素的当且仅当与之关联的唯一边权值为 1。

因此，设 G 中有 k 个 1 度点，则：

- 若存在两个 1 度点之间有边相连，则答案不超过 $|V|-k+2$。
- 否则，答案不超过 $|V|-\max\{|k|-1,0\}$。

实践表明，这个上界很可能就是紧的，因此下面考虑构造。

如何构造互素的整数？

如何构造互素的整数？相邻的正整数一定互素！

如何构造互素的整数？相邻的正整数一定互素！

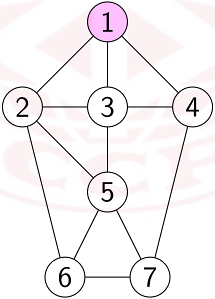
希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径 “经过” 这个点)。

中国计算机学会 China Computer Federation

如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径“经过”这个点)。

对于 G 是连通图的情形，只要贪心取链即可：

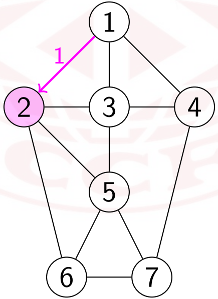


中国计算机学会 China Computer Federation

如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径 “经过” 这个点)。

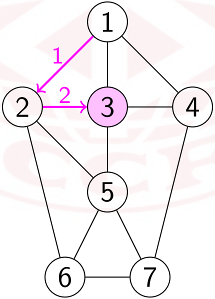
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径 “经过” 这个点)。

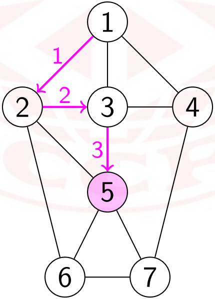
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径 “经过” 这个点)。

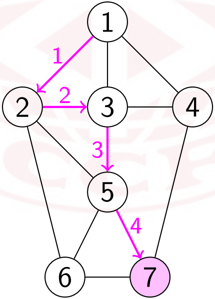
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径 “经过” 这个点)。

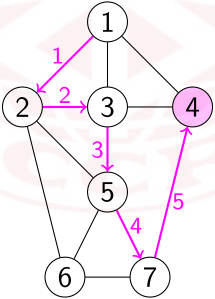
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径 “经过” 这个点)。

对于 G 是连通图的情形，只要贪心取链即可：

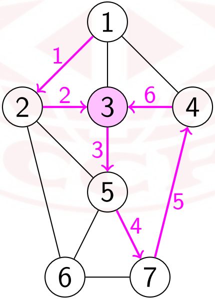


中国计算机学会 China Computer Federation

如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径 “经过” 这个点)。

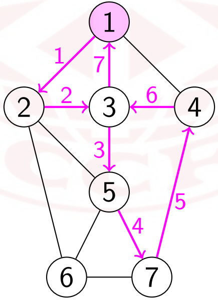
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径"经过"这个点)。

对于 G 是连通图的情形，只要贪心取链即可：

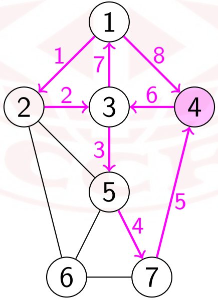


中国计算机学会 China Computer Federation

如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径 “经过” 这个点)。

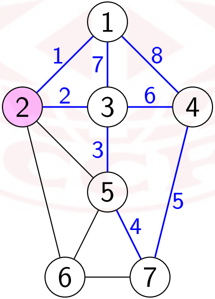
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径“经过”这个点)。

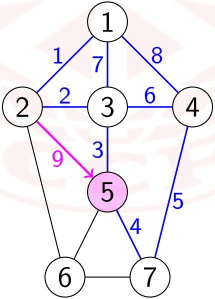
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径“经过”这个点)。

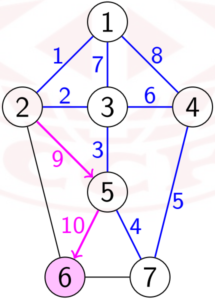
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径 “经过” 这个点)。

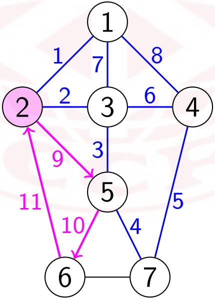
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径 “经过” 这个点)。

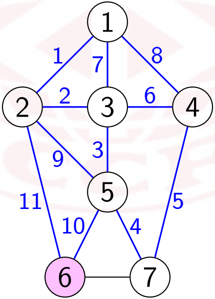
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径“经过”这个点)。

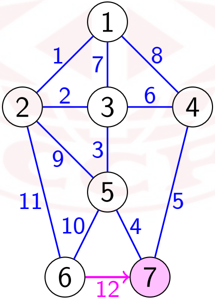
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径“经过”这个点)。

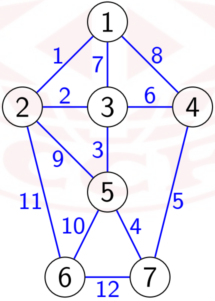
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径 “经过” 这个点)。

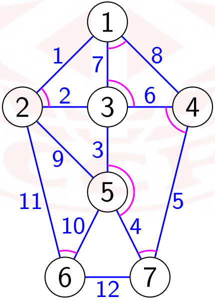
对于 G 是连通图的情形，只要贪心取链即可：



如何构造互素的整数？相邻的正整数一定互素！

希望对于每个度数 ≥ 2 的点，总有两条出边是相邻的 (即作为路径 “经过” 这个点)。

对于 G 是连通图的情形，只要贪心取链即可：



当 G 不再连通时，这个算法会出什么问题呢？

中国计算机学会 China Computer Federation

当 G 不再连通时，这个算法会出什么问题呢？

唯一的问题在于，当我们对其它连通分量贪心时，第一条边的权值不再是 1，这将导致最初的顶点 v 不一定互素。

当 G 不再连通时，这个算法会出什么问题呢？

唯一的问题在于，当我们对其它连通分量贪心时，第一条边的权值不再是 1，这将导致最初的顶点 v 不一定互素。

但就连通分量而言，当固定区间后，有的连通分量本身就无法完成：如 $3, 4, 5, 6$ 就无法分配给一个长度为 4 的圈使得所有点都互素，因为与 6 互素的只有 5。

当 G 不再连通时，这个算法会出什么问题呢？

唯一的问题在于，当我们对其它连通分量贪心时，第一条边的权值不再是 1，这将导致最初的顶点 v 不一定互素。

但就连通分量而言，当固定区间后，有的连通分量本身就无法完成：如 $3, 4, 5, 6$ 就无法分配给一个长度为 4 的圈使得所有点都互素，因为与 6 互素的只有 5。

不过，对于非圈的连通分量 G，我们只需简单修改一下该算法。

- 如果连通分量存在 1 度点，那么显然从它开始。

- 如果连通分量存在 1 度点，那么显然从它开始。
- 否则，该连通分量中一定存在一个圈，且圈中至少有一个点的度数 ≥ 3，记这个点为 r。

- 如果连通分量存在 1 度点，那么显然从它开始。
- 否则，该连通分量中一定存在一个圈，且圈中至少有一个点的度数 ≥ 3，记这个点为 r。

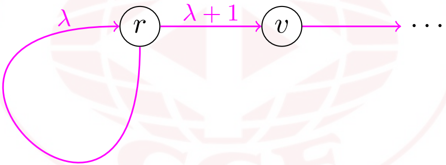


图 13

现在只剩下一类问题：圈。

现在只剩下一类问题：圈。

我们需要将这些圈作为整体考虑，即假设 G 是由若干个圈构成的 2-正则图。

现在只剩下一类问题：圈。

我们需要将这些圈作为整体考虑，即假设 G 是由若干个圈构成的 2-正则图。

G 是二分图 $\Rightarrow$ G 只包含偶圈。

现在只剩下一类问题：圈。

我们需要将这些圈作为整体考虑，即假设 G 是由若干个圈构成的 2-正则图。

G 是二分图 $\Rightarrow G$ 只包含偶圈。

在这个子问题中，点和边的地位是对等的，因此可以考虑给点分配 $1 \sim |V|$ 的标号，使得相邻点标号互素。

中国计算机学会 China Computer Federation

现在只剩下一类问题：圈。

我们需要将这些圈作为整体考虑，即假设 G 是由若干个圈构成的 2-正则图。

G 是二分图 $\Rightarrow G$ 只包含偶圈。

在这个子问题中，点和边的地位是对等的，因此可以考虑给点分配 $1 \sim |V|$ 的标号，使得相邻点标号互素。

等价地，令 $H_n = (V_n, E_n)$，其中 $V_n = \{1, 2, \dots, n\}, E_n = \{(i, j) \,|\, 1 \le i, j \le n \text{ 且 } \gcd(i, j) = 1\}$。

等价地，令 $H_n = (V_n, E_n)$，其中 $V_n = \{1, 2, \ldots, n\}, E_n = \{(i, j) \mid 1 \le i, j \le n \text{ 且 } \gcd(i, j) = 1\}$。

找到 G 到 H_n 的嵌入！

中国计算机学会 China Computer Federation

等价地，令 $H_n = (V_n, E_n)$，其中 $V_n = \{1, 2, \ldots, n\}, E_n = \{(i, j) \mid 1 \le i, j \le n \text{ 且 } \gcd(i, j) = 1\}$。

找到 G 到 H_n 的嵌入！

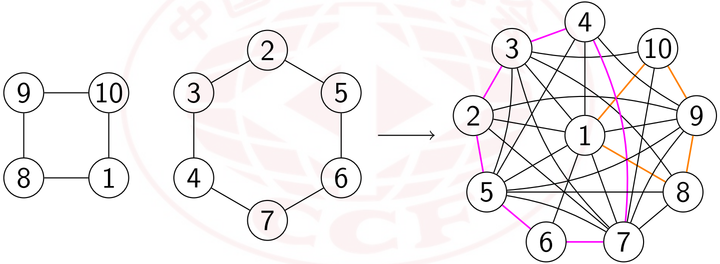


图 14

如何利用 H_n？寻找它的一个性质较好的生成子图，使得对这种 2-正则图 G，都存在到 I_n 的嵌入。

如何利用 H_n？寻找它的一个性质较好的生成子图，使得对这种 2-正则图 G，都存在到 I_n 的嵌入。

取 G 为梯子图 *(Ladder graph)* L_n (即 $2 \times n$ 网格图)！

如何利用 H_n? 寻找它的一个性质较好的生成子图，使得对这种 2-正则图 G，都存在到 I_n 的嵌入。

取 G 为梯子图 *(Ladder graph)* L_n (即 $2 \times n$ 网格图)!

所有 $2n$ 阶 2-正则二分图都可以嵌入到梯子图 L_n 中。

如何利用 H_n？寻找它的一个性质较好的生成子图，使得对这种 2-正则图 G，都存在到 I_n 的嵌入。

取 G 为梯子图 *(Ladder graph)* L_n (即 $2 \times n$ 网格图)！

所有 $2n$ 阶 2-正则二分图都可以嵌入到梯子图 L_n 中。

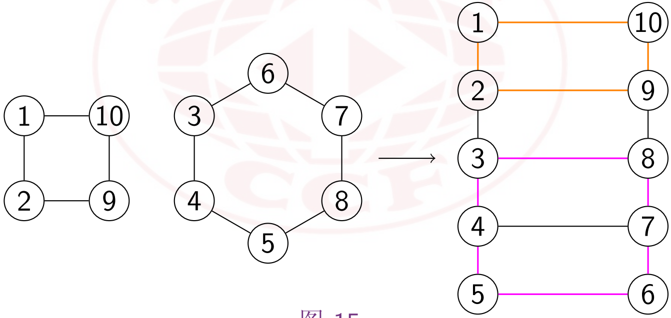


图 15

是否有 $L_n \subseteq H_{2n}$？

是否有 $L_n \subseteq H_{2n}$？
如果 $2n+1$ 是素数，那么容易构造。

是否有 $L_n \subseteq H_{2n}$？

如果 $2n+1$ 是素数，那么容易构造。

否则？取正奇数 $2k+1$ 使得 $2n+2k+1$ 是素数，然后将梯子搭到 $(n+k, n+k+1)$。

是否有 $L_n \subseteq H_{2n}$？

如果 $2n+1$ 是素数，那么容易构造。

否则？取正奇数 $2k+1$ 使得 $2n+2k+1$ 是素数，然后将梯子搭到 $(n+k, n+k+1)$。

如果这个梯子和 $2k$ 时构造的梯子相容，那么，就完成了 $2n$ 时的构造。

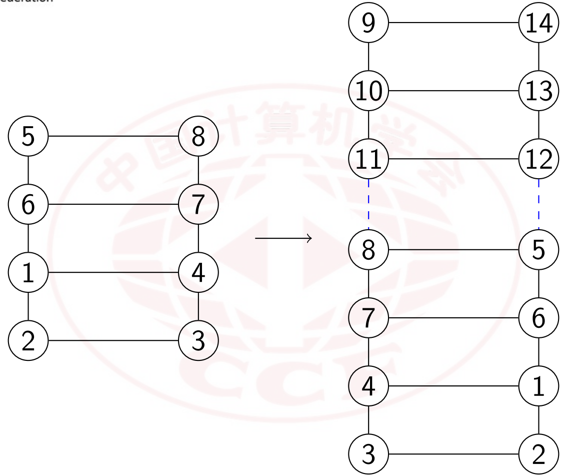


图 16

这样的构造一定存在吗？如果存在，怎样快速寻找？

这样的构造一定存在吗？如果存在，怎样快速寻找？

其实很简单，我们不需要证明对任意 n 存在，我们只需要证明对 $n \le 2 \times 10^5$ 存在，这样只需用程序暴力验证一下对于每个 $2n$ 的最小解 $c(2n)$。

中国计算机学会 China Computer Federation

这样的构造一定存在吗？如果存在，怎样快速寻找？

其实很简单，我们不需要证明对任意 n 存在，我们只需要证明对 $n \le 2 \times 10^5$ 存在，这样只需用程序暴力验证一下对于每个 $2n$ 的最小解 $c(2n)$。

经验证知 $\sum_{n=1}^{10^5} c(2n) = 1\,272\,400$，因此可以接受。

这样的构造一定存在吗？如果存在，怎样快速寻找？

其实很简单，我们不需要证明对任意 n 存在，我们只需要证明对 $n \le 2 \times 10^5$ 存在，这样只需用程序暴力验证一下对于每个 $2n$ 的最小解 $c(2n)$。

经验证知 $\sum_{n=1}^{10^5} c(2n) = 1\,272\,400$，因此可以接受。

对于一般图的情形，要注意 1 可能会分配给 1 度点之间的连边，因此实际实现的时候我们是对 $2 \sim 2n+1$ 进行 “搭梯子”。

给定正整数 a, b，初始时 $S = \{\frac{a}{b}, \frac{b}{a}\}$。你可以进行若干次如下操作：

- 选择 S 中两个数 x, y，令 $S \leftarrow S \cup \left\{\frac{x+y}{2}, \frac{2xy}{x+y}\right\}$。

求是否可以在有限步内使 $1 \in S$。$1 \le a, b \le 10^9$。

中国计算机学会 China Computer Federation

不难发现，如果 S 中只有一个元素 x，那么无论怎么操作 S 始终为 $\{x\}$。

中国计算机学会 China Computer Federation

不难发现，如果 S 中只有一个元素 x，那么无论怎么操作 S 始终为 $\{x\}$。

于是考虑在模意义下进行计算。不妨设 a, b 互素，取素数 $p \mid a+b$，则在 $\pmod p$ 意义下 $S=\{-1\}$。

中国计算机学会 China Computer Federation

不难发现，如果 S 中只有一个元素 x，那么无论怎么操作 S 始终为 $\{x\}$。

于是考虑在模意义下进行计算。不妨设 a, b 互素，取素数 $p \mid a+b$，则在 $\pmod p$ 意义下 $S = \{-1\}$。

这说明，如果欲使 $1 \in S$，$a+b$ 不能有奇素因子，即 $a+b$ 必须是 2 的幂。

不难发现，如果 S 中只有一个元素 x，那么无论怎么操作 S 始终为 $\{x\}$。

于是考虑在模意义下进行计算。不妨设 a, b 互素，取素数 $p \mid a+b$，则在 $(\bmod\ p)$ 意义下 $S = \{-1\}$。

这说明，如果欲使 $1 \in S$，$a+b$ 不能有奇素因子，即 $a+b$ 必须是 2 的幂。

另一方面，如果 $a+b$ 是 2 的幂，注意到

$$\frac{b}{a+b} \cdot \frac{a}{b} + \frac{a}{a+b} \cdot \frac{b}{a} = 1,$$

我们可以只用算术平均操作达成目的。

中国计算机学会 China Computer Federation

给定一个群 $(S,\circ)$，你需要构造一张无向简单图 $G=(V,E)$ 满足它的自同构群 $\operatorname{Aut} G \cong S$，且 $|V| \le 127$。

$1 \le |S| \le 63$。

中国计算机学会 China Computer Federation

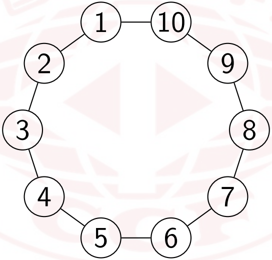
根据数据规模，猜测要求是 $|V| \leq 2|S|$。

根据数据规模，猜测要求是 $|V| \leq 2|S|$。

先考虑 $S \cong Z_n$。

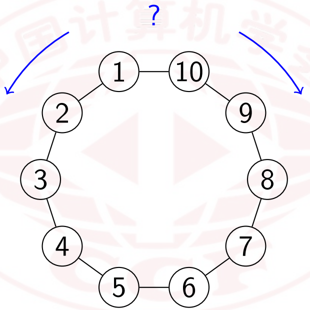
根据数据规模，猜测要求是 $|V| \le 2|S|$。

先考虑 $S \cong Z_n$。



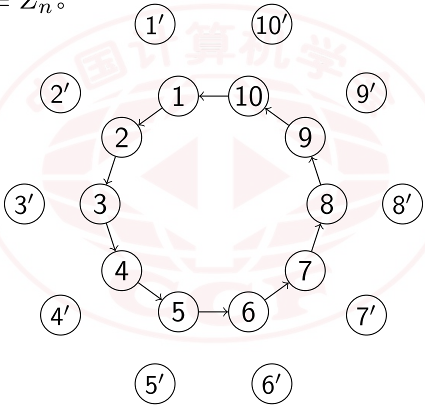
根据数据规模，猜测要求是 $|V| \le 2|S|$。

先考虑 $S \cong Z_n$。



根据数据规模，猜测要求是 $|V| \leq 2|S|$。

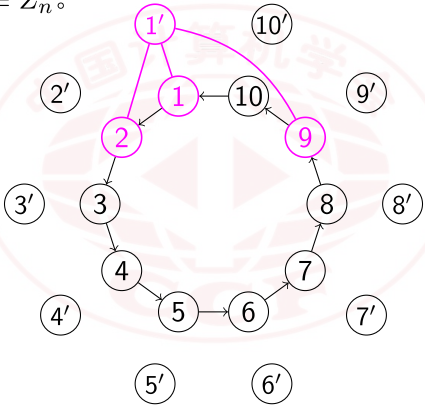
先考虑 $S \cong Z_n$。



中国计算机学会 China Computer Federation

根据数据规模，猜测要求是 $|V| \le 2|S|$。

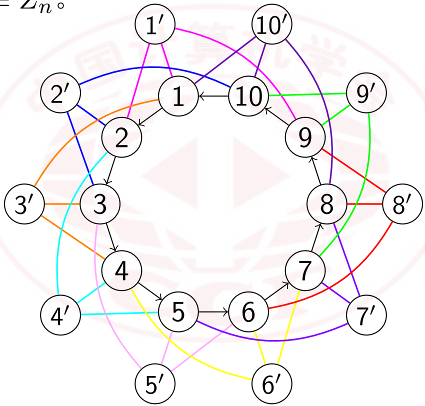
先考虑 $S \cong Z_n$。



中国计算机学会 China Computer Federation

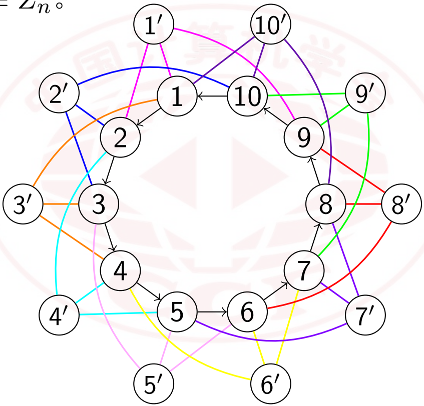
根据数据规模，猜测要求是 $|V| \leq 2|S|$。

先考虑 $S \cong Z_n$。



根据数据规模，猜测要求是 $|V| \le 2|S|$。

先考虑 $S \cong Z_n$。



这个构造适用于 $n \ge 6$。

$n \le 5$ 时，我们可以给一个稍劣一点的构造：

中国计算机学会 China Computer Federation

$n \le 5$ 时，我们可以给一个稍劣一点的构造：

图 17: Z_1

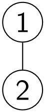


图 18: Z_2

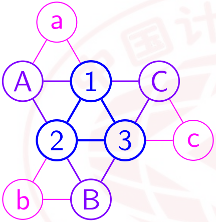


图 19: Z_3

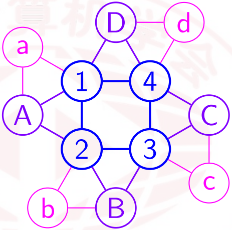


图 20: Z_4

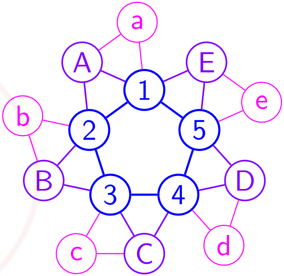


图 21: Z_5

对于 S 为一般群的情形，我们有

中国计算机学会
China Computer Federation

对于 S 为一般群的情形，我们有

Theorem 4.1 (Cayley)

任何一个有限群 S 都同构于一个 S 上的置换群，同构于一个由 $|S|$ 元置换构成的群。具体地，令左乘变换 $\varphi_g(x) = g \circ x$ 为 S 上的一个置换，则所有 $|S|$ 个置换构成的群与 S 同构。

中国计算机学会 China Computer Federation

对于 S 为一般群的情形，我们有

Theorem 4.1 (Cayley)

任何一个有限群 S 都同构于一个 S 上的置换群，同构于一个由 $|S|$ 元置换构成的群。具体地，令左乘变换 $\varphi_g(x) = g \circ x$ 为 S 上的一个置换，则所有 $|S|$ 个置换构成的群与 S 同构。

这说明，对于任意群 S，它一定同构于其自同构群 $\operatorname{Aut} S$ 的一个子群。根据这种思路，我们构建出一些图 G，使得 S 同构于 $\operatorname{Aut} G$ 的一个子群。

中国计算机学会 China Computer Federation

Definition 4.1 (Cayley graph)

对于群 $(S,\circ)$ 和子集 $H \subseteq S$，定义 H 在 S 中生成的 Cayley 图为一张有向图 $G=(V,E)$，记作 $\Gamma(S,H)$，其中

$$V=S,\ E=\left\{\left(x,\varphi_g(x)\right)\,\middle|\,x\in S\right\}.$$

中国计算机学会 China Computer Federation

Definition 4.1 (Cayley graph)

对于群 $(S,\circ)$ 和子集 $H \subseteq S$，定义 H 在 S 中生成的 Cayley 图为一张有向图 $G=(V,E)$，记作 $\Gamma(S,H)$，其中

$$V=S,\ E=\{(x,\varphi_g(x))\mid x\in S\}.$$

注意到所有 φ_g 均为 $\Gamma(S,H)$ 的自同构，于是

Theorem 4.2

对于 $\forall H\subseteq S$，设 $G=\Gamma(S,H)$ 是 H 在 S 中生成的 Cayley 图，则 S 同构于 $\operatorname{Aut} G$ 的一个子群。

注：关于图有向无向的问题，如果我们取的 H 满足 $x\in H \iff x^{-1}\in H$，那么所得的图 G 就自然可以看成一张无向图。

现在我们已经成功让所有 φ_g 均成为 G 的自同构，现在要做的，就是不让 G 中出现其它的自同构。

现在我们已经成功让所有 φ_g 均成为 G 的自同构，现在要做的，就是不让 G 中出现其它的自同构。

由于群中所有元素可逆，因此任给元素 $x, y \in S$，使得 $\varphi_g(x) = y$ 的 g 存在且唯一。

现在我们已经成功让所有 φ_g 均成为 G 的自同构，现在要做的，就是不让 G 中出现其它的自同构。

由于群中所有元素可逆，因此任给元素 $x, y \in S$，使得 $\varphi_g(x) = y$ 的 g 存在且唯一。

因此，对于图中任意一个点 $x \in S$，应当存在唯一的 g 使得 $\varphi_g(e) = x$。

中国计算机学会 China Computer Federation

现在我们已经成功让所有 φ_g 均成为 G 的自同构，现在要做的，就是不让 G 中出现其它的自同构。

由于群中所有元素可逆，因此任给元素 $x, y \in S$，使得 $\varphi_g(x) = y$ 的 g 存在且唯一。

因此，对于图中任意一个点 $x \in S$，应当存在唯一的 g 使得 $\varphi_g(e) = x$。

而事实上其中的 g 就等于 x。因此我们只需保证存在唯一的自同构，将顶点 e 置换到顶点 x。

现在我们已经成功让所有 φ_g 均成为 G 的自同构，现在要做的，就是不让 G 中出现其它的自同构。

由于群中所有元素可逆，因此任给元素 $x, y \in S$，使得 $\varphi_g(x) = y$ 的 g 存在且唯一。

因此，对于图中任意一个点 $x \in S$，应当存在唯一的 g 使得 $\varphi_g(e) = x$。

而事实上其中的 g 就等于 x。因此我们只需保证存在唯一的自同构，将顶点 e 置换到顶点 x。

要达成这一点，我们只需要“在给定 $\varphi(e) = x$ 的条件下，可以唯一确定其它元素的像”即可。

以 $S=\langle x,y,\omega \mid x^3=y^3=\omega^2=(\omega x)^2=(\omega y)^2=xyx^{-1}y^{-1}=1\rangle$

为例，这是一个 18 阶群，其中 $\{x,y,\omega\}$ 为它的一个生成集。

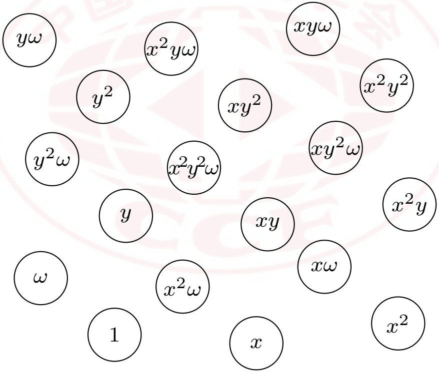
中国计算机学会 China Computer Federation

以 $S=\left\langle x,y,\omega \,\middle|\, x^3=y^3=\omega^2=(\omega x)^2=(\omega y)^2=xyx^{-1}y^{-1}=1\right\rangle$ 为例，这是一个 18 阶群，其中 $\{x,y,\omega\}$ 为它的一个生成集。

该群的 Cayley 图如下：

以 $S = \langle x, y, \omega \mid x^3 = y^3 = \omega^2 = (\omega x)^2 = (\omega y)^2 = xyx^{-1}y^{-1} = 1 \rangle$ 为例，这是一个 18 阶群，其中 $\{x, y, \omega\}$ 为它的一个生成集。

该群的 Cayley 图如下：

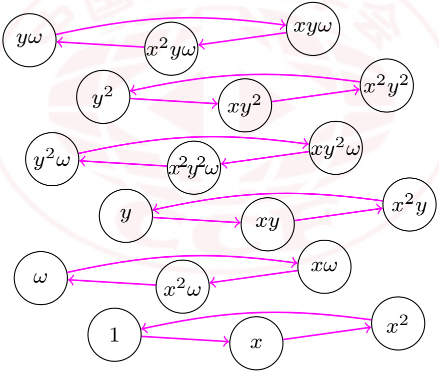


中国计算机学会 China Computer Federation

以 $S=\left\langle x,y,\omega \,\middle|\, x^3=y^3=\omega^2=(\omega x)^2=(\omega y)^2=xyx^{-1}y^{-1}=1\right\rangle$

为例，这是一个 18 阶群，其中 $\{x,y,\omega\}$ 为它的一个生成集。

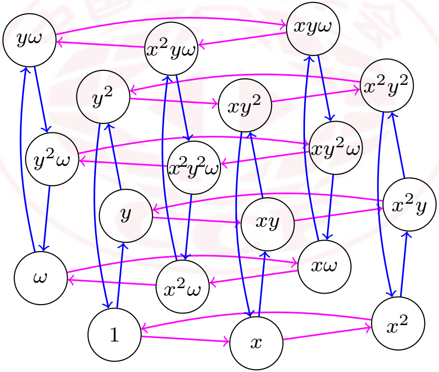
该群的 Cayley 图如下：



中国计算机学会 China Computer Federation

以 $S=\langle x,y,\omega \mid x^3=y^3=\omega^2=(\omega x)^2=(\omega y)^2=xyx^{-1}y^{-1}=1\rangle$ 为例，这是一个 18 阶群，其中 $\{x,y,\omega\}$ 为它的一个生成集。

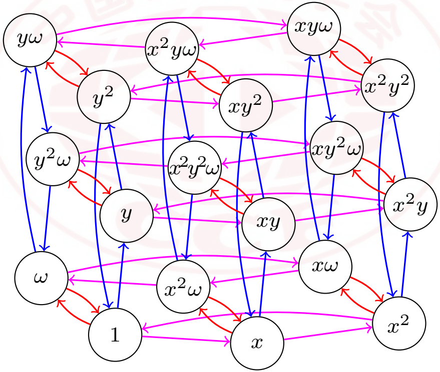
该群的 Cayley 图如下：



以 $S = \langle x, y, \omega \mid x^3 = y^3 = \omega^2 = (\omega x)^2 = (\omega y)^2 = xyx^{-1}y^{-1} = 1 \rangle$

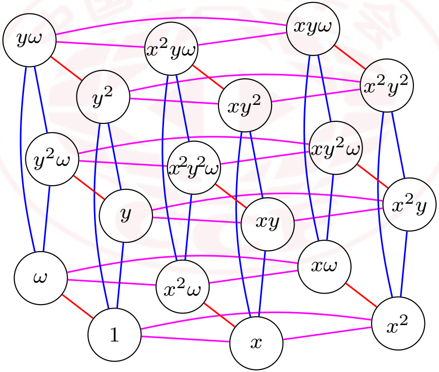
为例，这是一个 18 阶群，其中 $\{x, y, \omega\}$ 为它的一个生成集。

该群的 Cayley 图如下：



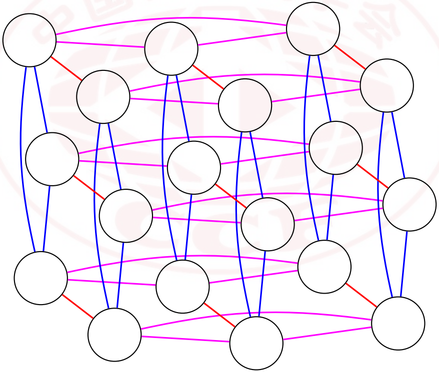
以 $S = \langle x, y, \omega \mid x^3 = y^3 = \omega^2 = (\omega x)^2 = (\omega y)^2 = xyx^{-1}y^{-1} = 1 \rangle$ 为例，这是一个 18 阶群，其中 $\{x, y, \omega\}$ 为它的一个生成集。

该群的 Cayley 图如下：

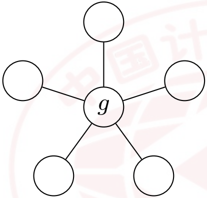


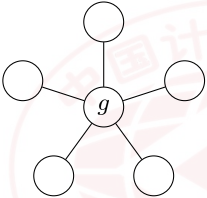
以 $S = \langle x, y, \omega \mid x^3 = y^3 = \omega^2 = (\omega x)^2 = (\omega y)^2 = xyx^{-1}y^{-1} = 1 \rangle$
为例，这是一个 18 阶群，其中 $\{x, y, \omega\}$ 为它的一个生成集。

该群的 Cayley 图如下：



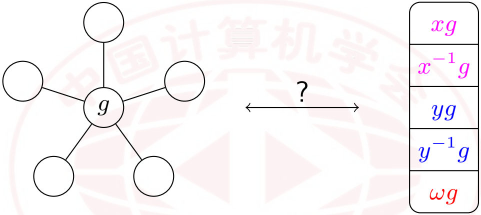

g

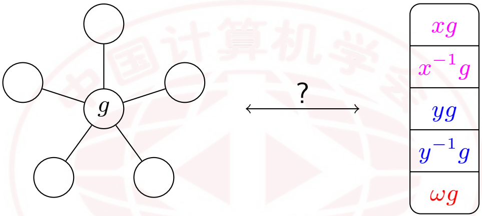

g



xg
$x^{-1}g$
yg
$y^{-1}g$
ωg

中国计算机学会
China Computer Federation





这样显然是无法还原的，因为我们丢失了方向信息和顺序信息。

中国计算机学会 China Computer Federation

如何在图中还原列表的顺序？

如何在图中还原列表的顺序？

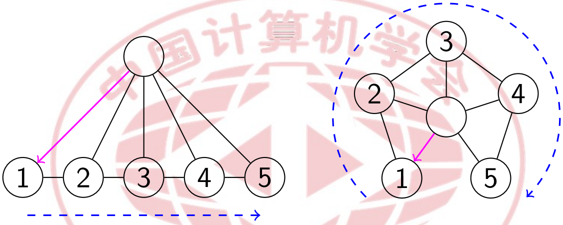


图 22

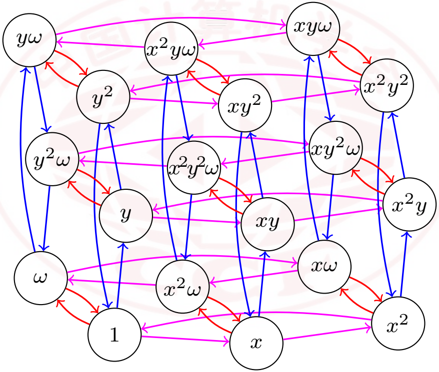
经典构造：知“1”推序列。

中国计算机学会
China Computer Federation

如何还原方向？

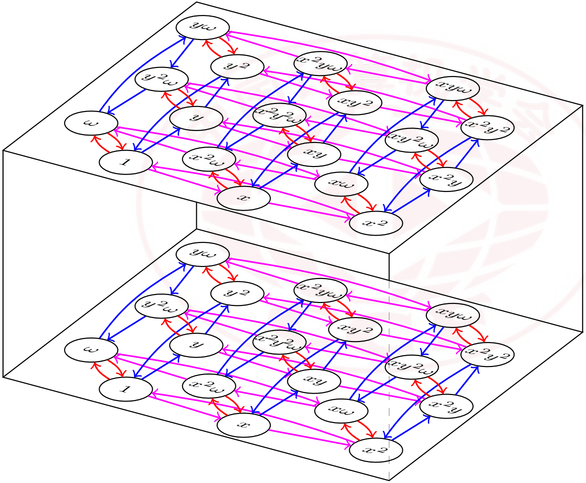
中国计算机学会 China Computer Federation

如何还原方向？ Clone!



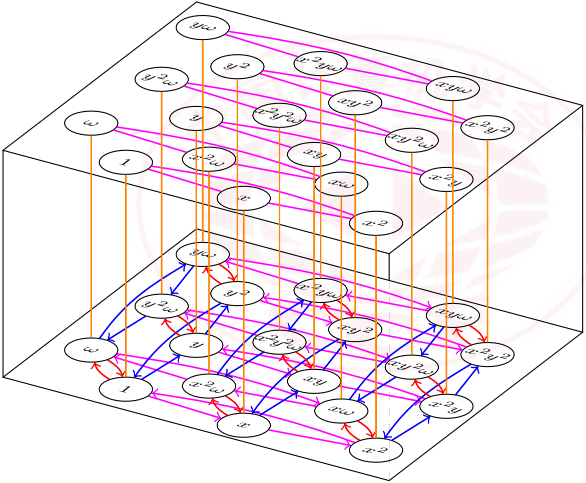
中国计算机学会 China Computer Federation

如何还原方向？ Clone!



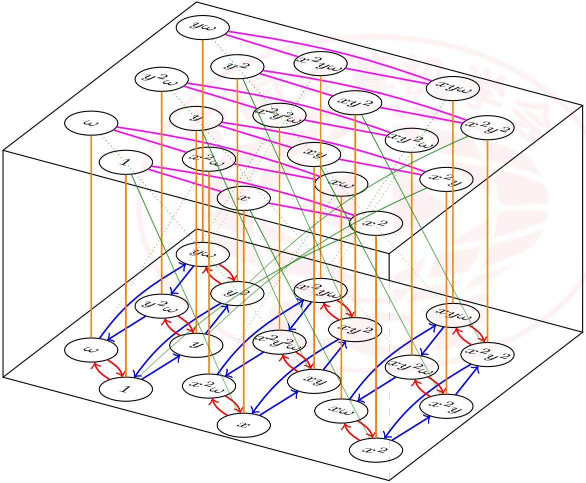
中国计算机学会 China Computer Federation

如何还原方向？Clone!



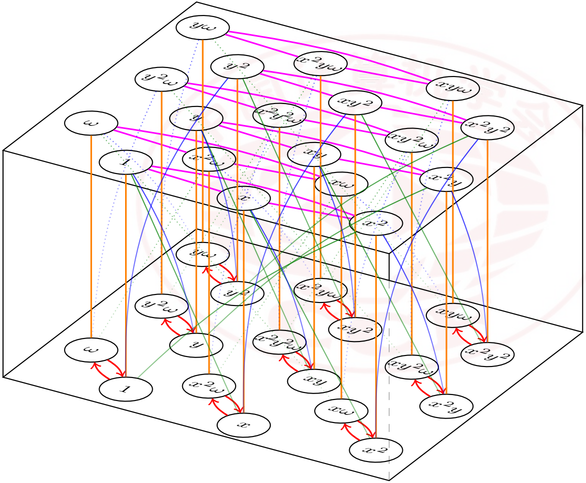
中国计算机学会
China Computer Federation

如何还原方向？Clone!



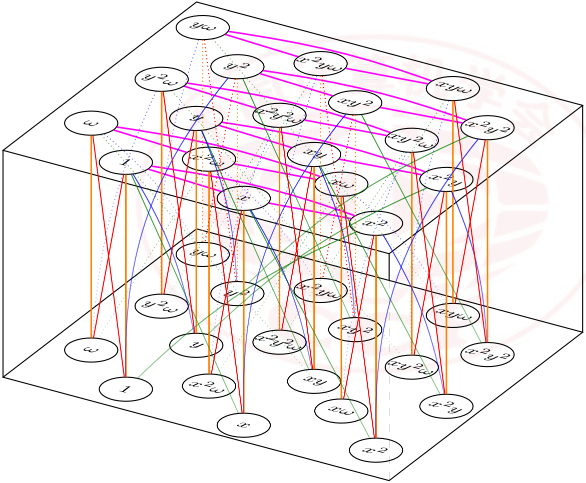
中国计算机学会 China Computer Federation

如何还原方向？ Clone!



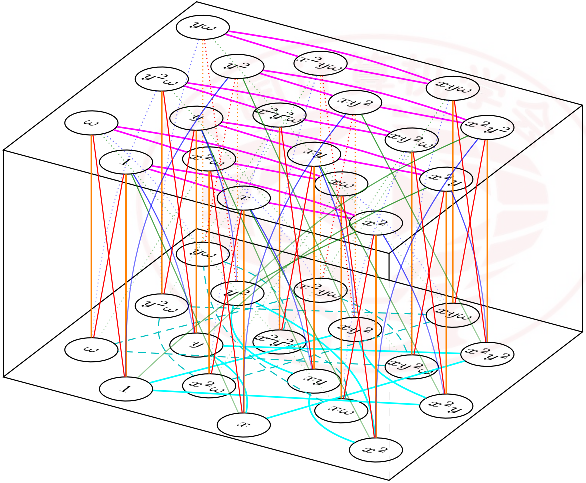
中国计算机学会 China Computer Federation

如何还原方向？ Clone!



中国计算机学会 China Computer Federation

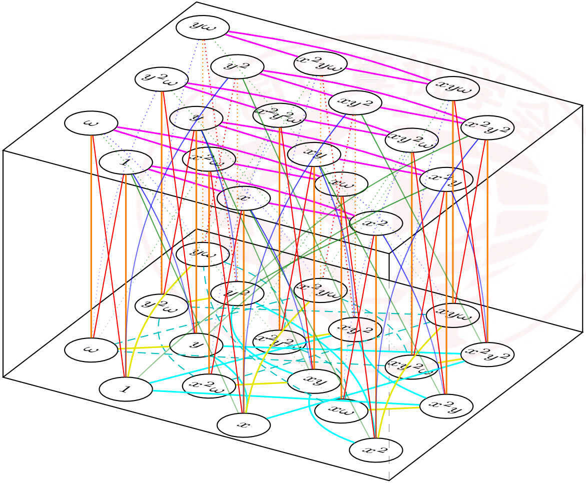
如何还原方向？Clone!



$$(\varphi_x(g), \varphi_y(g))$$

$$(g, yx^{-1}g)$$

如何还原方向？Clone!



$$(\varphi_y(g), \varphi_\omega(g))$$

$$(g, \omega y^{-1} g)$$

记所得到的新图中上层为 G_1，下层为 G_2，所做的 Cayley 图中所取的子集为 $H=\{h_1,h_2,\ldots,h_d\}$。

记所得到的新图中上层为 G_1，下层为 G_2，所做的 Cayley 图中所取的子集为 $H=\{h_1,h_2,\ldots,h_d\}$。

还原的具体步骤：

中国计算机学会 China Computer Federation

记所得到的新图中上层为 G_1，下层为 G_2，所做的 Cayley 图中所取的子集为 $H = \{h_1, h_2, \ldots, h_d\}$。

还原的具体步骤：

1. 区分 G_1 和 G_2。

记所得到的新图中上层为 G_1，下层为 G_2，所做的 Cayley 图中所取的子集为 $H = \{h_1, h_2, \ldots, h_d\}$。

还原的具体步骤：

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。

记所得到的新图中上层为 G_1，下层为 G_2，所做的 Cayley 图中所取的子集为 $H = \{h_1, h_2, \ldots, h_d\}$。

还原的具体步骤：

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像 $\varphi(e)$。

中国计算机学会 China Computer Federation

记所得到的新图中上层为 G_1，下层为 G_2，所做的 Cayley 图中所取的子集为 $H = \{h_1, h_2, \ldots, h_d\}$。

还原的具体步骤：

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像 $\varphi(e)$。
4. 已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。

记所得到的新图中上层为 G_1，下层为 G_2，所做的 Cayley 图中所取的子集为 $H = \{h_1, h_2, \ldots, h_d\}$。

还原的具体步骤：

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像 $\varphi(e)$。
4. 已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2 x), \ldots, \varphi(h_d x)$ 对应的点。

中国计算机学会 China Computer Federation

记所得到的新图中上层为 G_1，下层为 G_2，所做的 Cayley 图中所取的子集为 $H = \{h_1, h_2, \dots, h_d\}$。

还原的具体步骤：

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像 $\varphi(e)$。
4. 已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2 x), \dots, \varphi(h_d x)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

记所得到的新图中上层为 G_1，下层为 G_2，所做的 Cayley 图中所取的子集为 $H = \{h_1, h_2, \ldots, h_d\}$。

还原的具体步骤：

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像 $\varphi(e)$。
4. 已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2 x), \ldots, \varphi(h_d x)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

若通过步骤 6 能求出所有元素的像，H 必须能生成群 S，即，$\langle H \rangle = S$。

中国计算机学会 China Computer Federation

记所得到的新图中上层为 G_1，下层为 G_2，所做的 Cayley 图中所取的子集为 $H=\{h_1,h_2,\ldots,h_d\}$。

还原的具体步骤：

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像 $\varphi(e)$。
4. 已知 $\varphi(x)=y$，找到 $\varphi(h_1x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2x),\ldots,\varphi(h_dx)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

若通过步骤 6 能求出所有元素的像，H 必须能生成群 S，即，$\langle H\rangle=S$。

中国计算机学会 China Computer Federation

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像。
4. 已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2 x), \ldots, \varphi(h_d x)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

区分 G_1 和 G_2。

区分 G_1 和 G_2。

G_1, G_2 均为正则图。$\Delta(G_1) \le 2$！

区分 G_1 和 G_2。

G_1, G_2 均为正则图。$\Delta(G_1) \leq 2$！

$|S| \neq 1, 3, 5 \Rightarrow \Delta(G_2)$ 和 $(|S| - 1) - \Delta(G_2)$ 至少有一个和 $\Delta(G_1)$ 不同。

中国计算机学会 China Computer Federation

区分 G_1 和 G_2。

G_1, G_2 均为正则图。$\Delta(G_1) \le 2$！

$|S| \ne 1, 3, 5 \Rightarrow \Delta(G_2)$ 和 $(|S| - 1) - \Delta(G_2)$ 至少有一个和 $\Delta(G_1)$ 不同。

用度数区分！

中国计算机学会 China Computer Federation

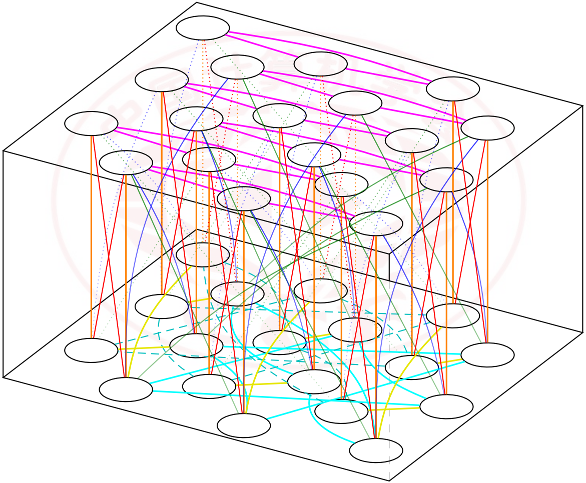
1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像。
4. 已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2 x), \ldots, \varphi(h_d x)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

中国计算机学会 China Computer Federation

找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。

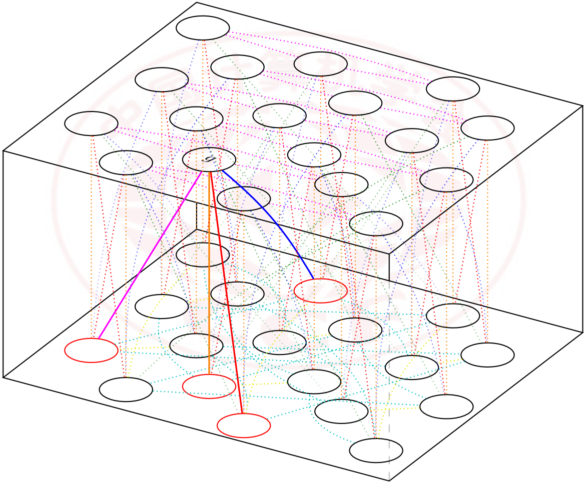
中国计算机学会 China Computer Federation

找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。

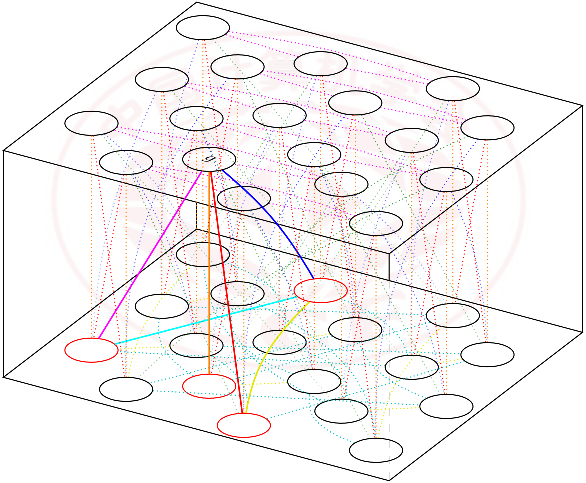


中国计算机学会 China Computer Federation

找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。



找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。



令 $N = N(g) \cap G_2$ 为 g 的邻域在 G_2 中的部分。

令 $N = N(g) \cap G_2$ 为 g 的邻域在 G_2 中的部分。

设 N 导出的子图 $G_2[N] = (V', E')$，则 $V' = \{\psi(g), \psi(h_1 g), \psi(h_2 g), \ldots, \psi(h_d g)\}$。

令 $N = N(g) \cap G_2$ 为 g 的邻域在 G_2 中的部分。

设 N 导出的子图 $G_2[N] = (V', E')$，则 $V' = \{\psi(g), \psi(h_1 g), \psi(h_2 g), \ldots, \psi(h_d g)\}$。

因此，$G_2[N]$ 包含两个连通分量，其中一个只包含一个顶点 $\psi(g)$，另外一个是非空链 $\psi(h_1 g) \leftrightarrow \psi(h_2 g) \leftrightarrow \cdots \leftrightarrow \psi(h_d g)$。

令 $N = N(g) \cap G_2$ 为 g 的邻域在 G_2 中的部分。

设 N 导出的子图 $G_2[N] = (V', E')$，则 $V' = \{\psi(g), \psi(h_1 g), \psi(h_2 g), \ldots, \psi(h_d g)\}$。

- 由构造知，$\psi(h_i g)$ 和 $\psi(h_{i+1} g)$ 在 G_2 中有连边。

因此，$G_2[N]$ 包含两个连通分量，其中一个只包含一个顶点 $\psi(g)$，另外一个是非空链 $\psi(h_1 g) \leftrightarrow \psi(h_2 g) \leftrightarrow \cdots \leftrightarrow \psi(h_d g)$。

令 $N = N(g) \cap G_2$ 为 g 的邻域在 G_2 中的部分。

设 N 导出的子图 $G_2[N] = (V', E')$，则 $V' = \{\psi(g), \psi(h_1 g), \psi(h_2 g), \dots, \psi(h_d g)\}$。

- 由构造知，$\psi(h_i g)$ 和 $\psi(h_{i+1} g)$ 在 G_2 中有连边。
- $d \geq 2$。

因此，$G_2[N]$ 包含两个连通分量，其中一个只包含一个顶点 $\psi(g)$，另外一个是非空链 $\psi(h_1 g) \leftrightarrow \psi(h_2 g) \leftrightarrow \cdots \leftrightarrow \psi(h_d g)$。

中国计算机学会 China Computer Federation

令 $N = N(g) \cap G_2$ 为 g 的邻域在 G_2 中的部分。

设 N 导出的子图 $G_2[N] = (V', E')$，则 $V' = \{\psi(g), \psi(h_1 g), \psi(h_2 g), \ldots, \psi(h_d g)\}$。

- 由构造知，$\psi(h_i g)$ 和 $\psi(h_{i+1} g)$ 在 G_2 中有连边。
- $d \geq 2$。
- $\psi(g)$ 和 $\psi(h_i g)$ 在 G_2 中没有连边。

因此，$G_2[N]$ 包含两个连通分量，其中一个只包含一个顶点 $\psi(g)$，另外一个是非空链 $\psi(h_1 g) \leftrightarrow \psi(h_2 g) \leftrightarrow \cdots \leftrightarrow \psi(h_d g)$。

令 $N = N(g) \cap G_2$ 为 g 的邻域在 G_2 中的部分。

设 N 导出的子图 $G_2[N] = (V', E')$，则 $V' = \{\psi(g), \psi(h_1 g), \psi(h_2 g), \ldots, \psi(h_d g)\}$。

- 由构造知，$\psi(h_i g)$ 和 $\psi(h_{i+1} g)$ 在 G_2 中有连边。
- $d \geq 2$。
- $\psi(g)$ 和 $\psi(h_i g)$ 在 G_2 中没有连边。
- 对 $|i-j| > 1$，$\psi(h_i g)$ 和 $\psi(h_j g)$ 在 G_2 中也没有连边。

因此，$G_2[N]$ 包含两个连通分量，其中一个只包含一个顶点 $\psi(g)$，另外一个是非空链 $\psi(h_1 g) \leftrightarrow \psi(h_2 g) \leftrightarrow \cdots \leftrightarrow \psi(h_d g)$。

令 $N = N(g) \cap G_2$ 为 g 的邻域在 G_2 中的部分。

设 N 导出的子图 $G_2[N] = (V', E')$，则 $V' = \{\psi(g), \psi(h_1 g), \psi(h_2 g), \ldots, \psi(h_d g)\}$。

- 由构造知，$\psi(h_i g)$ 和 $\psi(h_{i+1} g)$ 在 G_2 中有连边。
- $d \geq 2$。
- $h_{k+1} h_k^{-1} \neq h_i$。
- 对 $|i-j| > 1$，$\psi(h_i g)$ 和 $\psi(h_j g)$ 在 G_2 中也没有连边。

因此，$G_2[N]$ 包含两个连通分量，其中一个只包含一个顶点 $\psi(g)$，另外一个是非空链 $\psi(h_1 g) \leftrightarrow \psi(h_2 g) \leftrightarrow \cdots \leftrightarrow \psi(h_d g)$。

中国计算机学会 China Computer Federation

令 $N = N(g) \cap G_2$ 为 g 的邻域在 G_2 中的部分。

设 N 导出的子图 $G_2[N] = (V', E')$，则 $V' = \{\psi(g), \psi(h_1 g), \psi(h_2 g), \dots, \psi(h_d g)\}$。

- 由构造知，$\psi(h_i g)$ 和 $\psi(h_{i+1} g)$ 在 G_2 中有连边。
- $d \geq 2$。
- $h_{k+1} h_k^{-1} \neq h_i$。
- $h_{k+1} h_k^{-1} h_i \neq h_j \ (|i - j| > 1)$。

因此，$G_2[N]$ 包含两个连通分量，其中一个只包含一个顶点 $\psi(g)$，另外一个是非空链 $\psi(h_1 g) \leftrightarrow \psi(h_2 g) \leftrightarrow \cdots \leftrightarrow \psi(h_d g)$。

令 $N = N(g) \cap G_2$ 为 g 的邻域在 G_2 中的部分。

设 N 导出的子图 $G_2[N] = (V', E')$，则 $V' = \{\psi(g), \psi(h_1 g), \psi(h_2 g), \ldots, \psi(h_d g)\}$。

- 由构造知，$\psi(h_i g)$ 和 $\psi(h_{i+1} g)$ 在 G_2 中有连边。
- $d \geq 2$。
- $h_{k+1} h_k^{-1} \neq h_i$。 $\Leftarrow$ 任意一个 h_i 都无法被其余 h_k 表示
- $h_{k+1} h_k^{-1} h_i \neq h_j$ $(|i-j| > 1)$。

因此，$G_2[N]$ 包含两个连通分量，其中一个只包含一个顶点 $\psi(g)$，另外一个是非空链 $\psi(h_1 g) \leftrightarrow \psi(h_2 g) \leftrightarrow \cdots \leftrightarrow \psi(h_d g)$。

中国计算机学会 China Computer Federation

令 $N = N(g) \cap G_2$ 为 g 的邻域在 G_2 中的部分。

设 N 导出的子图 $G_2[N] = (V', E')$，则 $V' = \{\psi(g), \psi(h_1 g), \psi(h_2 g), \ldots, \psi(h_d g)\}$。

- 由构造知，$\psi(h_i g)$ 和 $\psi(h_{i+1} g)$ 在 G_2 中有连边。
- $d \geq 2$。
- $h_{k+1} h_k^{-1} \neq h_i$。 $\Leftarrow$ 任意一个 h_i 都无法被其余 h_k 表示
- $h_{k+1} h_k^{-1} h_i \neq h_j$ $(|i - j| > 1)$。

$\Uparrow$

H 是 S 的极小生成集

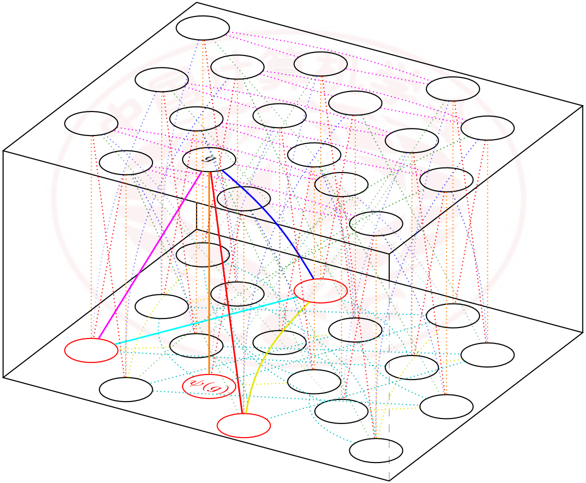
因此，$G_2[N]$ 包含两个连通分量，其中一个只包含一个顶点 $\psi(g)$，另外一个是非空链 $\psi(h_1 g) \leftrightarrow \psi(h_2 g) \leftrightarrow \cdots \leftrightarrow \psi(h_d g)$。

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像。
4. 已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2 x), \ldots, \varphi(h_d x)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

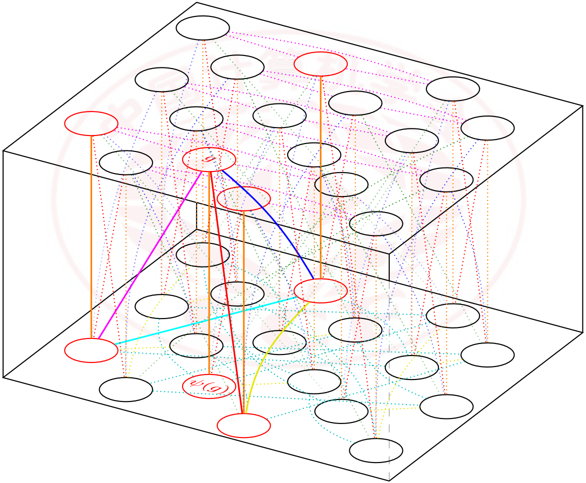
1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像。
4. 已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2 x), \ldots, \varphi(h_d x)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。

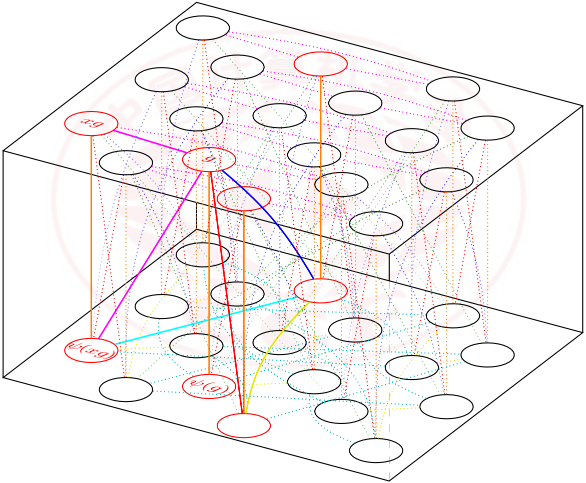
已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。



已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。



已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。



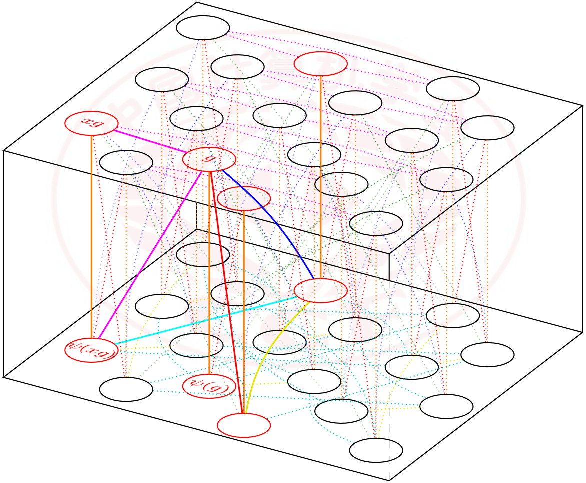
中国计算机学会 China Computer Federation

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像。
4. 已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2 x), \ldots, \varphi(h_d x)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

在步骤 4 的条件下，依次找到 $\varphi(h_2x), \ldots, \varphi(h_dx)$ 对应的点。

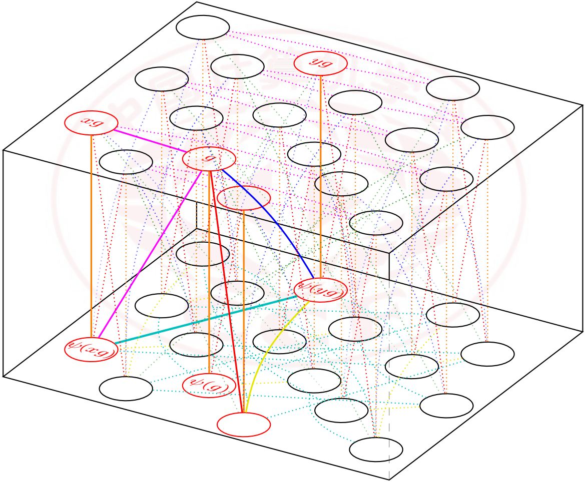
中国计算机学会 China Computer Federation

在步骤 4 的条件下，依次找到 $\varphi(h_2x),\ldots,\varphi(h_dx)$ 对应的点。



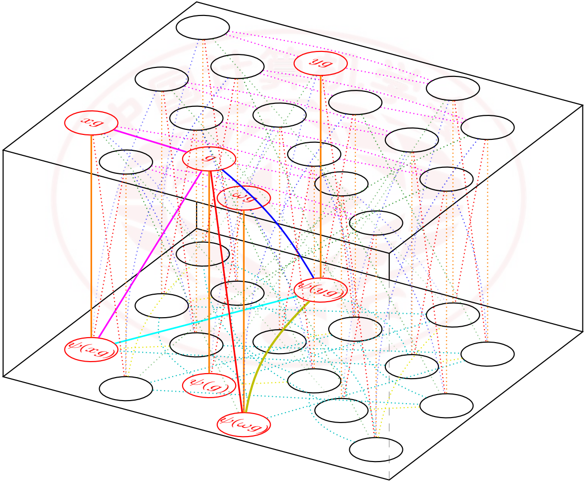
中国计算机学会 China Computer Federation

在步骤 4 的条件下，依次找到 $\varphi(h_2x),\dots,\varphi(h_dx)$ 对应的点。



中国计算机学会 China Computer Federation

在步骤 4 的条件下，依次找到 $\varphi(h_2x), \ldots, \varphi(h_dx)$ 对应的点。



中国计算机学会
China Computer Federation

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像。
4. 已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2 x), \ldots, \varphi(h_d x)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

中国计算机学会 China Computer Federation

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像。
4. 已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2 x), \ldots, \varphi(h_d x)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

回顾刚才过程中我们对 H 的要求：

中国计算机学会 China Computer Federation

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像。
4. 已知 $\varphi(x)=y$，找到 $\varphi(h_1x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2x),\dots,\varphi(h_dx)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

回顾刚才过程中我们对 H 的要求：

- $d\geq 2$。

中国计算机学会
China Computer Federation

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像。
4. 已知 $\varphi(x)=y$，找到 $\varphi(h_1x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2x),\ldots,\varphi(h_dx)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

回顾刚才过程中我们对 H 的要求：

- $d\geq 2$。
- H 是 S 的极小生成集，即 $\langle H\rangle=S$，但对 $\forall H'\subset H,\langle H'\rangle<S$。

中国计算机学会 China Computer Federation

1. 区分 G_1 和 G_2。
2. 找到 G_1 中的点和 G_2 中的点的一一对应关系 ψ。
3. 任取 G_1 中一点 v 作为 e 的像。
4. 已知 $\varphi(x) = y$，找到 $\varphi(h_1 x)$ 对应的点。
5. 在步骤 4 的条件下，依次找到 $\varphi(h_2 x), \ldots, \varphi(h_d x)$ 对应的点。
6. 重复步骤 4, 5，直到所有 $\varphi(x)$ 均被求出。

回顾刚才过程中我们对 H 的要求：

- $d \geq 2$。
- H 是 S 的极小生成集，即 $\langle H \rangle = S$，但对 $\forall H' \subset H, \langle H' \rangle < S$。

由于 S 不是循环群，因此 S 不存在大小为 1 的生成集。因此只需任取一个生成集贪心删元素即可得到满足条件的 H。

中国计算机学会 China Computer Federation

番外 之 本题的交互过程。

番外 之 本题的交互过程。

首先，通过要求选手给出 S 的置换表示，容易判断传入的图 G 是否满足 $S \leq \operatorname{Aut} G$。

番外 之 本题的交互过程。

首先，通过要求选手给出 S 的置换表示，容易判断传入的图 G 是否满足 $S \leq \operatorname{Aut} G$。

若 $S < \operatorname{Aut} G$，由 Lagrange 定理知 $|\operatorname{Aut} G| \geq 2\,|S|$。

番外 之 本题的交互过程。

首先，通过要求选手给出 S 的置换表示，容易判断传入的图 G 是否满足 $S \leq \operatorname{Aut} G$。

若 $S < \operatorname{Aut} G$，由 Lagrange 定理知 $|\operatorname{Aut} G| \geq 2|S|$。

重复进行如下过程：

- 交互库随机一个置换 g_1。

中国计算机学会 China Computer Federation

番外 之 本题的交互过程。

首先，通过要求选手给出 S 的置换表示，容易判断传入的图 G 是否满足 $S \le \operatorname{Aut} G$。

若 $S < \operatorname{Aut} G$，由 Lagrange 定理知 $|\operatorname{Aut} G| \ge 2\,|S|$。

重复进行如下过程：

- 交互库随机一个置换 g_1。
- 令 $H = (V, E')$，其中 $E' = \{(g_1(u), g_1(v)) \mid (u, v) \in E\}$。

番外 之 本题的交互过程。

首先，通过要求选手给出 S 的置换表示，容易判断传入的图 G 是否满足 $S \le \operatorname{Aut} G$。

若 $S < \operatorname{Aut} G$，由 Lagrange 定理知 $|\operatorname{Aut} G| \ge 2|S|$。

重复进行如下过程：

- 交互库随机一个置换 g_1。
- 令 $H = (V, E')$，其中 $E' = \{(g_1(u), g_1(v)) \mid (u, v) \in E\}$。
- 让选手寻找置换 g_2 满足 $(u, v) \in E \iff (g_2(u), g_2(v)) \in E'$。

中国计算机学会 China Computer Federation

番外 之 本题的交互过程。

首先，通过要求选手给出 S 的置换表示，容易判断传入的图 G 是否满足 $S \le \operatorname{Aut} G$。

若 $S < \operatorname{Aut} G$，由 Lagrange 定理知 $|\operatorname{Aut} G| \ge 2\,|S|$。

重复进行如下过程：

- 交互库随机一个置换 g_1。
- 令 $H = (V, E')$，其中 $E' = \{(g_1(u), g_1(v)) \mid (u, v) \in E\}$。
- 让选手寻找置换 g_2 满足 $(u, v) \in E \iff (g_2(u), g_2(v)) \in E'$。
- 则此时必有 $g_1^{-1} g_2 \in \operatorname{Aut} G$。若 $S < \operatorname{Aut} G$，则 $g_1^{-1} g_2 \in S$ 的概率不超过 $\frac{1}{2}$。

中国计算机学会 China Computer Federation

番外 之 本题的交互过程。

首先，通过要求选手给出 S 的置换表示，容易判断传入的图 G 是否满足 $S \le \operatorname{Aut} G$。

若 $S < \operatorname{Aut} G$，由 Lagrange 定理知 $|\operatorname{Aut} G| \ge 2\,|S|$。

重复进行如下过程：

- 交互库随机一个置换 g_1。
- 令 $H = (V, E')$，其中 $E' = \big\{\big(g_1(u), g_1(v)\big) \mid (u, v) \in E\big\}$。
- 让选手寻找置换 g_2 满足 $(u, v) \in E \iff \big(g_2(u), g_2(v)\big) \in E'$。
- 则此时必有 $g_1^{-1} g_2 \in \operatorname{Aut} G$。若 $S < \operatorname{Aut} G$，则 $g_1^{-1} g_2 \in S$ 的概率不超过 $\frac{1}{2}$。

于是重复充分多次即可完成 “证明”。

感谢大家的聆听，祝大家在冬令营取得优异的成绩！