

图论科研平台 Pro - 用户使用指南

版本: v6.1

最后更新: 2025-11-22

适用人群: 图论学习者、研究者、教师和学生

目录

第一部分: 快速入门

- 1. 欢迎使用
- 2. 快速开始
- 3. 五分钟教程

第二部分: 完整功能参考

- 4. 图的创建与编辑
- 5. 图生成器系统
- 6. 视图控制
- 7. 图论算法与参数
- 8. 算法可视化
- 9. 文件操作
- 10. 快捷键参考
- 11. 主题与个性化

第三部分: 附录

- 12. 常见问题 FAQ
- 13. 技术信息

第一部分：快速入门

1. 欢迎使用

1.1 平台简介

图论科研平台 Pro 是一个功能强大的纯前端图论可视化工具，专为图论学习、教学和科研设计。通过直观的交互界面，您可以轻松创建、编辑图，并实时查看各种图论算法的计算结果和可视化效果。

1.2 核心特色

- 🌟 零安装即用 - 直接在浏览器中打开 `index.html` 即可使用，无需安装任何软件或依赖
- 🎨 15 种图生成器 - 一键生成完全图、网格图、彼得森图等经典图结构
- 📊 15+ 种算法 - 自动计算直径、染色数、连通度、哈密顿性等图论参数
- 🕒 9 种可视化模式 - 直观展示算法结果，包括染色方案、最短路径、割点割边等
- ⌚ 专业数学渲染 - 支持 LaTeX 公式显示，符合学术规范
- 🌙 双主题支持 - 亮色/暗色主题自由切换
- ⚡ 高性能计算 - Web Worker 后台计算，不阻塞界面操作

1.3 适用场景

- 📚 图论课程教学 - 演示图论概念和算法原理
- 🔬 科研验证 - 构造反例、验证猜想
- 🎓 学生学习 - 通过可视化理解抽象的图论概念
- 📄 论文配图 - 生成规范的图结构插图

2. 快速开始

2.1 打开平台

方法一：本地使用

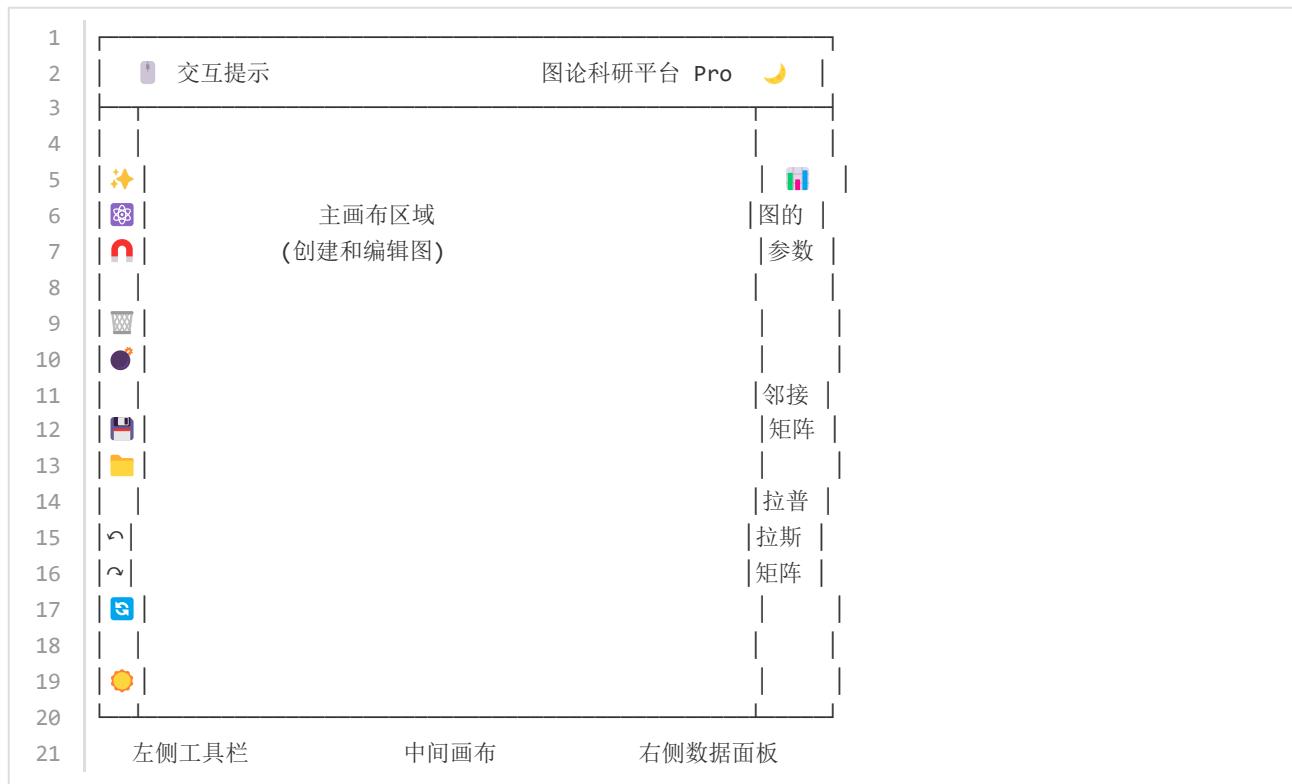
1. 下载项目文件夹
2. 双击打开 `index.html` 文件
3. 浏览器会自动打开平台（推荐使用 Chrome、Edge 或 Firefox）

方法二：在线使用

- 如果项目已部署到服务器，直接访问网址即可

2.2 界面总览

打开平台后，您会看到三个主要区域：



左侧工具栏 (72px 宽)

从上到下依次是：

- 交互提示 - 显示操作提示
- 图生成器 - 打开图生成器
- 物理布局 - 自动优化节点位置
- 智能吸附 - 节点自动对齐
- 删除选中 - 删除选中的对象
- 清空画布 - 删除所有内容
- 保存 - 保存图为文件
- 打开 - 加载已保存的图
- 撤销 - 撤销上一步操作
- 重做 - 重做下一步操作
- 重置视图 - 回到原点
- 主题切换 - 切换亮色/暗色主题

中间画布区域

- 白色（或深色）背景的绘图区域
- 支持拖拽、缩放、平移
- 左下角显示当前缩放比例

右侧数据面板 (380px 宽, 可折叠)

- **图的参数** - 显示顶点数、边数、度数、直径、染色数等
 - **邻接矩阵 A** - 显示邻接矩阵和特征值
 - **拉普拉斯矩阵 L** - 显示拉普拉斯矩阵和特征值
-

3. 五分钟教程

教程 1：手动创建一个简单图 (2 分钟)

目标： 创建一个三角形图 (K_3)

1. 创建第一个节点

- 在画布空白处双击鼠标左键
- 一个蓝色圆圈节点出现，带有弹性动画效果

2. 创建第二个节点

- 在画布另一处再次双击
- 第二个节点出现

3. 连接两个节点

- 按住键盘上的 Shift 键
- 鼠标光标会变成十字形 +
- 从第一个节点拖动到第二个节点
- 松开鼠标，一条黑色的边出现

4. 创建第三个节点并连线

- 按住 Shift 键
- 从第一个节点拖动到空白处
- 松开鼠标，自动创建第三个节点并连线

5. 完成三角形

- 按住 Shift 键
- 将第二个节点和第三个节点连接起来

6. 查看图的参数

- 观察右侧面板自动显示：
 - $|V| = 3$ (3 个顶点)
 - $|E| = 3$ (3 条边)
 - $\Delta = 2$ (最大度为 2)
 - $\text{diam}(G) = 1$ (直径为 1)
 - $\chi(G) = 3$ (需要 3 种颜色)

恭喜！您已经手动创建了第一个图。

教程 2：使用图生成器生成完全图 K_5 (1 分钟)

目标：一键生成 5 个顶点的完全图

1. 打开图生成器

- 点击左侧工具栏的  图标
- 或按快捷键（如果工具栏有提示）

2. 选择图类型

- 在弹出的图生成器窗口左侧
- 点击 "完全图 K_n "

3. 设置参数

- 右侧出现参数输入框： $n =$
- 输入数字 5

4. 生成图

- 点击底部的 "生成" 按钮
- 画布会清空（因为没有勾选"保留画布现有图"）
- 自动生成一个 5 个顶点的完全图，节点呈圆形排列

5. 查看结果

- 右侧面板显示：
 - $|V| = 5$ (5 个顶点)
 - $|E| = 10$ (10 条边, 完全图边数为 $n(n-1)/2$)
 - $\Delta = 4$ (每个顶点度数为 4)
 - $\chi(G) = 5$ (5 种颜色)

教程 3：使用算法可视化 (2 分钟)

目标：查看 K_5 的顶点染色方案

1. 等待计算完成

- 右侧面板中， $\chi(G)$ 旁边出现数字 5
- 如果显示 "计算中..."，请稍等片刻

2. 开启顶点染色可视化

- 找到右侧面板的 " $\chi(G) = 5$ "
- 点击右边的  眼睛图标

3. 观察可视化效果

- 画布上的 5 个节点分别显示不同颜色
- 每个节点右上角显示颜色编号 (1、2、3、4、5)
- 顶部出现紫色横幅提示："顶点染色可视化 | $\chi(G) = 5$ "

4. 退出可视化

- 再次点击  图标
- 或点击横幅右侧的  按钮

■ 节点恢复为默认蓝色

5. 尝试其他可视化

- 点击 $\text{diam}(G)$ 旁的  - 高亮显示直径路径 (紫色加粗)
- 点击 $\alpha(G)$ 旁的  - 高亮显示最大独立集 (红色节点)
- 点击 $\chi'(G)$ 旁的  - 边染色 (彩色边)

恭喜! 您已经掌握了平台的基本使用方法。

第二部分：完整功能参考

4. 图的创建与编辑

4.1 节点操作

4.1.1 创建节点

- 双击空白处 - 在鼠标位置创建新节点
- Shift + 拖拽到空白处 - 从现有节点拖动到空白处，自动创建新节点并连线

4.1.2 选择节点

- 单击节点 - 选中单个节点 (节点边框变为橙色)
- Ctrl + 单击 - 多选节点 (按住 Ctrl 依次点击多个节点)
- 框选 - 在空白处按住鼠标左键拖动，框选区域内的节点 (蓝色半透明矩形框)
- Ctrl + A - 全选所有节点

4.1.3 移动节点

- 拖拽节点 - 单击并拖动节点到新位置
- 拖拽多个节点 - 选中多个节点后，拖动任意一个，所有选中节点同步移动
- 智能吸附 - 开启  功能后，节点会自动对齐到其他节点或辅助线 (绿色虚线)

4.1.4 删除节点

- Del 或 Backspace - 删除选中的节点 (及其相关的边)
- 点击工具栏  - 删除选中的节点
- 右键菜单 → 删除 - 删除选中的节点

4.2 边操作

4.2.1 创建边

- **Shift + 拖拽** - 按住 Shift 键，从一个节点拖动到另一个节点
 - 光标会变成十字形 +
 - 拖动时显示虚线预览
 - 松开鼠标后创建边
- **自动避免重复边** - 如果两个节点间已有边，不会创建第二条边

4.2.2 选择边

- **单击边** - 选中单个边（边变为橙色加粗）
- **框选** - 框选会同时选中区域内的节点和边

4.2.3 删除边

- **Del 或 Backspace** - 删除选中的边
- **点击工具栏**  - 删除选中的边
- **右键菜单 → 删除** - 删除选中的边

4.3 编辑功能

4.3.1 撤销与重做

- **撤销** - Ctrl + Z 或点击 ⌘ 图标
 - 最多保存 30 步历史记录
 - 撤销操作包括：创建、删除、移动节点/边
- **重做** - Ctrl + Y 或 Ctrl + Shift + Z 或点击 ⌘ 图标

4.3.2 复制、粘贴、剪切

- **复制** - Ctrl + C 或右键菜单 → 复制
 - 复制选中的节点及其之间的边
- **粘贴** - Ctrl + V 或右键菜单 → 粘贴
 - 粘贴到鼠标位置附近
 - 保持节点间的相对位置关系
 - 边也会一并复制
- **剪切** - Ctrl + X 或右键菜单 → 剪切
 - 剪切 = 复制 + 删除
- **副本** - Ctrl + D 或右键菜单 → 副本
 - 快捷操作：复制并立即粘贴

4.3.3 清空画布

- 点击工具栏  图标 - 删除所有节点和边
- 确认提示 - 清空前会弹出确认对话框 (如果有内容)

4.4 右键上下文菜单

在画布上右键点击会出现上下文菜单，根据当前状态显示不同选项：

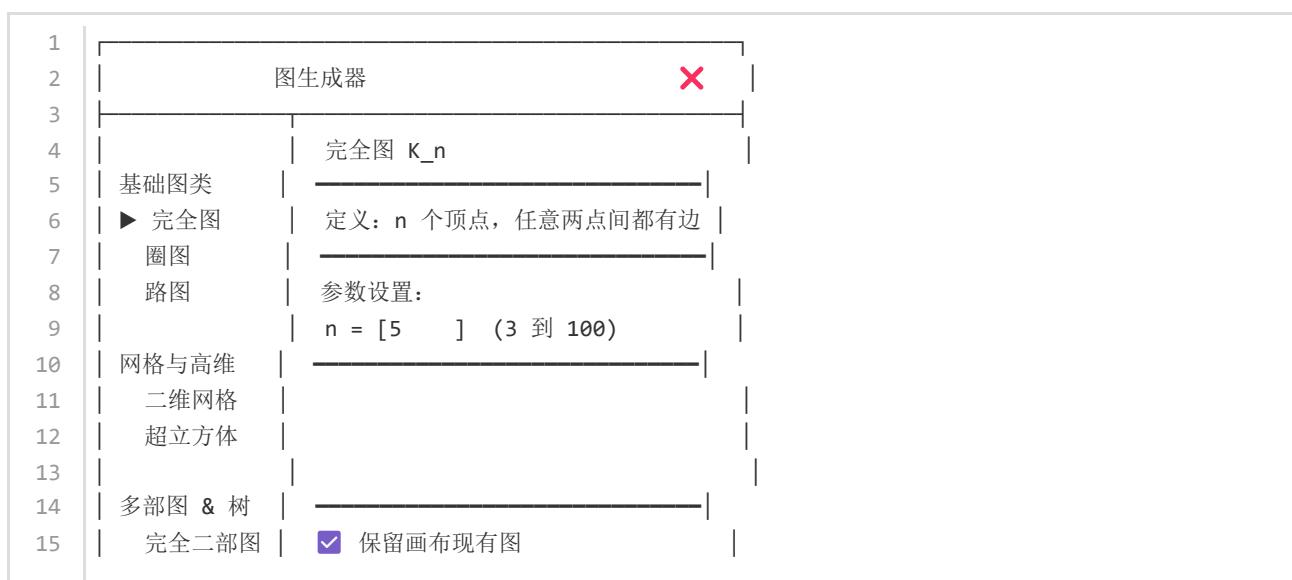
条件	菜单项
有选中对象	 复制 (Ctrl+C)
有选中对象	 副本 (Ctrl+D)
有选中对象	 剪切 (Ctrl+X)
有选中对象	 删除 (Del)
有剪贴板内容	 粘贴 (Ctrl+V)

5. 图生成器系统

5.1 打开图生成器

- 点击工具栏  图标
- 弹出图生成器模态框

5.2 图生成器界面



16	星图	[取消]	[生成]
17	满二叉树		
18			

5.3 图类型分类

5.3.1 基础图类

图名称	符号	参数	说明
完全图	K_n	n: 顶点数 (3-100)	任意两点间都有边, 共 $n(n-1)/2$ 条边
圈图	C_n	n: 顶点数 (3-100)	n 个顶点闭合成环路
路图	P_n	n: 顶点数 (2-100)	n 个顶点线性排列

5.3.2 网格与高维结构

图名称	符号	参数	说明
二维网格	G_{m,n}	m: 行数 (2-20) n: 列数 (2-20)	$m \times n$ 矩形网格, 最多 400 个节点
超立方体	Q_n	n: 维度 (1-6)	n 维超立方, 2^n 个顶点

5.3.3 多部图 & 树

图名称	符号	参数	说明
完全二部图	K_{n,m}	n: 左侧顶点数 (1-50) m: 右侧顶点数 (1-50)	两组独立集之间全连接
星图	S_n	n: 叶子节点数 (1-100)	1 个中心节点 + n 个叶子节点
满二叉树	T_h	h: 深度 (1-7)	深度为 h 的完美二叉树, $2^{(h+1)-1}$ 个节点

5.3.4 随机图模型

图名称	符号	参数	说明
Erdős-Rényi 随机图	G_{n,p}	n: 顶点数 (2-100) p: 连边概率 (0-1)	每对顶点以概率 p 连边

5.3.5 特殊构造 & 反例

图名称	参数	说明
轮图	n: 外圈顶点数 (3-100)	圈图 C_n + 中心点
彼得森图	无参数	经典的 (3,5)-笼图, 10 个顶点, 15 条边

图名称	参数	说明
格勒奇图	无参数	最小无三角形 4-色图, 11 个顶点
希伍德图	无参数	(3,6)-笼图, 14 个顶点, 21 条边
图兰图	n: 顶点数 (2-100) r: 部数 (1-10)	不含 $K_{\{r+1\}}$ 的极值图

5.4 使用步骤

1. **选择图类型** - 点击左侧列表中的图名称
2. **设置参数** - 在右侧输入框中输入参数值
 - 输入框会显示合法范围 (如 "3 到 100")
 - 超出范围会自动限制
3. **选择生成模式**
 - 勾选"保留画布现有图" - 在当前图上追加新图 (不清空画布)
 - 不勾选 - 清空画布后生成新图 (默认)
4. **点击"生成"** - 生成图并自动关闭模态框

5.5 生成后的效果

- 图自动定位到画布中心
- 节点按照特定布局排列 (圆形、网格、树形等)
- 右侧面板自动开始计算图的参数
- 可以继续手动编辑生成的图

5.6 追加模式示例

场景: 在 K_5 的基础上添加一个 C_4

1. 生成 K_5 (不勾选"保留画布现有图")
2. 再次打开图生成器
3. 选择"圈图 C_n ", 输入 $n=4$
4. **勾选 "保留画布现有图"**
5. 点击"生成"
6. 结果: 画布上同时存在 K_5 和 C_4 (两个不相交的图)

6. 视图控制

6.1 缩放操作

■ 滚轮缩放

- 向上滚动：放大 (Zoom In)
- 向下滚动：缩小 (Zoom Out)
- 缩放范围： $0.1x \sim 5x$
- 缩放中心：鼠标位置

■ 缩放指示器

- 左下角显示当前缩放比例 (如 "100%")
- 实时更新

6.2 平移操作

- **中键拖拽** - 按住鼠标中键 (滚轮) 拖动画布
- **空格 + 拖拽** - 按住空格键，鼠标左键拖动画布
- **拖动时光标变为抓手** 🤲

6.3 重置视图

- **按键盘 0 (数字零)** - 回到原点，缩放比例重置为 100%
- **点击工具栏 重置图标** - 同样效果

6.4 物理引擎布局

6.4.1 开启/关闭

- **点击工具栏 物理图标** - 切换物理布局开关
- **图标颜色变化**
 - 深色：已关闭
 - 高亮：已开启

6.4.2 工作原理

- **力导向算法** - 基于弹簧引力 + 库仑斥力模型
 - 相邻节点之间有吸引力 (弹簧)
 - 所有节点之间有排斥力 (电荷)
 - 有中心引力防止节点漂移

- **实时动画** - 节点自动移动到最优位置
 - 图变得更美观、对称
 - 减少边的交叉

6.4.3 使用建议

- 在创建复杂图后开启，让节点自动排列
- 如果需要精确控制节点位置，建议关闭
- 物理引擎会持续运行直到节点稳定

6.5 智能吸附功能

6.5.1 开启/关闭

- 点击工具栏  图标 - 切换智能吸附开关

6.5.2 吸附类型

1. 水平/垂直对齐

- 拖动节点时，如果接近其他节点的 X 或 Y 坐标
- 显示绿色虚线辅助线
- 自动对齐

2. 连接边中点吸附

- 拖动节点时，如果有两个邻居节点
- 自动对齐到两个邻居的中点位置

3. 线性投影吸附

- 对齐到两个邻居节点连线的投影点
- 保持三个节点共线

6.5.3 视觉反馈

- **绿色虚线** - 显示对齐参考线
- **自动吸附** - 节点自动移动到对齐位置

7. 图论算法与参数

7.1 右侧面板总览

右侧数据面板分为三个卡片：

1. **图的参数**  - 基础信息、度数统计、结构特征、连通性
2. **邻接矩阵 A** - 矩阵显示 + 特征值列表

3. 拉普拉斯矩阵 L - 矩阵显示 + 特征值列表

每个卡片都可以点击标题栏展开/折叠。

7.2 基础信息

参数	符号	说明	复杂度
顶点数	$ V $	图中节点的数量	$O(1)$
边数	$ E $	图中边的数量	$O(1)$

7.3 度数统计

参数	符号	说明	复杂度
最大度	Δ (Delta)	所有节点中度数最大的值	$O(V)$
最小度	δ (delta)	所有节点中度数最小的值	$O(V)$
平均度	\bar{d} (d-bar)	所有节点度数的平均值, 等于 $2 E / V $	$O(V)$

7.4 结构特征

7.4.1 直径 diam(G)

- **定义** - 图中任意两点间最短路径的最大值
- **算法** - BFS (广度优先搜索)
- **复杂度** - $O(V^2)$
- **可视化** - 点击  按钮, 高亮显示直径路径 (紫色加粗)

示例

- 完全图 K_n : 直径 = 1 (任意两点间距离为 1)
- 路图 P_n : 直径 = $n-1$ (从一端到另一端)
- 圈图 C_n : 直径 = $\lfloor n/2 \rfloor$

7.4.2 顶点染色数 $\chi(G)$

- **定义** - 给节点染色, 相邻节点颜色不同, 所需的最少颜色数
- **算法** - DSatur 启发式算法 (贪心 + 饱和度优先)
- **复杂度** - $O(V^2)$
- **可视化** - 点击  按钮, 节点显示彩色, 右上角显示颜色编号

示例

- 完全图 K_n : $\chi(G) = n$ (每个节点需要不同颜色)
- 二部图: $\chi(G) = 2$ (两种颜色即可)
- 圈图 C_n : $\chi(G) = 2$ (n 为偶数) 或 3 (n 为奇数)

7.4.3 边染色数 $\chi'(G)$

- **定义** - 给边染色, 相邻边 (共享节点的边) 颜色不同, 所需的最少颜色数
- **算法** - 回溯搜索 + 位掩码优化
- **复杂度** - 指数级 ($K \leq 31$ 时可计算)
- **限制** - 如果最大度 $\Delta > 31$, 显示 "-" (无法计算)
- **可视化** - 点击 按钮, 边显示彩色, 边中点显示颜色编号

Vizing 定理

- $\Delta \leq \chi'(G) \leq \Delta + 1$
- 简单图的边染色数要么等于最大度, 要么等于最大度 +1

7.4.4 独立数 $\alpha(G)$

- **定义** - 最大独立集的大小 (独立集: 节点集合中任意两点不相邻)
- **算法** - Bron-Kerbosch 最大团算法 (在补图上求最大团)
- **复杂度** - $O(3^{(V/3)})$
- **可视化** - 点击 按钮, 独立集节点显示为红色, 带橙色虚线边框

示例

- 完全图 K_n : $\alpha(G) = 1$ (任意两点都相邻)
- 空图 (无边) : $\alpha(G) = n$ (所有节点都独立)
- 二部图 $K_{\{n,m\}}$: $\alpha(G) = \max(n, m)$

7.4.5 两点距离 $dist(u,v)$

- **前提** - 选中恰好 2 个节点时才显示
- **定义** - 两点间最短路径的长度
- **算法** - BFS
- **复杂度** - $O(V+E)$
- **可视化** - 点击 按钮, 高亮显示最短路径 (绿色加粗)
- **不连通** - 如果两点不连通, 显示 " ∞ "

7.4.6 哈密顿路径 (Ham. Path)

- **定义** - 经过每个顶点恰好一次的路径
- **算法** - 回溯搜索 + 位掩码 + 启发式剪枝
- **复杂度** - 指数级 ($N \leq 24$ 时可计算)
- **显示**
 - "Yes" - 存在哈密顿路径

- "No" - 不存在
- "Slow" - 图太大 ($N > 24$) , 跳过计算
- **可视化** - 点击  按钮 (仅在 "Yes" 时可用) , 高亮显示哈密顿路径 (紫色加粗)

7.4.7 哈密顿圈 (Ham. Cycle)

- **定义** - 经过每个顶点恰好一次并回到起点的闭合路径
- **算法** - 回溯搜索 + 位掩码 + 割边预检测
- **复杂度** - 指数级 ($N \leq 24$ 时可计算)
- **显示**
 - "Yes" - 存在哈密顿圈
 - "No" - 不存在
 - "Slow" - 图太大 ($N > 24$) , 跳过计算
- **可视化** - 点击  按钮 (仅在 "Yes" 时可用) , 高亮显示哈密顿圈 (紫色加粗闭合路径)

Dirac 定理

- 如果 $n \geq 3$ 且 $\delta \geq n/2$, 则图有哈密顿圈

7.5 连通性分析

7.5.1 顶点连通度 κ (kappa)

- **定义** - 最少需要删除多少个顶点才能使图不连通 (或变为单点图)
- **算法** - 最大流算法 (顶点拆分技术)
- **复杂度** - $O(V^3)$
- **限制** - $N > 50$ 时显示 "Slow" (跳过计算)
- **可视化** - 点击  按钮, 高亮显示最小割点集 (橙色闪烁节点)

特殊值

- $\kappa = 0$: 图不连通
- $\kappa = 1$: 存在割点 (删除该点后图不连通)
- $\kappa = n-1$: 完全图 K_n

7.5.2 边连通度 λ (lambda)

- **定义** - 最少需要删除多少条边才能使图不连通
- **算法** - Stoer-Wagner 最小割算法
- **复杂度** - $O(V^3)$
- **限制** - $N > 50$ 时显示 "Slow" (跳过计算)
- **可视化** - 点击  按钮, 高亮显示最小割边集 (橙色闪烁边)

关系

- $\kappa \leq \lambda \leq \delta$ (顶点连通度 \leq 边连通度 \leq 最小度)

7.6 邻接矩阵 A

7.6.1 定义

- **邻接矩阵** $A = (a_{ij})$, 其中:
 - $a_{ij} = 1$ 如果节点 i 和 j 相邻
 - $a_{ij} = 0$ 否则

7.6.2 显示

- **矩阵显示** - 实时渲染, 非零元素高亮显示 (蓝色背景)
- **特征值列表** - 按降序排列, 带重数标记
 - 例如: "2.00 ($\times 2$), 1.00, -1.00 ($\times 2$)"

7.6.3 图论意义

- A^2 的 (i,j) 元素 - 从节点 i 到节点 j 长度为 2 的路径数量
- $\text{tr}(A) = 0$ - 迹为 0 (无自环)
- **最大特征值** - 与图的连通性相关

7.7 拉普拉斯矩阵 L

7.7.1 定义

- **拉普拉斯矩阵** $L = D - A$, 其中:
 - D 是度数矩阵 (对角矩阵, $D_{ii} = \deg(i)$)
 - A 是邻接矩阵

7.7.2 显示

- **矩阵显示** - 实时渲染
- **特征值列表** - 按降序排列

7.7.3 图论意义

- **L 是半正定矩阵** - 所有特征值 ≥ 0
- **最小特征值 = 0** - 重数等于连通分量数
- **第二小特征值 (代数连通度)** - 衡量图的连通性, 越大越连通
- **Kirchhoff 定理** - 生成树数量与 L 的余子式相关

8. 算法可视化

8.1 可视化总览

平台支持 9 种算法可视化模式，通过点击右侧面板中算法旁的  图标开启。

8.2 可视化列表

编号	算法	触发位置	视觉效果	说明
1	顶点染色	$\chi(G)$ 旁 	节点彩色 + 右上角数字徽章	DSatur 算法的染色方案
2	边染色	$\chi'(G)$ 旁 	边彩色 + 中点数字徽章	回溯算法的染色方案
3	直径路径	$diam(G)$ 旁 	紫色加粗路径	高亮最长最短路径
4	距离路径	$dist(u,v)$ 旁 	绿色加粗路径	高亮两点间最短路径
5	最大独立集	$\alpha(G)$ 旁 	红色节点 + 橙色虚线边框	高亮独立集节点
6	顶点连通度	κ 旁 	橙色闪烁节点	高亮最小割点集
7	边连通度	λ 旁 	橙色闪烁边	高亮最小割边集
8	哈密顿路径	Ham. Path 旁 	紫色加粗路径	高亮哈密顿路径
9	哈密顿圈	Ham. Cycle 旁 	紫色加粗闭合路径	高亮哈密顿回路

8.3 可视化操作

8.3.1 开启可视化

- 等待算法计算完成（显示具体数值，不是 "计算中..."）
- 点击算法旁的  图标
- 画布显示可视化效果
- 顶部出现紫色横幅提示

8.3.2 退出可视化

- 方法 1 - 再次点击同一个  图标
- 方法 2 - 点击横幅右侧的  按钮
- 方法 3 - 点击其他算法的  图标（切换到新的可视化）

8.3.3 可视化横幅

1	顶点染色可视化 $\chi(G) = 5$	X
2		
3		

- 左侧 - 可视化模式名称 + 结果值
- 右侧 - X 关闭按钮

8.4 可视化效果详解

8.4.1 顶点染色

- **节点颜色** - 每个节点显示不同颜色 (根据染色方案)
- **数字徽章** - 节点右上角显示颜色编号 (1, 2, 3, ...)
- **颜色方案** - 使用固定的颜色序列 (红、绿、蓝、黄、紫...)

8.4.2 边染色

- **边颜色** - 每条边显示不同颜色
- **数字徽章** - 边中点显示颜色编号
- **颜色方案** - 与顶点染色相同

8.4.3 路径高亮 (直径、距离、哈密顿)

- **边加粗** - 路径上的边宽度增加到 5px
- **颜色**
 - 直径路径: 紫色 (#9333ea)
 - 距离路径: 绿色 (#22c55e)
 - 哈密顿路径/圈: 紫色 (#9333ea)

8.4.4 节点高亮 (独立集、割点)

- **节点颜色** - 红色 (独立集) 或橙色 (割点)
- **闪烁效果** - 割点会有呼吸灯效果 (CSS 动画)
- **边框** - 独立集节点带橙色虚线边框

8.4.5 边高亮 (割边)

- **边颜色** - 橙色 (#f97316)
- **闪烁效果** - 呼吸灯动画

8.5 可视化注意事项

- **互斥性** - 同一时间只能开启一个可视化模式
- **计算前提** - 算法未计算完成时，图标不可点击（灰色）
- **哈密顿可视化** - 仅在结果为 "Yes" 时可用
- **距离可视化** - 仅在选中恰好 2 个节点时可用

9. 文件操作

9.1 保存图

9.1.1 操作方法

- **方法 1** - 点击工具栏 图标
- **方法 2** - 按快捷键 Ctrl + S

9.1.2 保存格式

- **文件格式** - JSON
- **文件名** - graph_<时间戳>.json
 - **示例**: graph_1732262400000.json

9.1.3 文件内容

```
1  {
2      "version": "5.5",
3      "timestamp": "2025-11-22T08:00:00.000Z",
4      "nodes": [
5          {"id": 1, "x": 400, "y": 300},
6          {"id": 2, "x": 500, "y": 300},
7          {"id": 3, "x": 450, "y": 400}
8      ],
9      "edges": [
10         {"s": 1, "t": 2},
11         {"s": 2, "t": 3},
12         {"s": 3, "t": 1}
13     ],
14     "nextId": 4
15 }
```

9.1.4 字段说明

- **version** - 数据格式版本号（当前为 5.5）
- **timestamp** - 保存时间（ISO 8601 格式）
- **nodes** - 节点数组
 - **id** - 节点 ID（唯一标识）

- `x, y` - 节点坐标
- `edges` - 边数组
 - `s` - 起点节点 ID
 - `t` - 终点节点 ID
- `nextId` - 下一个可用的节点 ID

9.2 打开图

9.2.1 操作方法

- **方法 1** - 点击工具栏  图标
- **方法 2** - 按快捷键 `Ctrl + O`

9.2.2 打开步骤

1. 点击打开按钮
2. 浏览器弹出文件选择对话框
3. 选择 `.json` 文件
4. 点击"打开"

9.2.3 加载效果

- **清空画布** - 删除当前所有节点和边
- **加载新图** - 根据 JSON 文件创建节点和边
- **重置视图** - 自动回到原点 $(0, 0)$
- **重新计算** - 右侧面板自动重新计算所有参数

9.2.4 文件验证

- **格式检查** - 自动验证 JSON 格式是否正确
- **版本兼容** - 支持旧版本数据格式（向下兼容）
- **错误提示** - 如果文件格式错误，显示错误信息

9.3 文件管理建议

- **命名规范** - 建议手动重命名文件为有意义的名称
 - 例如：`K5_complete_graph.json`、`peterson_graph.json`
- **备份重要图** - 复杂的图建议多次保存
- **版本控制** - 可以使用 Git 管理图文件

10. 快捷键参考

10.1 编辑操作

快捷键	功能	说明
Ctrl + A	全选	选中所有节点
Ctrl + C	复制	复制选中的节点和边
Ctrl + V	粘贴	粘贴到鼠标位置附近
Ctrl + X	剪切	复制并删除选中对象
Ctrl + D	副本	复制并立即粘贴
Del	删除	删除选中的节点/边
Backspace	删除	同 Del (macOS 用户习惯)
Ctrl + Z	撤销	撤销上一步操作 (最多 30 步)
Ctrl + Y	重做	重做下一步操作
Ctrl + Shift + Z	重做	同 Ctrl+Y (替代方案)

10.2 文件操作

快捷键	功能	说明
Ctrl + S	保存	保存图为 JSON 文件
Ctrl + O	打开	加载 JSON 文件

10.3 视图操作

操作	功能	说明
滚轮向上	放大	以鼠标位置为中心放大
滚轮向下	缩小	以鼠标位置为中心缩小
中键拖拽	平移	拖动画布
空格 + 拖拽	平移	按住空格, 左键拖动画布
0 (数字零)	重置视图	回到原点, 缩放比例 100%

10.4 交互模式

按键/操作	模式	说明
Shift	连线模式	按住 Shift, 光标变为十字, 拖拽连线
Ctrl	多选模式	按住 Ctrl, 点击多个节点进行多选
Esc	取消	退出模态框 / 清空选区
双击空白处	创建节点	在鼠标位置创建新节点
右键	上下文菜单	打开右键菜单

10.5 快捷键速查表

打印以下表格作为速查:

1	快捷键速查表	
2	编辑操作	Ctrl+A/C/V/X/D/Z/Y Del
3	文件操作	Ctrl+S Ctrl+O
4	视图控制	滚轮缩放 中键/空格拖拽 按 Ø 重置
5	创建/连线	双击创建 Shift拖拽连线
6	多选	Ctrl点击 或 框选

11. 主题与个性化

11.1 主题切换

11.1.1 操作方法

- 点击工具栏底部的 或 图标
 - 当前为亮色主题, 点击切换到暗色
 - 当前为暗色主题, 点击切换到亮色

11.1.2 主题效果

亮色主题 (默认)

- 背景: 白色 (#ffffff)
- 文字: 深灰色 (#374151)
- 节点: 蓝色 (#3b82f6)
- 边: 黑色 (#1f2937)

- 面板：浅灰色背景 (#f9fafb)

暗色主题

- 背景：深色 (#1a1a2e)
- 文字：浅灰色 (#e5e7eb)
- 节点：亮蓝色 (#60a5fa)
- 边：浅灰色 (#9ca3af)
- 面板：深灰色背景 (#16213e)

11.1.3 自动保存

- **本地存储** - 主题偏好自动保存到浏览器 localStorage
- **下次打开** - 自动应用上次选择的主题

11.2 界面自定义

11.2.1 折叠面板

- **点击卡片标题栏** - 展开/折叠该卡片
 - 图的参数 (默认展开)
 - 邻接矩阵 A (默认折叠)
 - 拉普拉斯矩阵 L (默认折叠)

11.2.2 隐藏/显示右侧面板

- 目前版本不支持完全隐藏右侧面板
 - 建议：缩小浏览器窗口或使用缩放功能
-

第三部分：附录

12. 常见问题 FAQ

12.1 性能相关

Q1：为什么有些算法显示 "Slow" 或 "计算中..."？

A：某些图论算法的时间复杂度很高（指数级），为了不阻塞界面，平台设置了以下限制：

算法	限制	原因
顶点/边连通度	$N > 50$ 跳过	$O(V^3)$ 复杂度
哈密顿路径/圈	$N > 24$ 跳过	指数级回溯
边染色	$\Delta > 31$ 跳过	位掩码限制 (32 位整数)

建议:

- 对于大图，这些算法确实难以在合理时间内计算
- 如果需要这些参数，建议简化图（删除部分节点）

Q2：创建了很大的图（如 100 个节点），为什么很卡？

A：大图会导致以下性能问题：

- **渲染压力** - 100 个节点 + 数千条边的绘制
- **算法计算** - 连通性、哈密顿性等算法计算量大
- **物理引擎** - 力导向布局每帧计算所有节点的受力

优化建议:

1. **关闭物理引擎** - 点击  图标关闭自动布局
2. **关闭不必要的可视化** - 退出当前可视化模式
3. **避免创建超大图** - 推荐节点数 < 50

Q3：计算卡住不动怎么办？

A：

1. **等待计算完成** - 某些算法（如哈密顿性）可能需要几秒到几十秒
2. **刷新页面** - 如果确实卡死（超过 1 分钟无响应），刷新浏览器
3. **查看控制台** - 按 F12 打开浏览器开发者工具，查看是否有错误信息

12.2 操作技巧

Q4：如何精确对齐节点？

A：

1. **开启智能吸附** - 点击  图标
2. **拖动节点** - 节点会自动对齐到其他节点的 X 或 Y 坐标
3. **观察绿色虚线** - 虚线表示对齐参考线

高级技巧：

- 先创建一个"参考节点"作为基准
 - 依次拖动其他节点对齐到参考节点
-

Q5：如何避免创建重复的边？

A:

- **自动检测** - 平台会自动检测重复边，不会创建第二条边
 - **视觉反馈** - 如果尝试创建重复边，虚线预览会消失，松开鼠标后不会创建边
-

Q6：如何创建不连通的图（多个连通分量）？

A:

1. **方法 1：手动创建** - 在不同位置双击创建节点，不要连线
 2. **方法 2：使用追加模式**
 - 先生成第一个图（如 K_3 ）
 - 打开图生成器，勾选"保留画布现有图"
 - 生成第二个图（如 C_4 ）
 - 结果：画布上有两个不相交的图
-

Q7：如何快速创建规则图形（如正方形、正五边形）？

A:

- **使用图生成器** - 生成圈图 C_n
 - C_4 = 正方形
 - C_5 = 正五边形
 - C_6 = 正六边形
 - **手动微调** - 生成后可以手动拖动节点调整位置
-

12.3 算法说明

Q8：顶点染色数和边染色数有什么区别？

A:

- **顶点染色** $\chi(G)$ - 给节点染色，相邻节点颜色不同
 - 应用：寄存器分配、排课问题

- **边染色** $\chi'(G)$ - 给边染色，相邻边（共享节点）颜色不同
 - 应用：时间片分配、通道分配

关系：

- $\Delta \leq \chi'(G) \leq \Delta + 1$ (Vizing 定理)
 - $\chi(G)$ 和 $\chi'(G)$ 没有直接关系
-

Q9：什么是哈密顿路径和哈密顿圈？

A:

- **哈密顿路径** - 经过每个顶点恰好一次的路径
- **哈密顿圈** - 经过每个顶点恰好一次并回到起点的闭合路径

判定：

- **困难性** - 判定一个图是否有哈密顿路径/圈是 NP-完全问题
- **充分条件** - Dirac 定理：如果 $n \geq 3$ 且 $\delta \geq n/2$, 则图有哈密顿圈

示例：

- K_n (完全图) - 总是有哈密顿圈
 - C_n (圈图) - 本身就是哈密顿圈
 - P_n (路图) - 有哈密顿路径，无哈密顿圈
-

Q10：顶点连通度和边连通度的区别？

A:

- **顶点连通度 κ** - 最少需要删除多少个顶点才能使图不连通
 - 删除的是节点（及其相关的边）
- **边连通度 λ** - 最少需要删除多少条边才能使图不连通
 - 只删除边

关系：

- $\kappa \leq \lambda \leq \delta$
 - 例如：星图 S_n 的 $\kappa=1$ (删除中心点), $\lambda=1$ (删除任意一条边), $\delta=1$
-

Q11：什么是独立集？

A:

- **独立集** - 节点集合，其中任意两个节点都不相邻
- **最大独立集** - 大小最大的独立集
- **独立数 $a(G)$** - 最大独立集的大小

应用：

- 调度问题（选择不冲突的任务）
- 顶点覆盖问题 ($|V| - \alpha(G)$ = 最小顶点覆盖数)

示例：

- K_n 的 $\alpha(G) = 1$ （任意两点都相邻）
 - 空图的 $\alpha(G) = n$ （所有节点都独立）
-

12.4 文件与兼容性

Q12：保存的 JSON 文件可以在其他软件中使用吗？

A:

- **格式简单** - JSON 格式非常通用，可以被大多数编程语言解析
- **自定义格式** - 但文件结构是平台专用的，其他图论软件可能不直接兼容
- **数据提取** - 可以编写脚本提取 nodes 和 edges 数据，转换为其他格式

建议：

- 如果需要导出到其他软件，手动转换 JSON 数据
 - 或者使用截图/录屏功能保存可视化结果
-

Q13：平台支持导入其他格式的图吗（如 GraphML、DOT）？

A:

- **当前版本** - 仅支持平台自己的 JSON 格式
 - **未来计划** - 可能增加对 GraphML、DOT、Edge List 等格式的支持
-

12.5 其他问题

Q14：如何导出图的图片？

A:

- **方法 1：浏览器截图**
 - Windows: Win + Shift + S (截图工具)
 - Mac: Cmd + Shift + 4
 - 或使用浏览器扩展（如 Awesome Screenshot）

■ 方法 2：全屏截图

- 按 F11 进入浏览器全屏模式
- 截图后退出全屏

未来功能：

- 可能增加“导出为 PNG/SVG”功能
-

Q15：平台是否需要联网使用？

A:

- **依赖库** - 平台使用了外部 CDN 库 (Math.js、KaTeX、Font Awesome)
- **首次加载** - 需要联网下载这些库 (约 1-2 MB)
- **浏览器缓存** - 下载后浏览器会缓存，后续可离线使用

完全离线使用：

- 下载依赖库到本地，修改 index.html 中的 CDN 链接为本地路径
-

13. 技术信息

13.1 浏览器兼容性

推荐浏览器

- Google Chrome 90+ (推荐)
- Microsoft Edge 90+ (推荐)
- Firefox 88+
- Safari 14+

不支持

- Internet Explorer - 不支持 (IE 已停止维护)
- 旧版浏览器 (2020 年前的版本)

所需浏览器特性

- Canvas API (绘图)
 - Web Workers (后台计算)
 - ES6+ (箭头函数、类、模板字符串等)
 - CSS Variables (主题切换)
 - LocalStorage (保存主题偏好)
-

13.2 外部依赖库

库名称	版本	用途	CDN 链接
Math.js	v11.8.0	矩阵运算、特征值计算	https://unpkg.com/mathjs@11.8.0/
KaTeX	v0.16.8	LaTeX 数学公式渲染	https://cdn.jsdelivr.net/npm/katex@0.16.8/
Font Awesome	v6.0	图标库	https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/

无需 npm 安装 - 所有依赖通过 CDN 加载，无需本地安装。

13.3 项目信息

- **项目名称** - 图论科研平台 Pro
- **版本** - v6.1
- **代码量** - 约 2,269 行 (不含依赖库)
- **文件大小** - 约 128 KB
- **开发语言** - HTML5 + CSS3 + JavaScript (ES6+)
- **架构** - 纯前端，无需后端服务器

文件结构

```
1 graph-platform/
2   └── index.html      # 主页面
3   └── css/
4     └── styles.css    # 样式表
5   └── js/
6     └── app.js         # 核心应用逻辑
7     └── config.js     # 图生成器配置
8     └── worker.js      # 算法计算引擎 (Web Worker)
9   └── docs/
10    └── README.md      # 项目说明文档
11    └── USER_GUIDE.md  # 用户使用指南 (本文档)
```

13.4 开源许可

- **许可证** - 请查看项目根目录的 LICENSE 文件
- **贡献** - 欢迎提交 Issue 和 Pull Request

13.5 联系方式

- **问题反馈** - 请通过项目 GitHub/GitLab 提交 Issue
- **功能建议** - 欢迎在 Issue 中提出新功能建议

附录：术语表

术语	英文	定义
顶点	Vertex	图中的节点
边	Edge	连接两个节点的线段
度	Degree	一个节点连接的边的数量
路径	Path	一系列节点的序列，相邻节点间有边
圈/环	Cycle	起点和终点相同的闭合路径
连通	Connected	图中任意两点间都存在路径
连通分量	Connected Component	图的最大连通子图
割点	Cut Vertex	删除后使图不连通的节点
割边	Cut Edge / Bridge	删除后使图不连通的边
独立集	Independent Set	节点集合，其中任意两点不相邻
团	Clique	节点集合，其中任意两点都相邻（完全子图）
二部图	Bipartite Graph	节点可分为两组，边只连接不同组的节点
平面图	Planar Graph	可以在平面上绘制且边不相交的图
正则图	Regular Graph	所有节点度数相同的图

结语

感谢您使用 **图论科研平台 Pro!**

本指南覆盖了平台的所有功能和操作方法。如果您在使用过程中遇到问题，请参考：

1. **常见问题 FAQ** (第 12 节)
2. **快捷键参考** (第 10 节)
3. **五分钟教程** (第 3 节)

祝您使用愉快，研究顺利！

版本历史

- v1.0 (2025-11-22) - 初始版本，覆盖平台 v6.1 的所有功能

文档维护

- 如发现文档错误或需要更新，请提交 Issue

本文档由 *Claude Code* 生成，基于对 *graph-platform* 项目的详细分析。