Design and Analysis of Algorithms

Part IV: Graph Algorithms

Lecture 20: Basic Concepts in Graphs

童咏昕

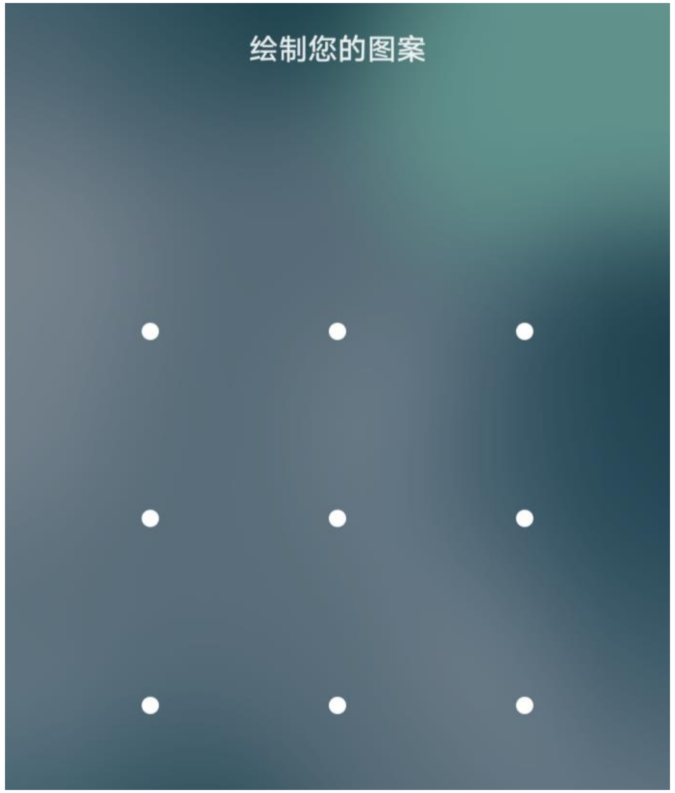
北京航空航天大学
计算机学院

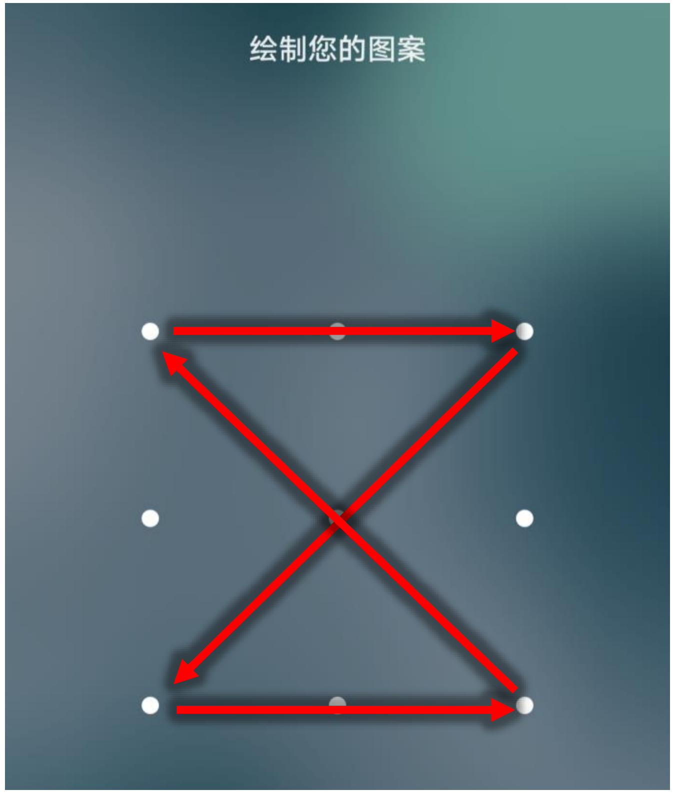
图算法篇概述

- 在算法课程第四部分“图算法”主题中，我们将主要聚焦于如下经典问题：
  - Basic Concepts in Graph Algorithms（图算法的基本概念）
  - Breadth-First Search (BFS，广度优先搜索）
  - Depth-First Search (DFS，深度优先搜索）
  - Cycle Detection（环路检测）
  - Topological Sort （拓扑排序）
  - Strongly Connected Components（强连通分量）
  - Minimum Spanning Trees (最小生成树)
  - Single Source Shortest Path (单源最短路径)
  - All-Pairs Shortest Paths (所有点对最短路径)
  - Bipartite Graph Matching（二分图匹配）
  - Maximum/Network Flows (最大流/网络流)

图的背景

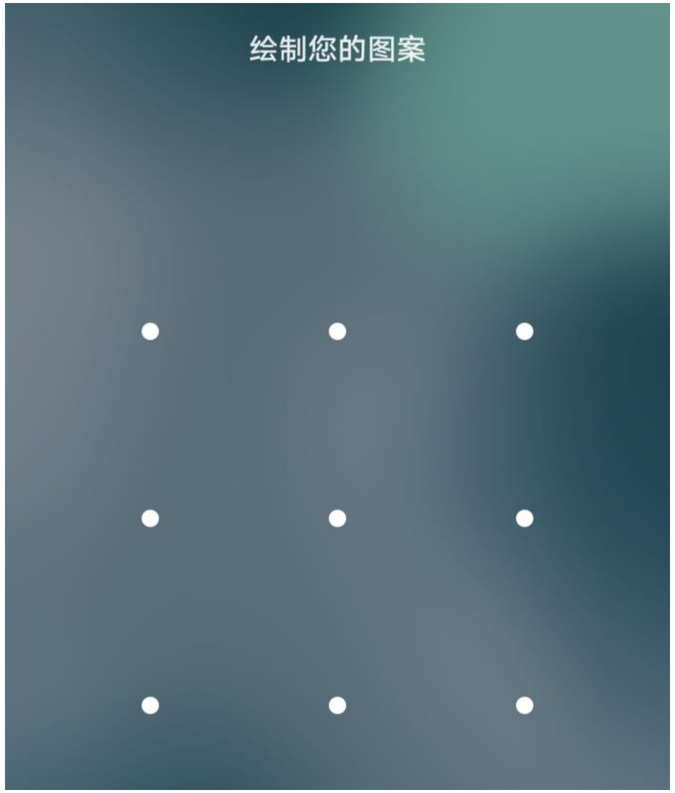
- 一笔画问题：手机解锁图案需一笔画出

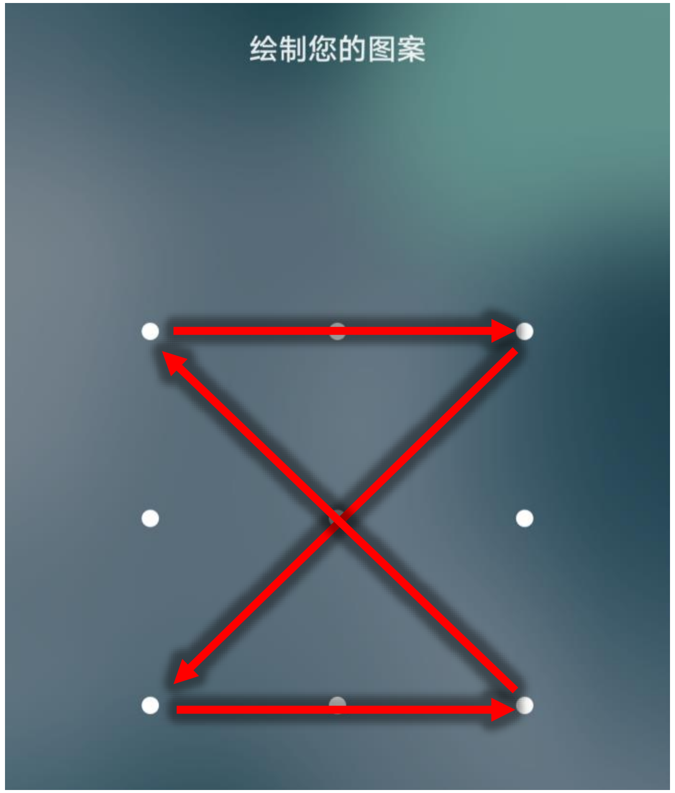




图的背景

- 一笔画问题：手机解锁图案需一笔画出

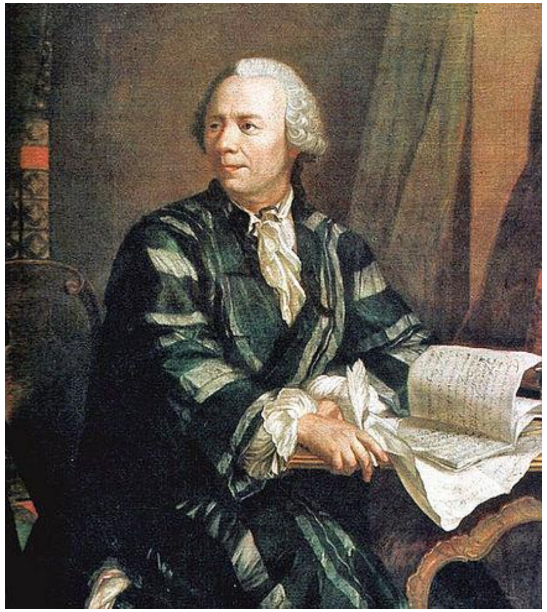




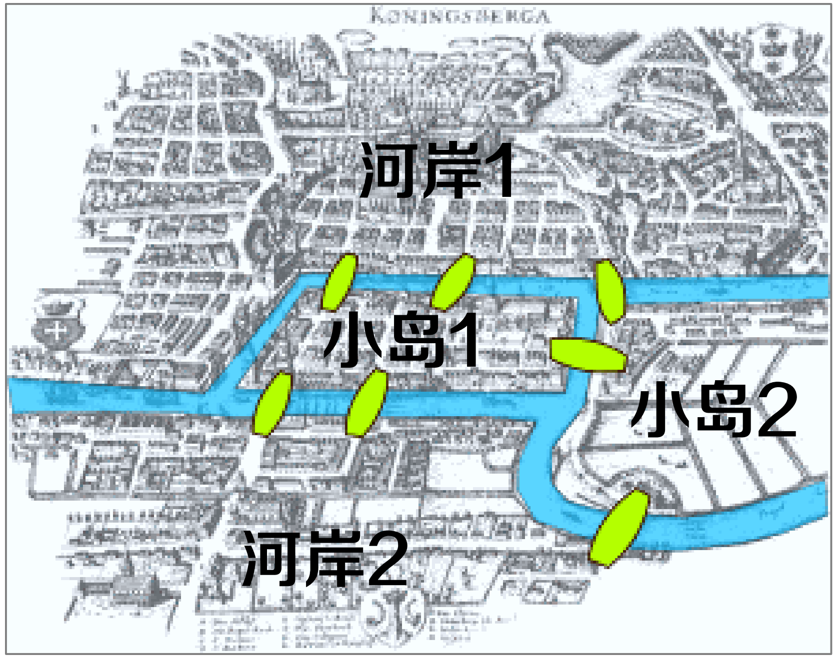
哪些图案可以仅用一笔就画出来?

图的背景

- 柯尼斯堡七桥问题：七座桥连接河岸和两个小岛，步行者怎样才能**不重复、不遗漏**地一次走完七座桥？

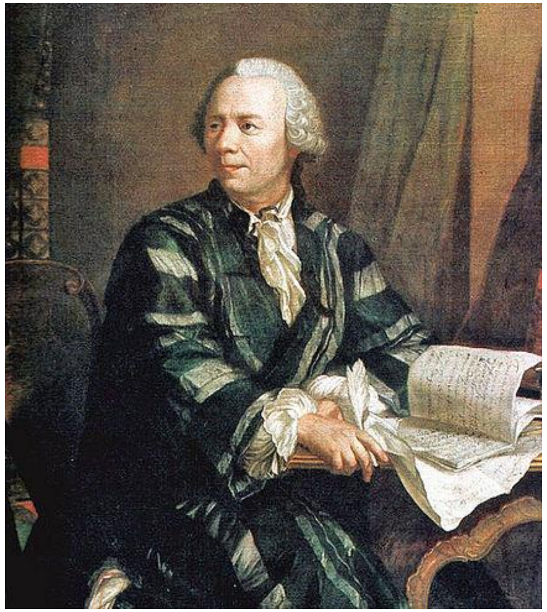


瑞士数学家
欧拉

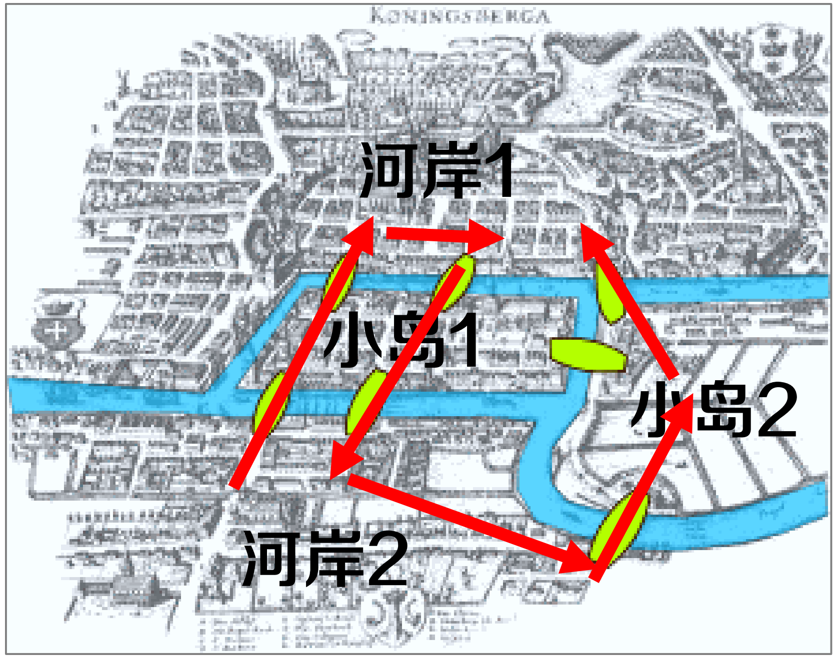


图的背景

- 柯尼斯堡七桥问题：七座桥连接河岸和两个小岛，步行者怎样才能**不重复、不遗漏**地一次走完七座桥？

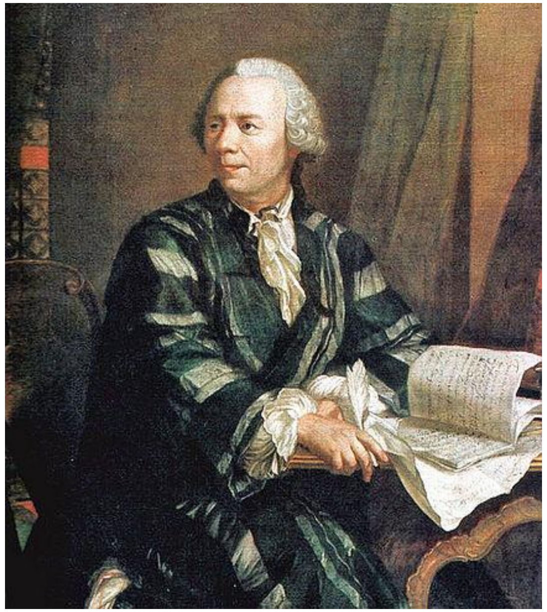


瑞士数学家
欧拉

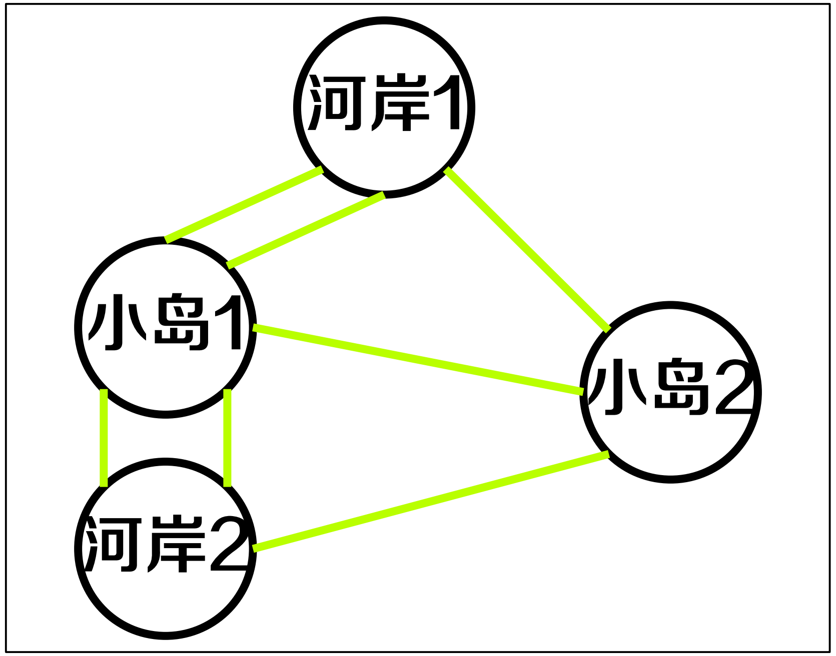


图的背景

- 柯尼斯堡七桥问题：七座桥连接河岸和两个小岛，步行者怎样才能**不重复、不遗漏**地一次走完七座桥？



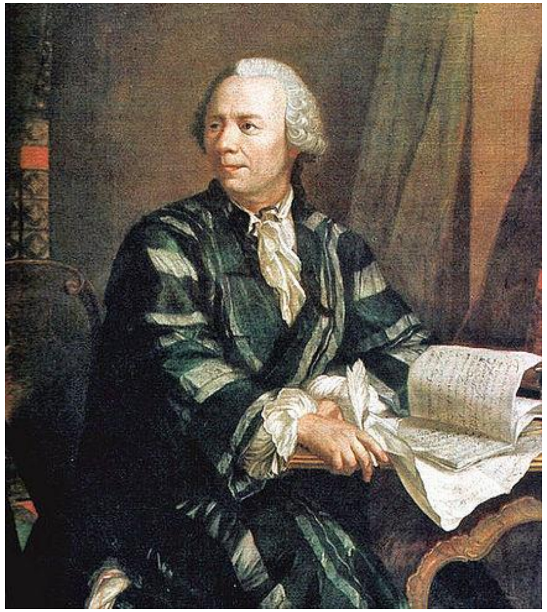
瑞士数学家
欧拉



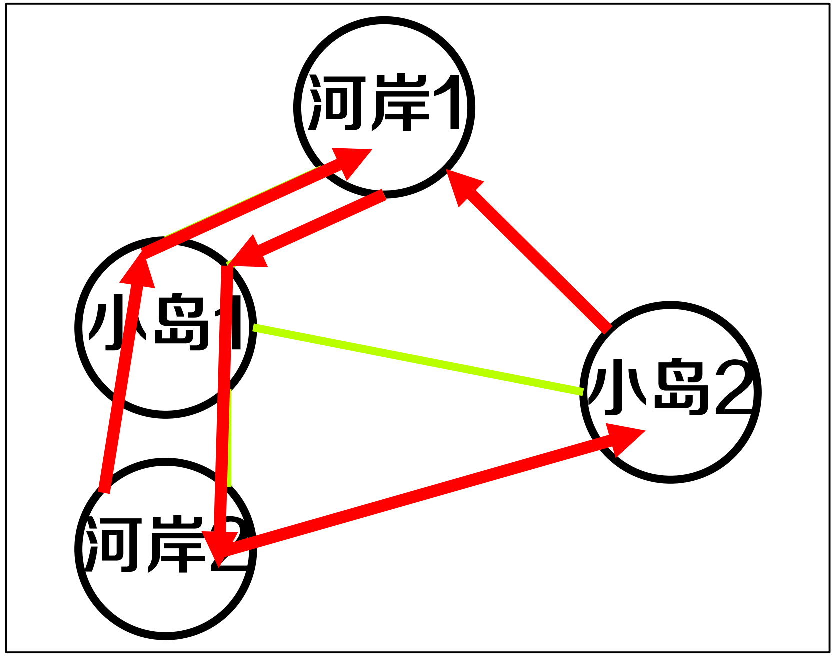
经过抽象之后，仅保留**点和边**的结构称为图

图的背景

- 柯尼斯堡七桥问题：七座桥连接河岸和两个小岛，步行者怎样才能**不重复、不遗漏**地一次走完七座桥？



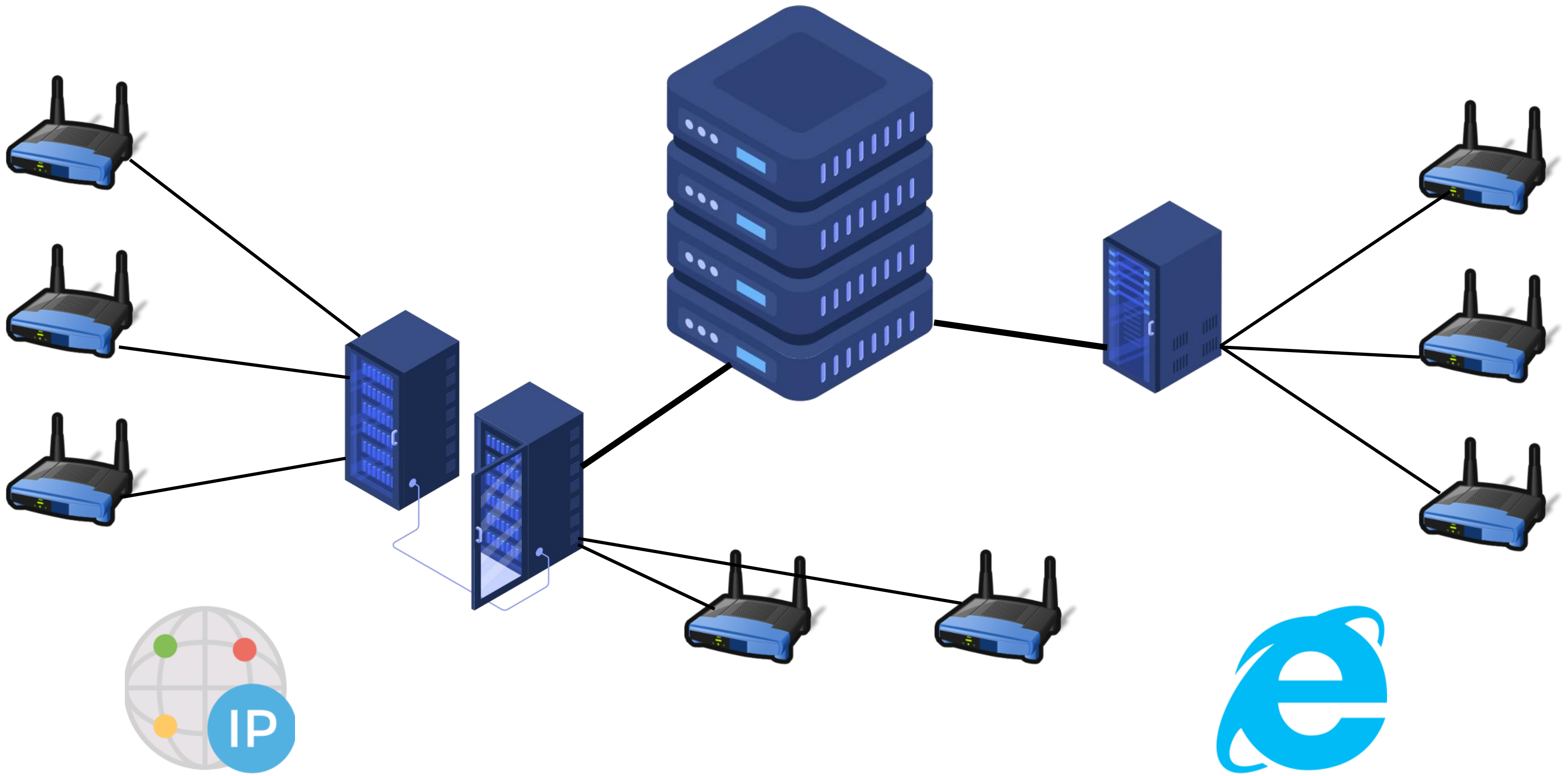
瑞士数学家
欧拉



忽略河岸和小岛的形状大小，重新建模为一笔画问题

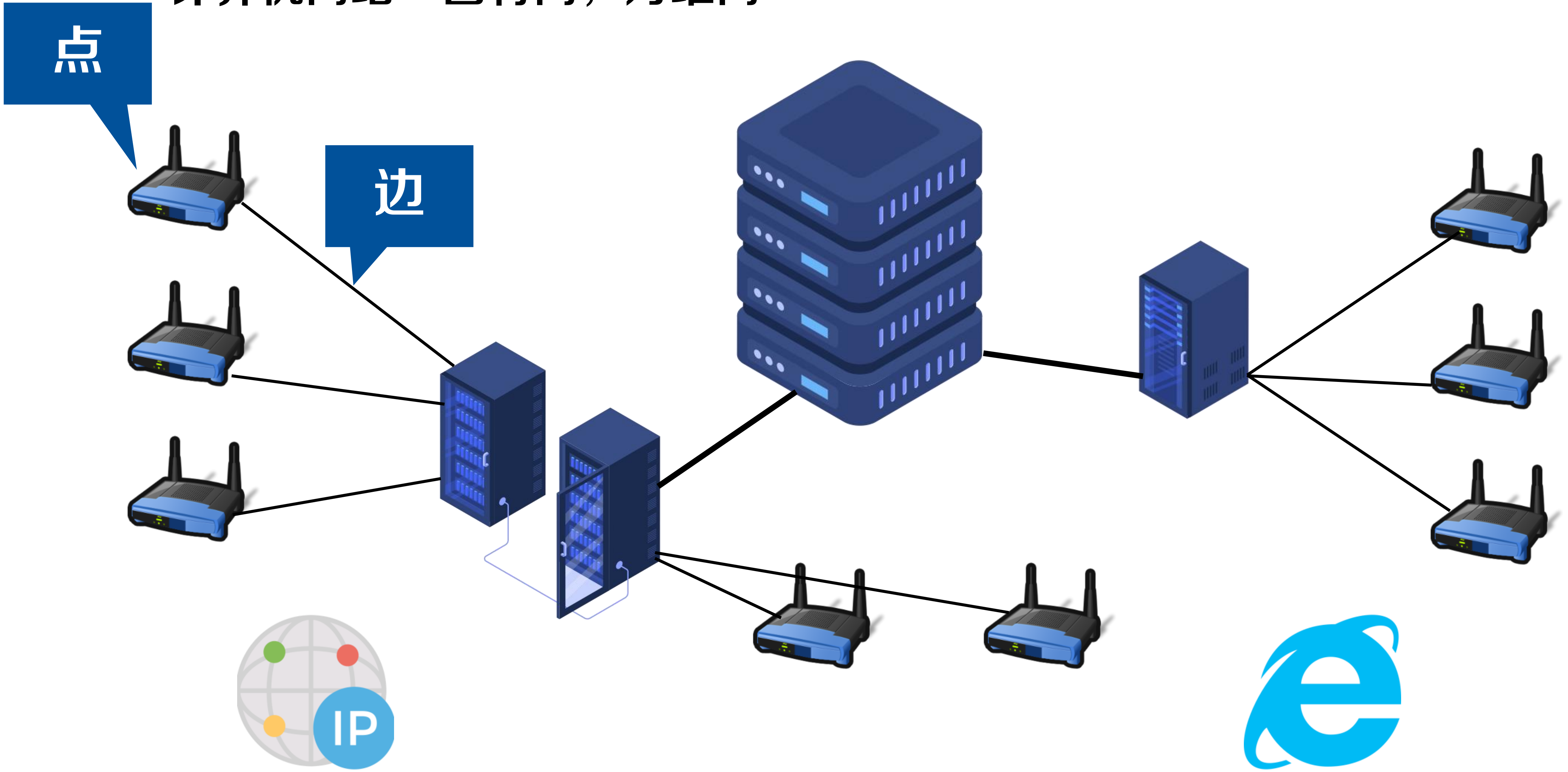
图的背景

- 图：现实中常见结构
  - 计算机网络：因特网，万维网



图的背景

- 图：现实中常见结构
  - 计算机网络：因特网，万维网



图的背景

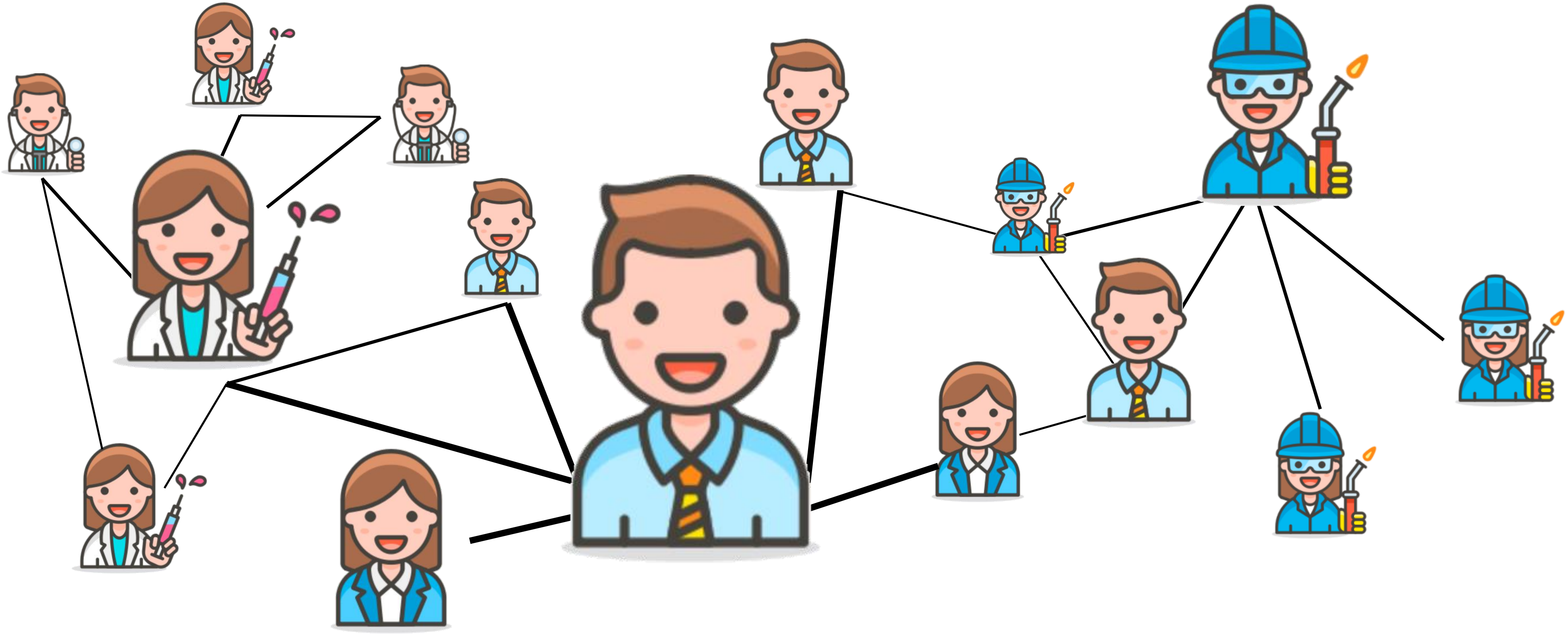
- 图：现实中常见结构
  - 交通出行：北京地铁图

图的背景

- 图：现实中常见结构
  - 交通出行：北京地铁图

图的背景

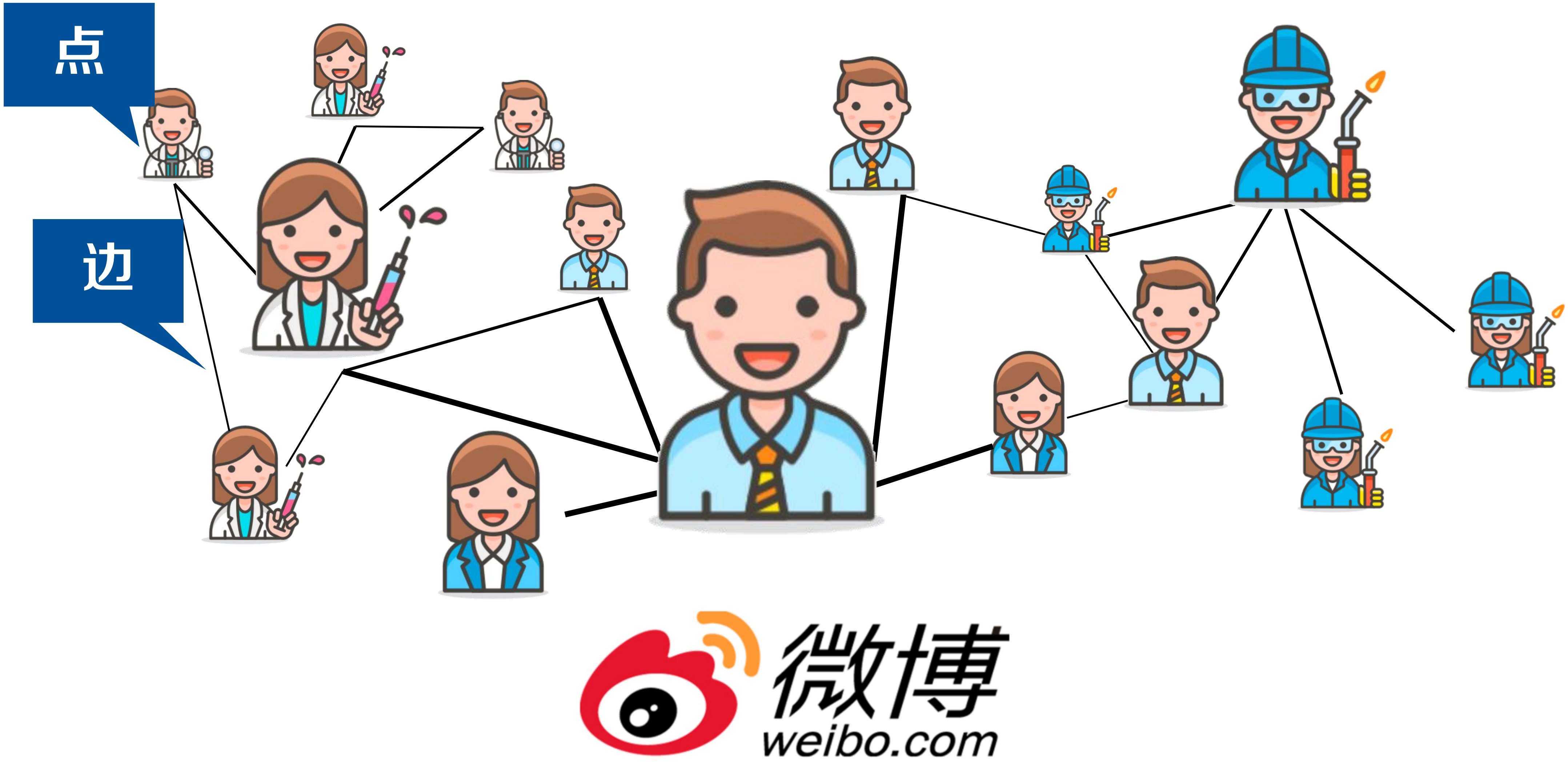
- 图：现实中常见结构
  - 社交网络：微博



微博 weibo.com

图的背景

- 图：现实中常见结构
  - 社交网络：微博



图的概念：图的定义

- 图可以表示为一个二元组$G=<V,E>$，其中
  - V表示非空顶点集，其元素称为顶点(Vertex)
  - E表示边集，其元素称为边(Edge)

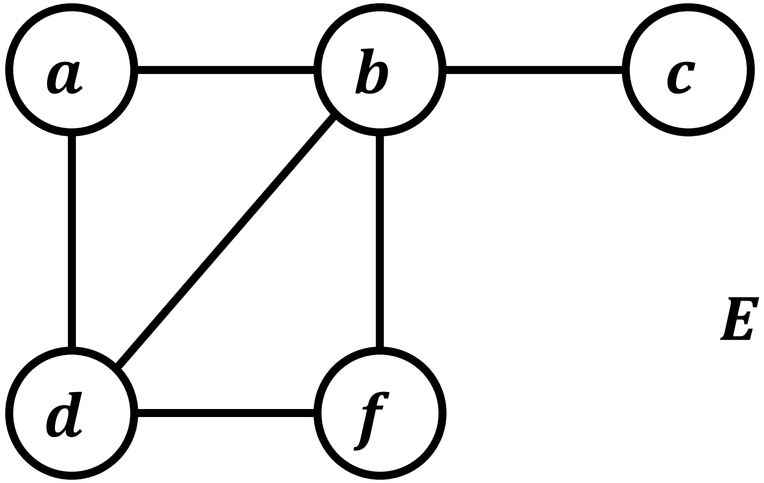
图的概念：图的定义

- 图可以表示为一个二元组$\boldsymbol{G}=<\boldsymbol{V},\boldsymbol{E}>$，其中
  - $\boldsymbol{V}$表示非空顶点集，其元素称为顶点(Vertex)
  - $\boldsymbol{E}$表示边集，其元素称为边(Edge)
- $\boldsymbol{e}=(\boldsymbol{u},\boldsymbol{v})$表示一条边，其中$\boldsymbol{u}\in\boldsymbol{V},\boldsymbol{v}\in\boldsymbol{V},\boldsymbol{e}\in\boldsymbol{E}$

图的概念：图的定义

- 图可以表示为一个二元组$G=<V,E>$，其中
  - V表示非空顶点集，其元素称为顶点(Vertex)
  - E表示边集，其元素称为边(Edge)
- $e=(u,v)$表示一条边，其中$u\in V, v\in V, e\in E$

- 无向图$G_1=<V_1,E_1>$

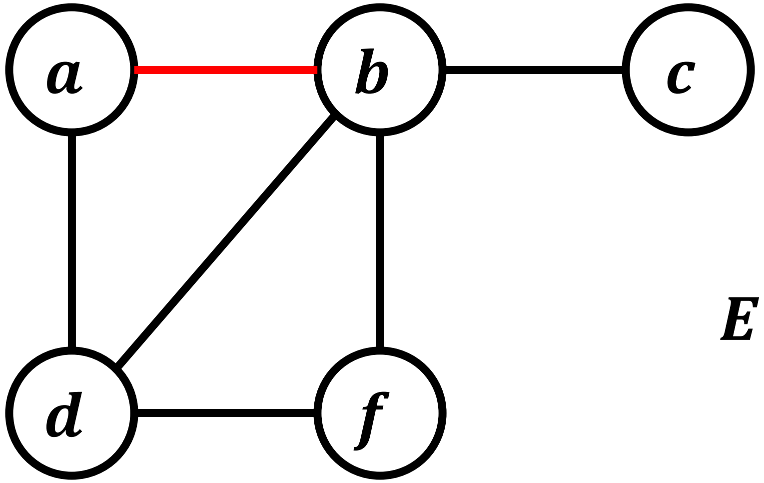


$$V_1=\{a,b,c,d,f\}$$

$$E_1=\{(a,b),(a,d),(b,c),(b,d),(b,f),(d,f)\}$$

图的概念：图的定义

- 图可以表示为一个二元组$G=<V,E>$，其中
  - V表示非空顶点集，其元素称为顶点(Vertex)
  - E表示边集，其元素称为边(Edge)
- $e=(u,v)$表示一条边，其中$u\in V, v\in V, e\in E$
- 无向图$G_1=<V_1,E_1>$



$$V_1=\{a,b,c,d,f\}$$

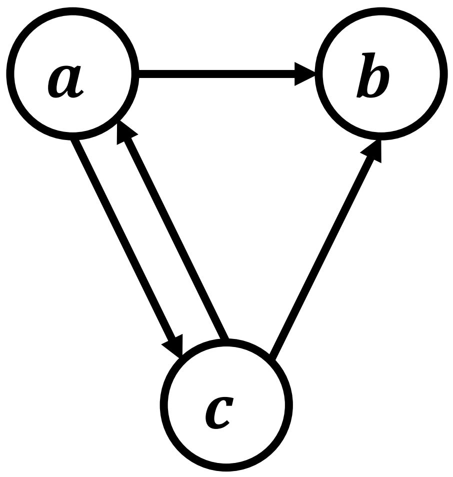
$$E_1=\{(a,b),(a,d),(b,c),(b,d),(b,f),(d,f)\}$$

(a,b)和(b,a)被视作同一条边

图的概念：图的定义

- 图可以表示为一个二元组$G =< V, E >$，其中
  - V表示非空顶点集，其元素称为顶点(Vertex)
  - E表示边集，其元素称为边(Edge)
- $e = (u, v)$表示一条边，其中$u \in V, v \in V, e \in E$

- 有向图$G_2 =< V_2, E_2 >$



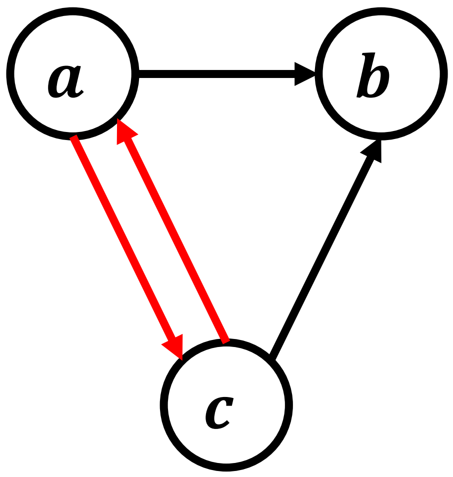
$$V_2 = \{a, b, c\}$$

$$E_2 = \{(a, b), (a, c), (c, a), (c, b)\}$$

图的概念：图的定义

- 图可以表示为一个二元组$G=<V,E>$，其中
  - V表示非空顶点集，其元素称为顶点(Vertex)
  - E表示边集，其元素称为边(Edge)
- $e=(u,v)$表示一条边，其中$u\in V, v\in V, e\in E$

- 有向图$G_2=<V_2,E_2>$



$$V_2=\{a,b,c\}$$

$$E_2=\{(a,b),(a,c),(c,a),(c,b)\}$$

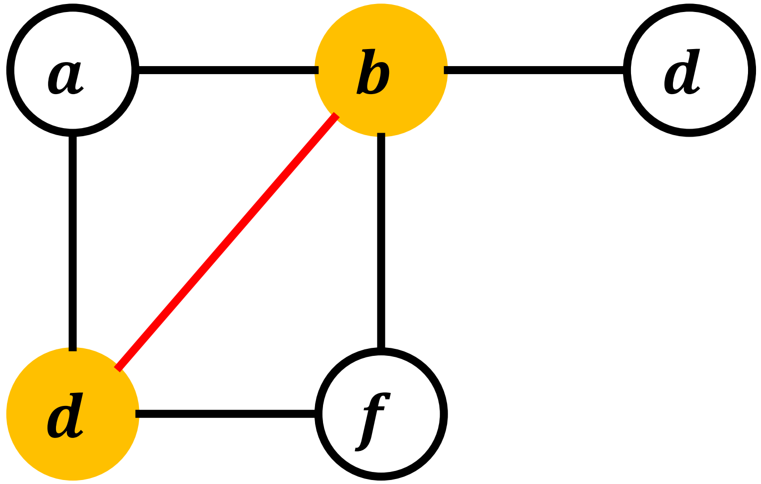
(a,c)和(c,a)是两条不同的边

图的概念：相邻与关联

- **相邻(Adjacent)**
  - 边(u, v)连接的顶点u和v相邻
- **关联(Incident)**
  - 边(u, v)和其连接的顶点u(或v)相互关联

图的概念：相邻与关联

- 相邻(Adjacent)
  - 边(u,v)连接的顶点u和v相邻
- 关联(Incident)
  - 边(u,v)和其连接的顶点u(或v)相互关联



顶点b和d相邻

(b,d)与顶点b和d关联

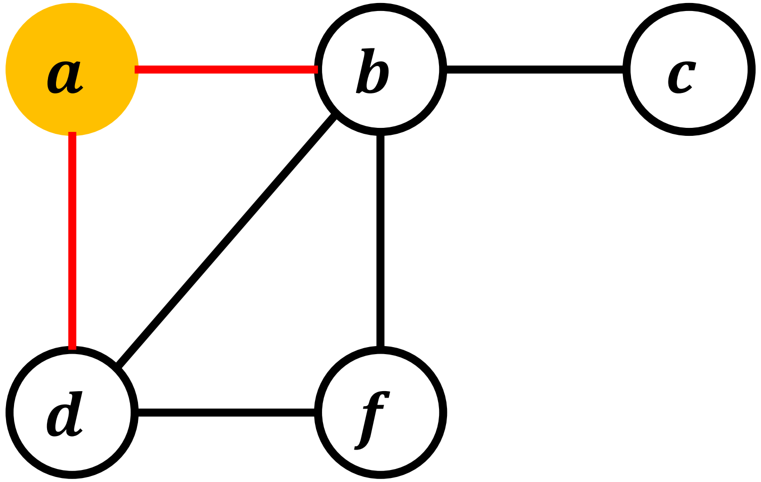
图的概念：度

- **顶点的度(Degree of a Vertex)**
  - 顶点v的度$deg(v)$是v关联的边数
- **图的度(Degree of a Graph)**
  - 图$G=<V,E>$的度，是图各顶点的度之和，$deg(G)=\sum_{v\in V}deg(v)$

图的概念：度

- 顶点的度(Degree of a Vertex)
  - 顶点v的度$deg(v)$是v关联的边数
- 图的度(Degree of a Graph)
  - 图$G=<V,E>$的度，是图各顶点的度之和，$deg(G)=\sum_{v\in V} deg(v)$

- 图$G_1=<V_1,E_1>$



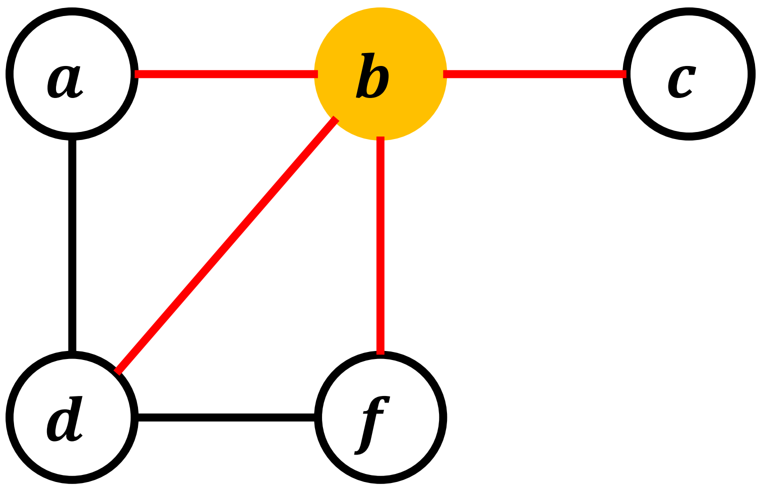
$V_1=\{a,b,c,d,f\}$

$deg(a)=2$

图的概念：度

- 顶点的度(Degree of a Vertex)
  - 顶点v的度$deg(v)$是v关联的边数
- 图的度(Degree of a Graph)
  - 图$G=<V,E>$的度，是图各顶点的度之和，$deg(G)=\sum_{v\in V} deg(v)$

- 图$G_1=<V_1,E_1>$



$V_1=\{a,b,c,d,f\}$

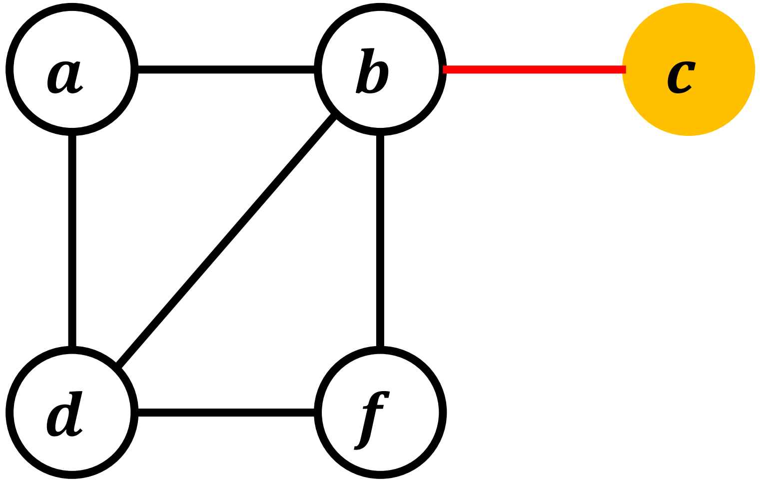
$deg(a)=2$

$deg(b)=4$

图的概念：度

- 顶点的度(Degree of a Vertex)
  - 顶点v的度$deg(v)$是v关联的边数
- 图的度(Degree of a Graph)
  - 图$G=<V,E>$的度，是图各顶点的度之和，$deg(G)=\sum_{v\in V} deg(v)$

- 图$G_1=<V_1,E_1>$



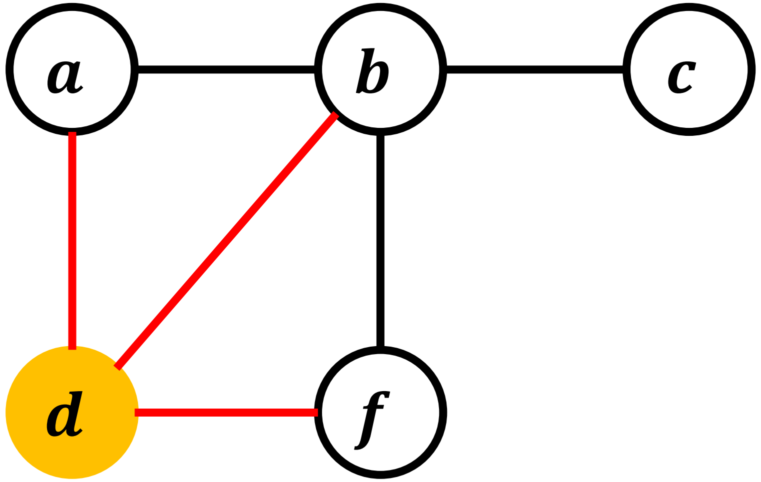
$V_1=\{a,b,c,d,f\}$

$deg(a)=2$
$deg(b)=4$
$deg(c)=1$

图的概念：度

- 顶点的度(Degree of a Vertex)
  - 顶点v的度$deg(v)$是v关联的边数
- 图的度(Degree of a Graph)
  - 图$G=<V,E>$的度，是图各顶点的度之和，$deg(G)=\sum_{v\in V} deg(v)$

- 图$G_1=<V_1,E_1>$



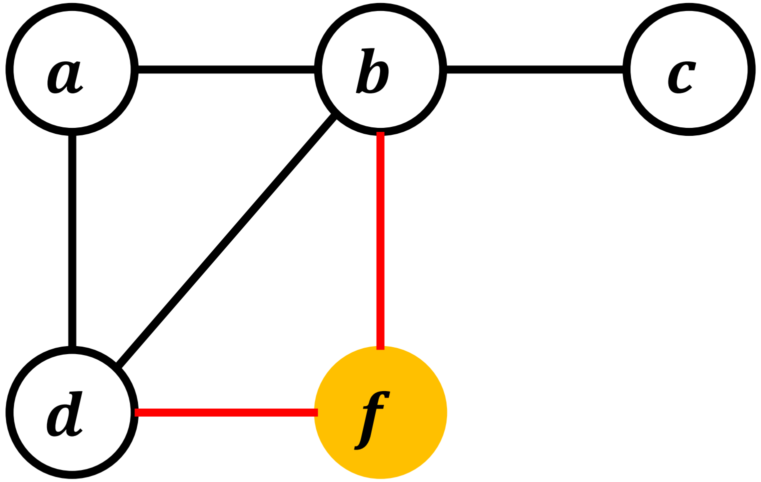
$V_1=\{a,b,c,d,f\}$

$deg(a)=2$
$deg(b)=4$
$deg(c)=1$
$deg(d)=3$

图的概念：度

- 顶点的度(Degree of a Vertex)
  - 顶点v的度$deg(v)$是v关联的边数
- 图的度(Degree of a Graph)
  - 图$G =< V, E >$的度，是图各顶点的度之和，$deg(G) = \sum_{v\in V} deg(v)$

- 图$G_1 =< V_1, E_1 >$



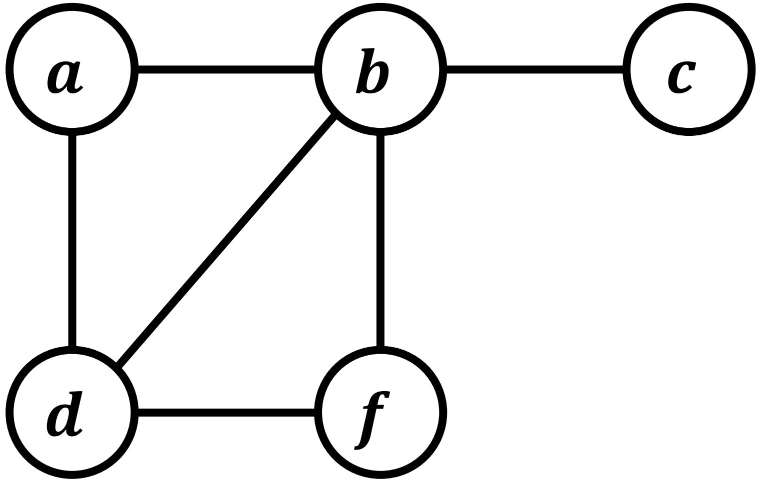
$V_1 = \{a, b, c, d, f\}$

$deg(a) = 2$
$deg(b) = 4$
$deg(c) = 1$
$deg(d) = 3$
$deg(f) = 2$

图的概念：度

- 顶点的度(Degree of a Vertex)
  - 顶点v的度$deg(v)$是v关联的边数
- 图的度(Degree of a Graph)
  - 图$G = < V, E >$的度，是图各顶点的度之和，$deg(G) = \sum_{v \in V} deg(v)$

- 图$G_1 = < V_1, E_1 >$



$$V_1 = \{a, b, c, d, f\}$$

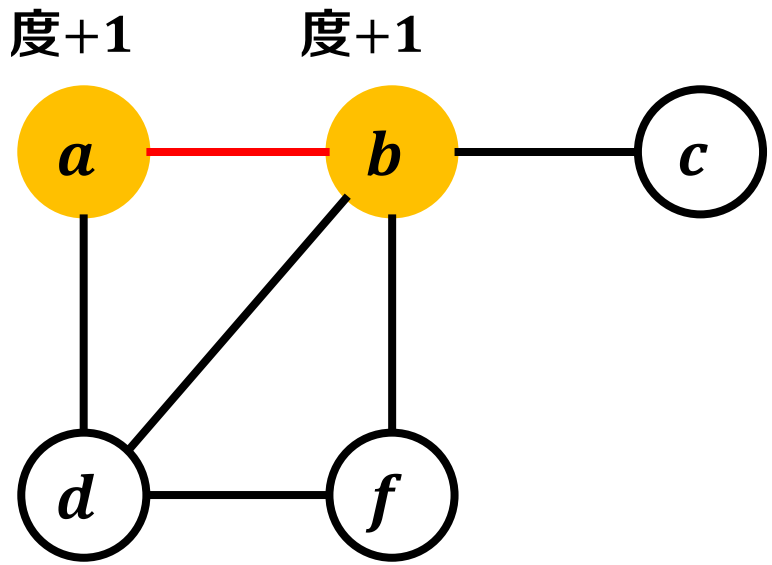
$$deg(a) = 2$$
$$deg(b) = 4$$
$$deg(c) = 1$$
$$deg(d) = 3$$
$$deg(f) = 2$$
$$deg(G_1) = 2 + 4 + 1 + 3 + 2 = 12$$

握手定理

- **握手定理(Handshaking Lemma)**
  - 无向图的度是边数的两倍， $deg(G) = 2|E|$

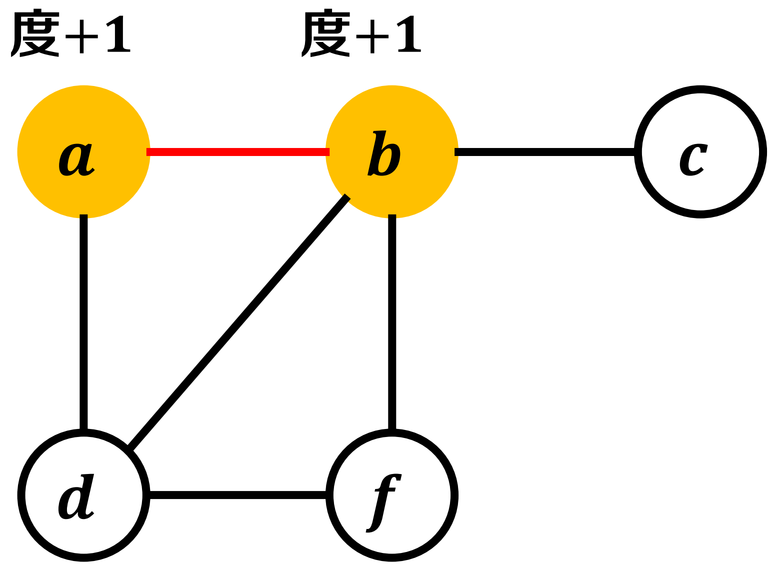
握手定理

- **握手定理(Handshaking Lemma)**
  - 无向图的度是边数的两倍， $deg(G) = 2|E|$
- **证明**
  - 边$e = (u, v)$在$deg(u)$和$deg(v)$中各被计算一次



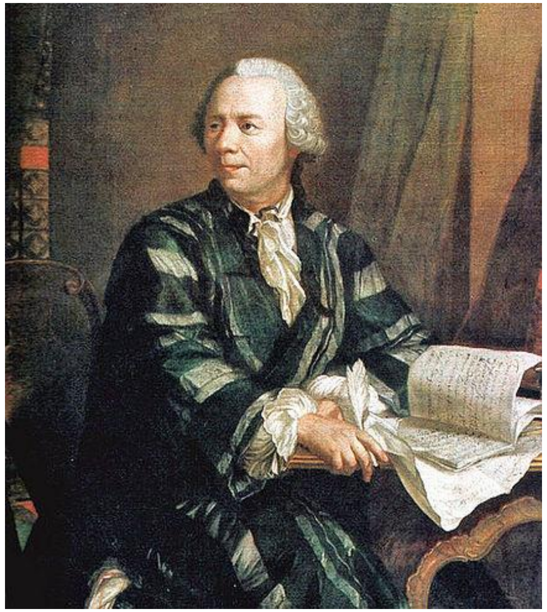
握手定理

- **握手定理(Handshaking Lemma)**
  - 无向图的度是边数的两倍， $deg(G) = 2|E|$
- **证明**
  - 边$e = (u, v)$在$deg(u)$和$deg(v)$中各被计算一次
  - 每条边为图的度贡献为2，$deg(G) = \sum_{e \in E} 2 = 2|E|$

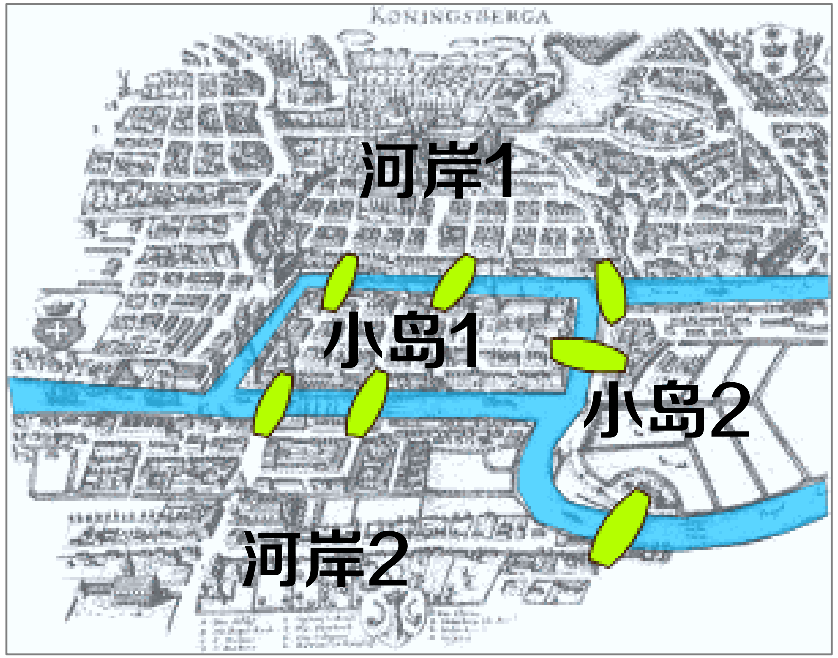


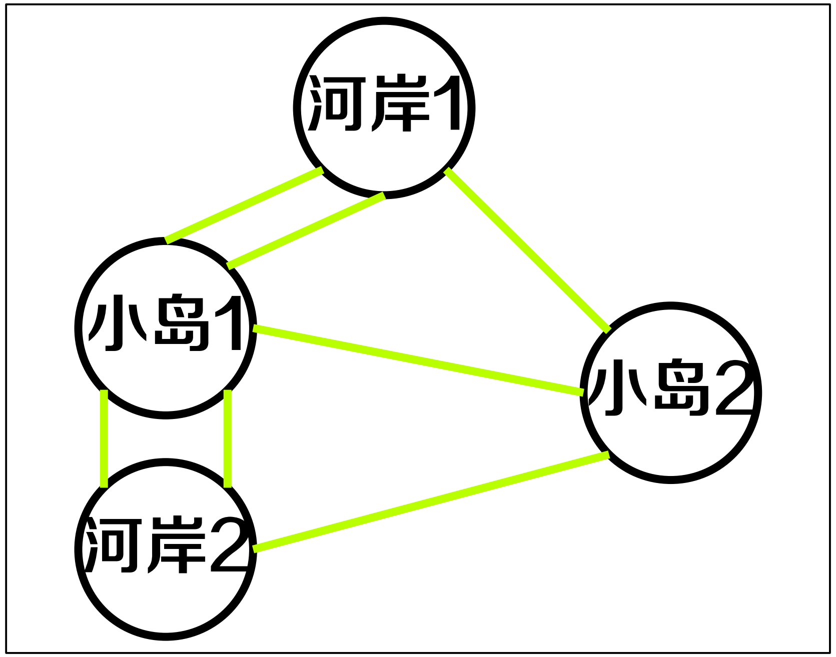
柯尼斯堡七桥问题

- 七座桥连接河岸和两个小岛，步行者怎样才能**不重复**、**不遗漏**地一次走完七座桥？



瑞士数学家
欧拉

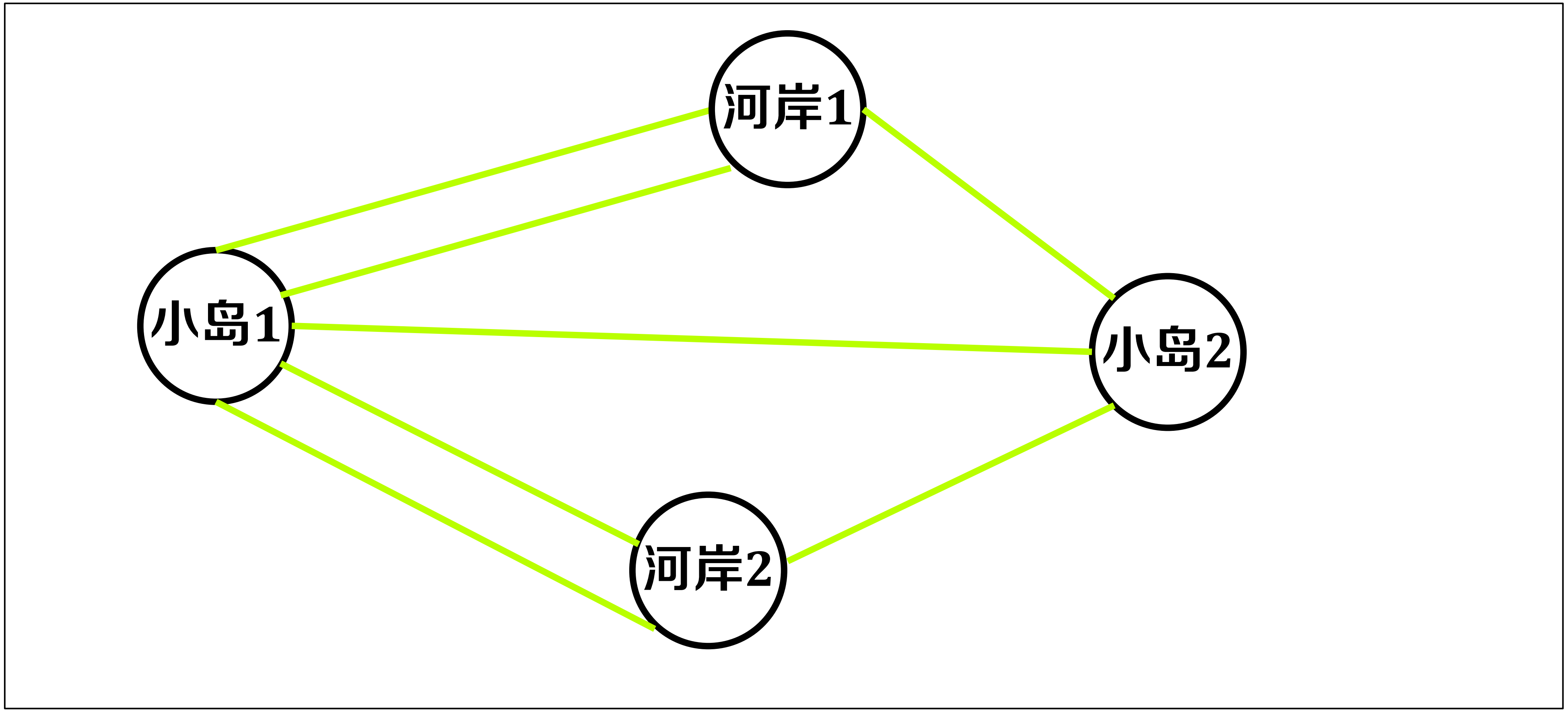




仅保留点和边的结构，建模为一笔画问题

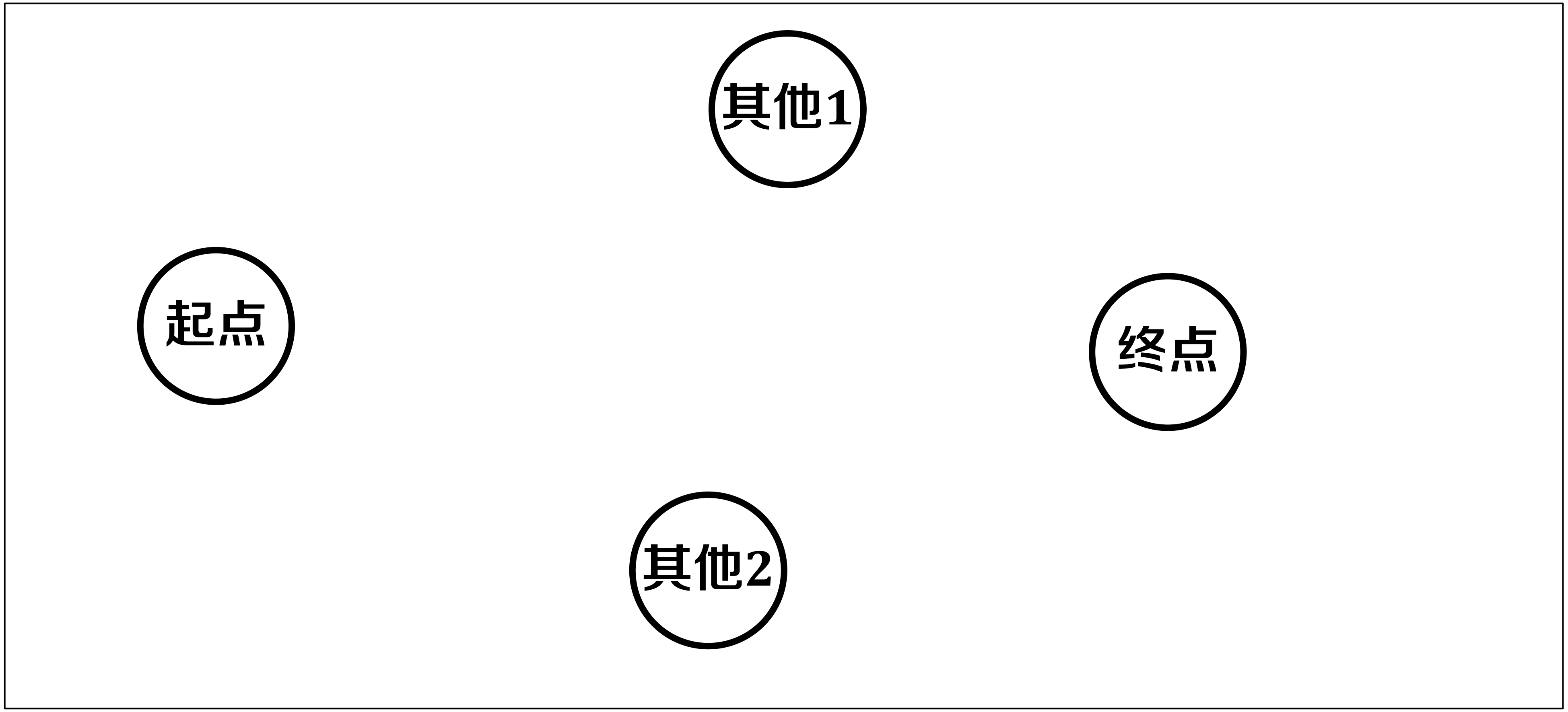
柯尼斯堡七桥问题

- 七座桥连接河岸和两个小岛，步行者怎样才能**不重复**、**不遗漏**地一次走完七座桥？



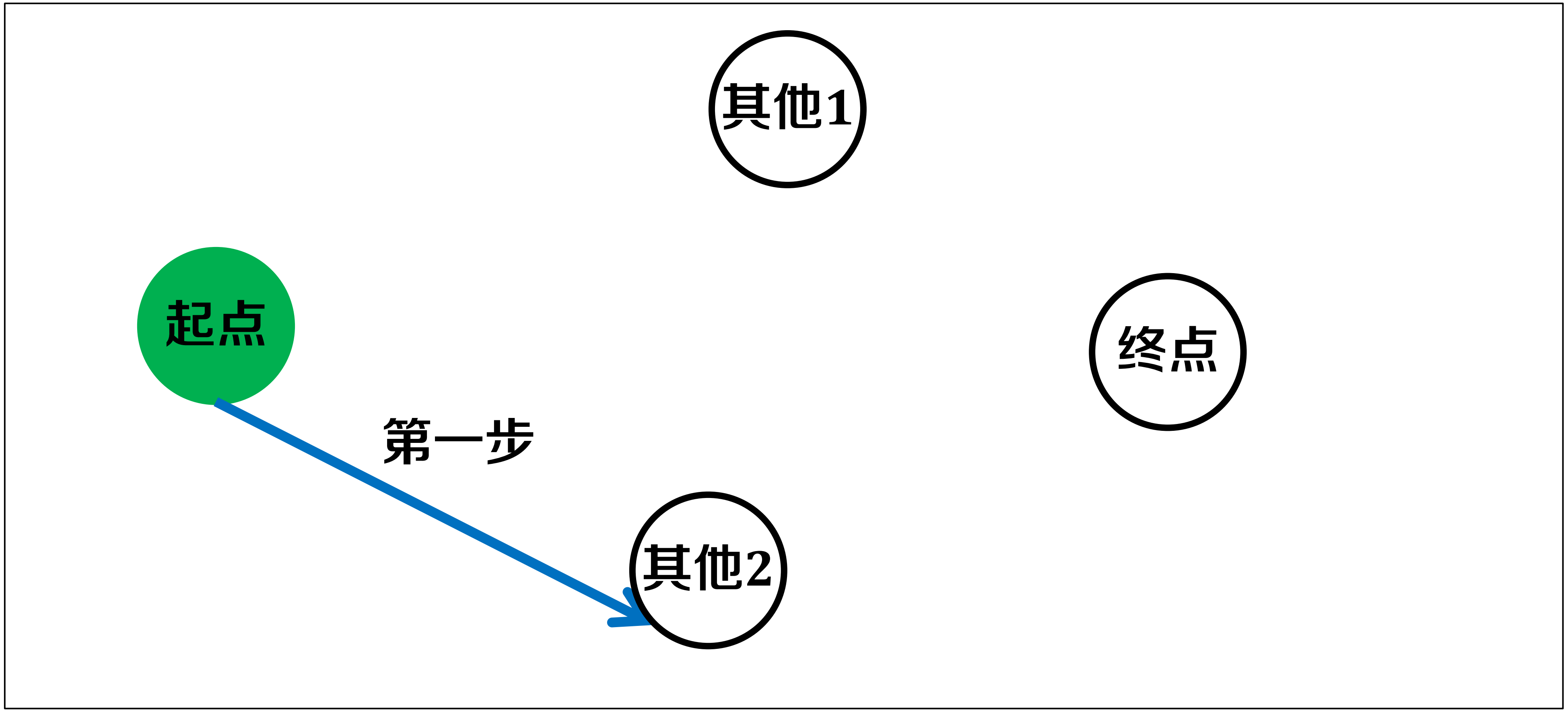
柯尼斯堡七桥问题

- 从起点出发，经过图中所有边，最终到达终点



柯尼斯堡七桥问题

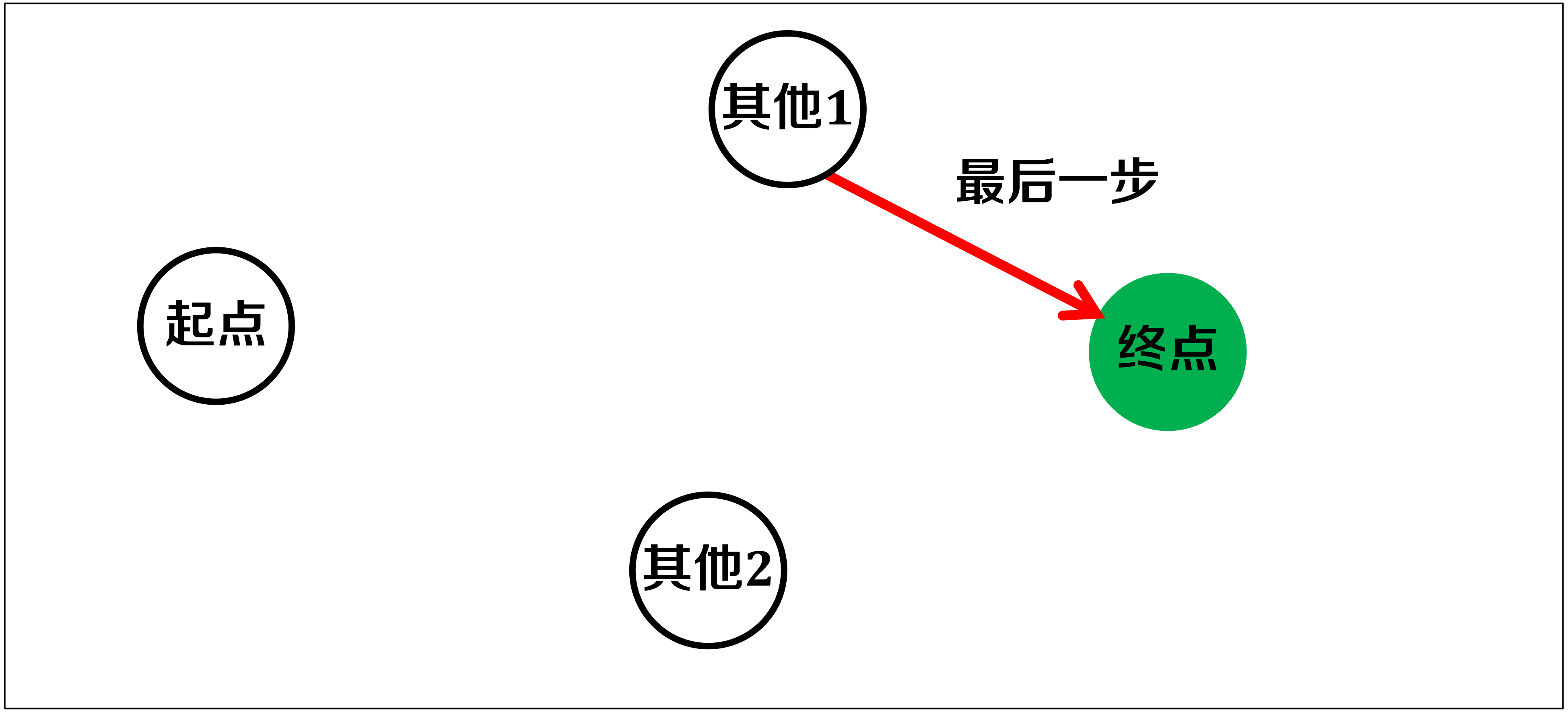
- 从起点出发，经过图中所有边，最终到达终点



起点：第一步需要从一条边离开

柯尼斯堡七桥问题

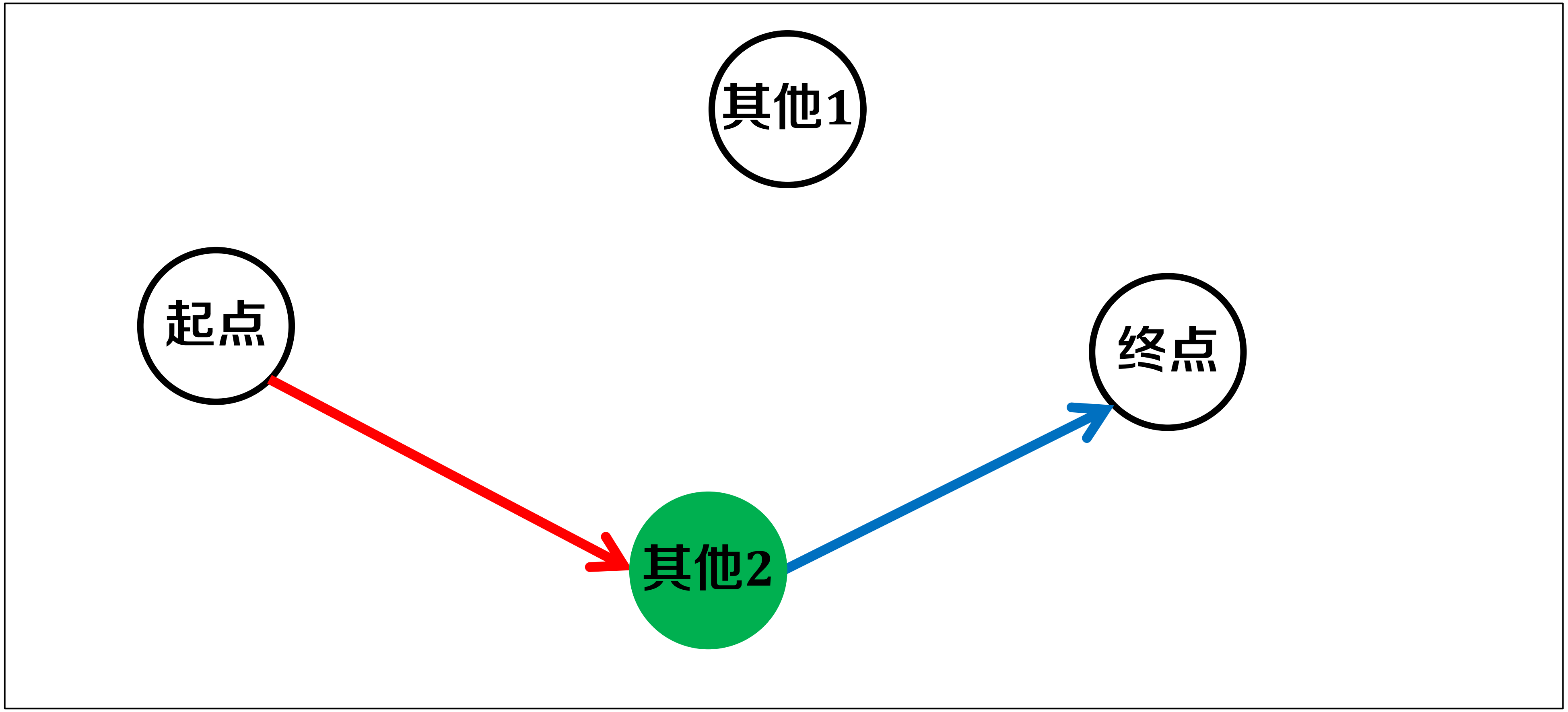
- 从起点出发，经过图中所有边，最终到达终点



终点：最后一步需要从一条边到达

柯尼斯堡七桥问题

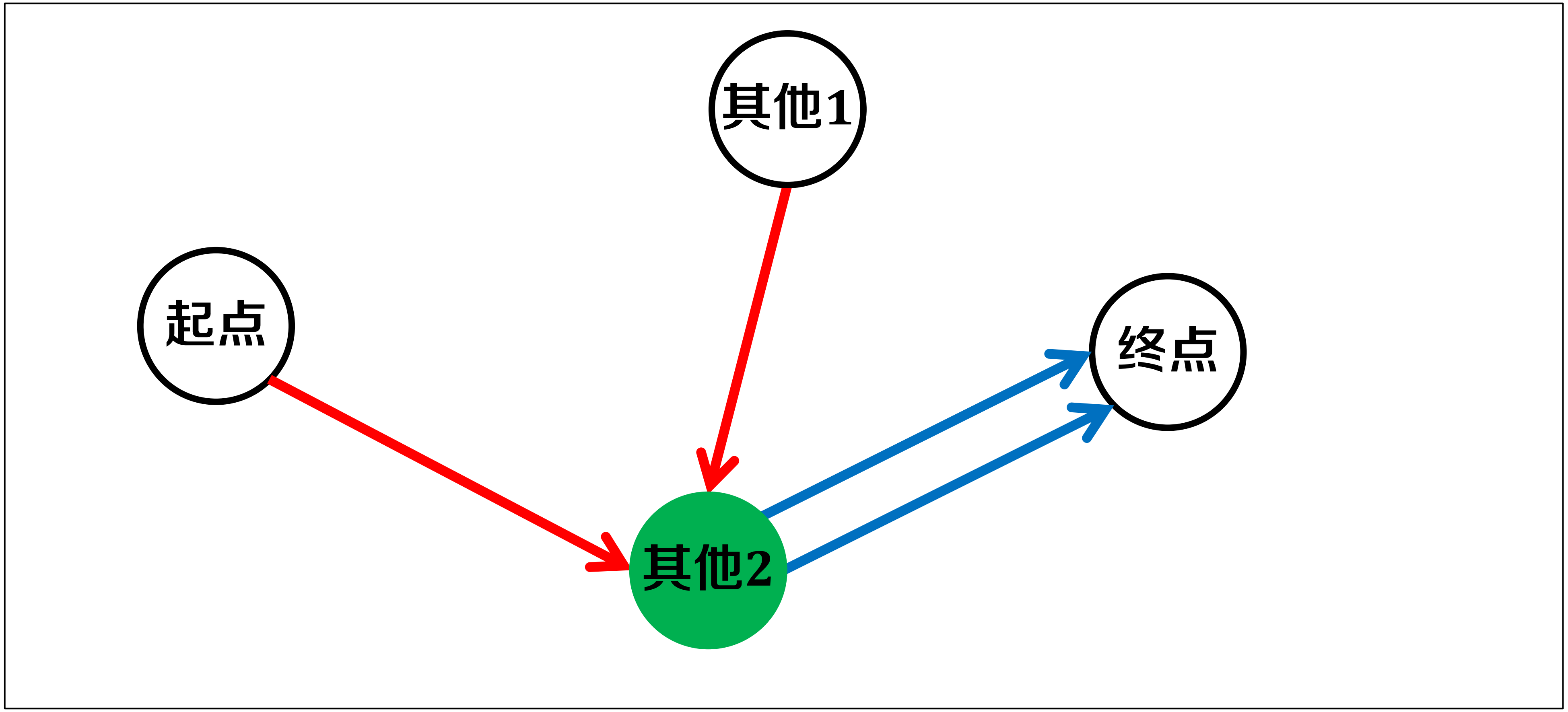
- 从起点出发，经过图中所有边，最终到达终点



其他顶点：从一条边到达后，需要从另一条边离开

柯尼斯堡七桥问题

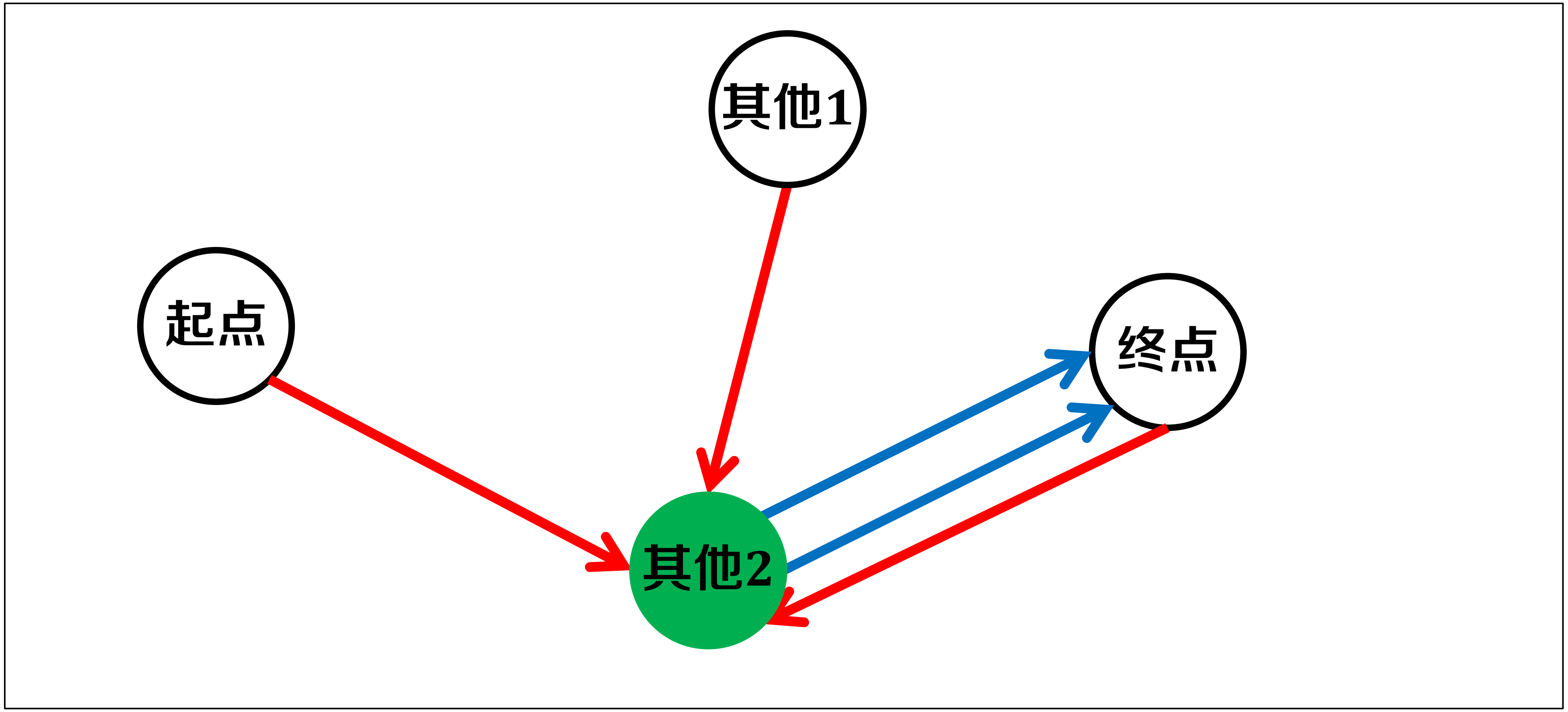
- 从起点出发，经过图中所有边，最终到达终点



其他顶点：从一条边到达后，需要从另一条边离开

柯尼斯堡七桥问题

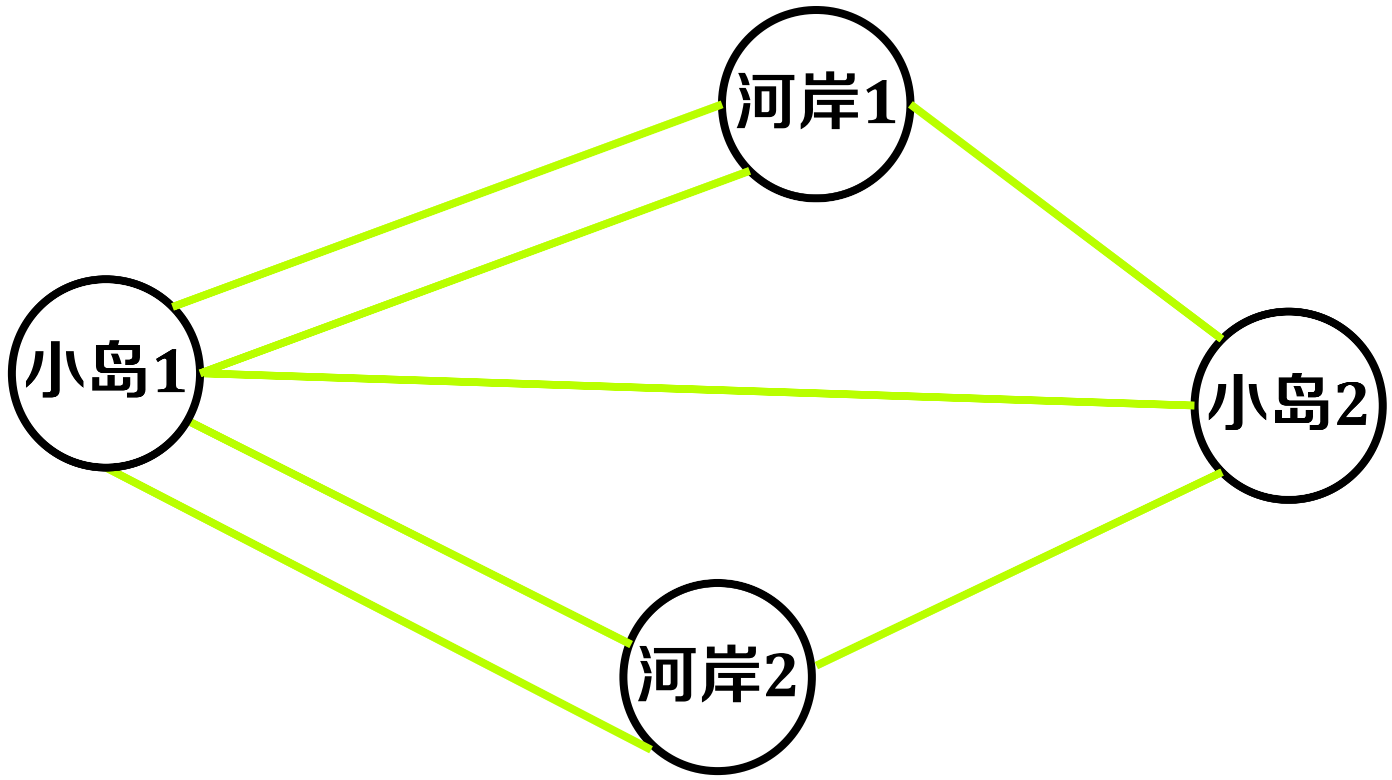
- 从起点出发，经过图中所有边，最终到达终点

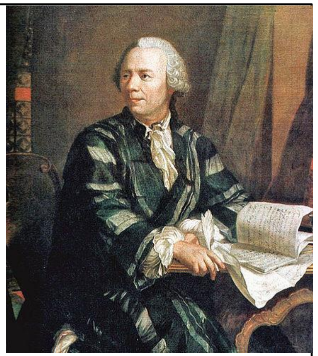


其他顶点的度为必须为**偶数**，否则无法离开

柯尼斯堡七桥问题

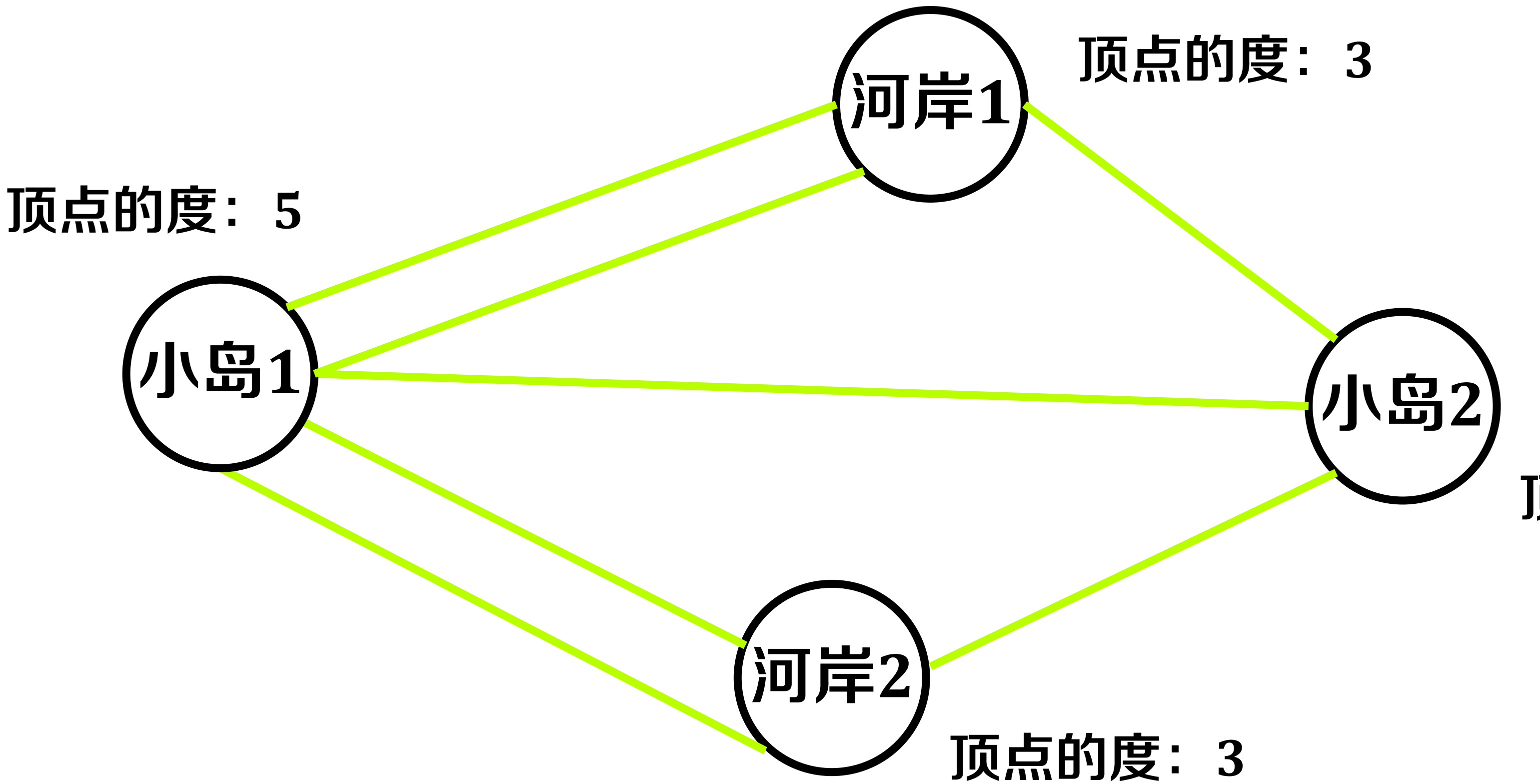
- 从起点出发，经过图中所有边，最终到达终点

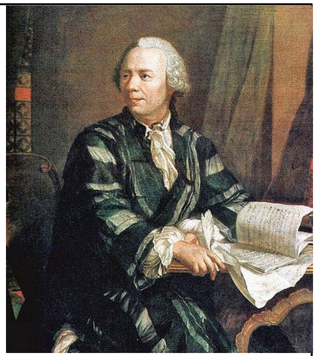




柯尼斯堡七桥问题

- 从起点出发，经过图中所有边，最终到达终点

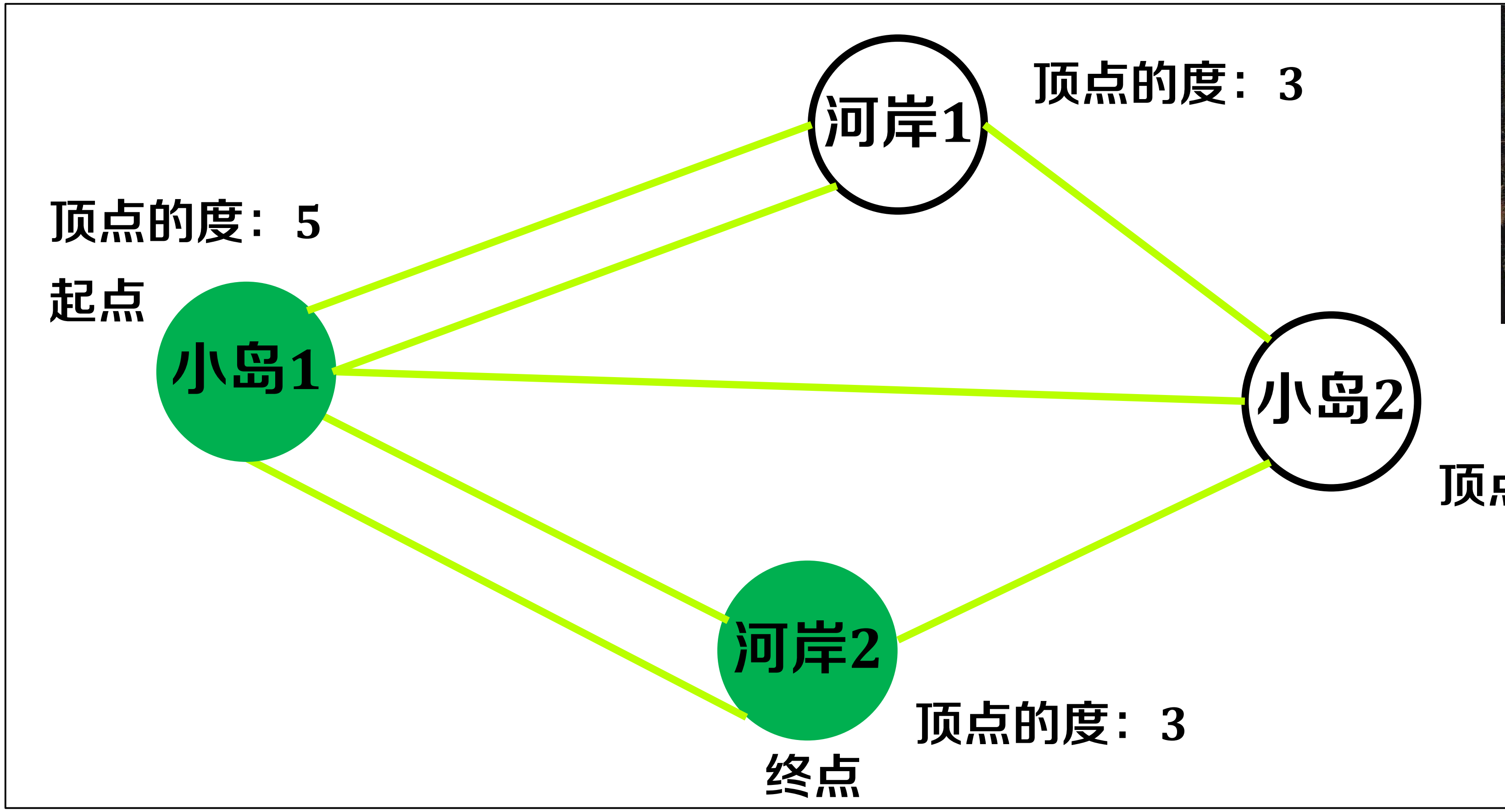


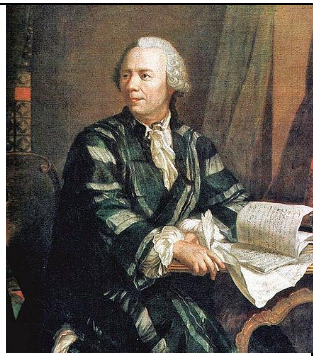


计算出所有顶点的度

柯尼斯堡七桥问题

- 从起点出发，经过图中所有边，最终到达终点

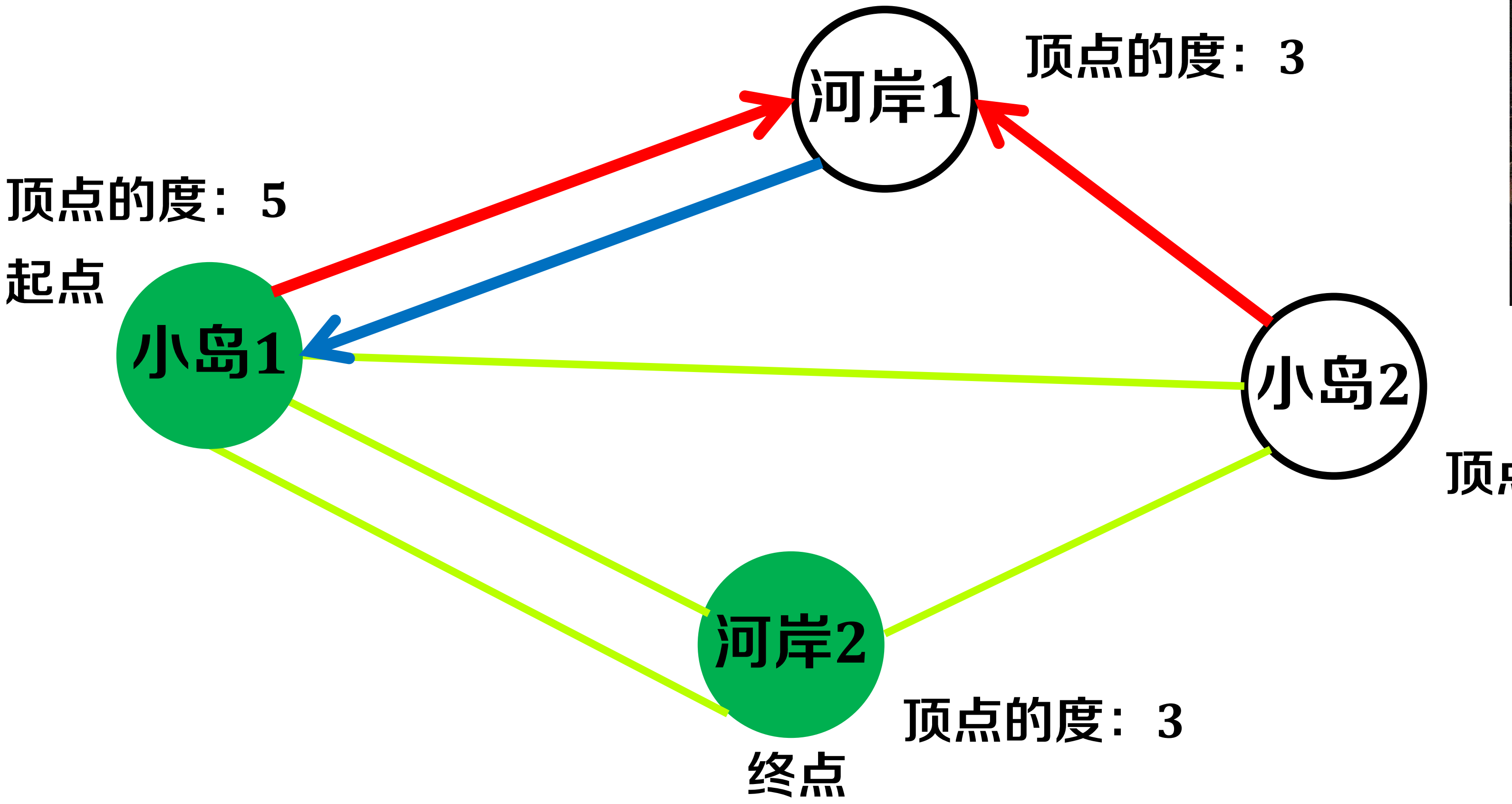


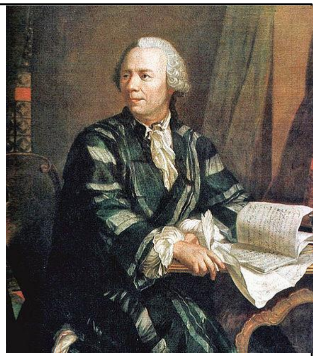


选择任意起点和终点

柯尼斯堡七桥问题

- 从起点出发，经过图中所有边，最终到达终点

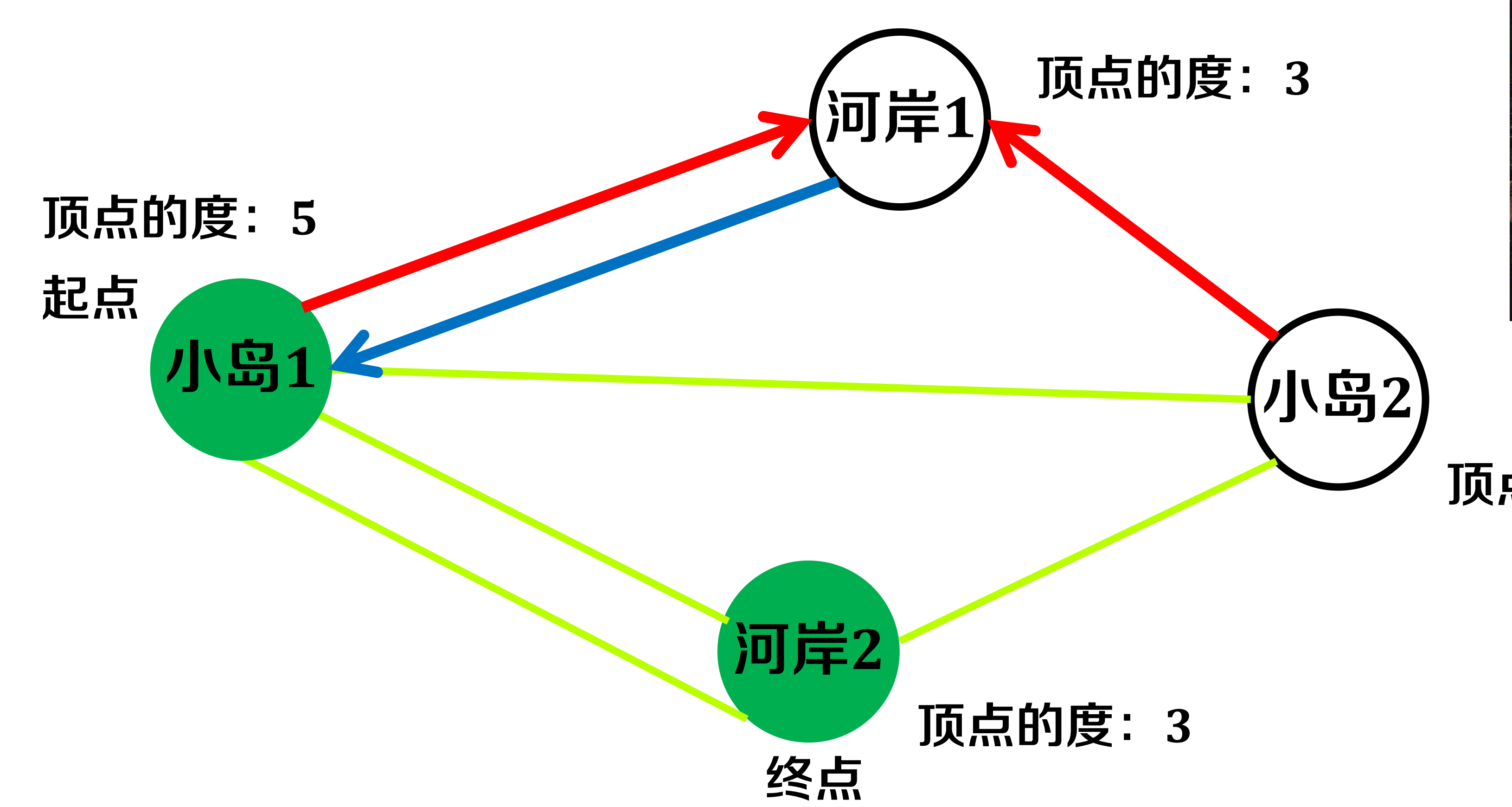


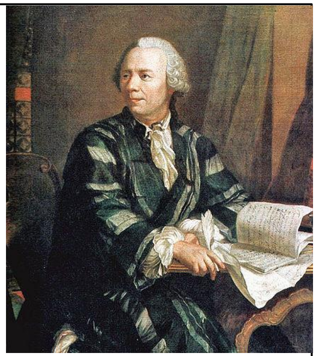


选择任意起点和终点，都存在无法离开的其他顶点

柯尼斯堡七桥问题

- 从起点出发，经过图中所有边，最终到达终点





柯尼斯堡七桥问题无解

图的概念：路径

- **路径(Path)**
  - 图中一个的顶点序列$< v_0, v_1, \dots, v_k >$称为v_0到v_k的**路径**

图的概念：路径

- **路径(Path)**
  - 图中一个的顶点序列$< v_0, v_1, \ldots, v_k >$称为$v_0$到$v_k$的路径
  - 路径包含顶点$v_0, v_1, \ldots, v_k$和边$(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)$

图的概念：路径

- 路径(Path)
  - 图中一个的顶点序列$< v_0, v_1, \ldots, v_k >$称为$v_0$到$v_k$的路径
  - 路径包含顶点$v_0, v_1, \ldots, v_k$和边$(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)$
  - 存在路径$< v_0, v_1, \ldots, v_k >$，则$v_0$**可达**$v_k$

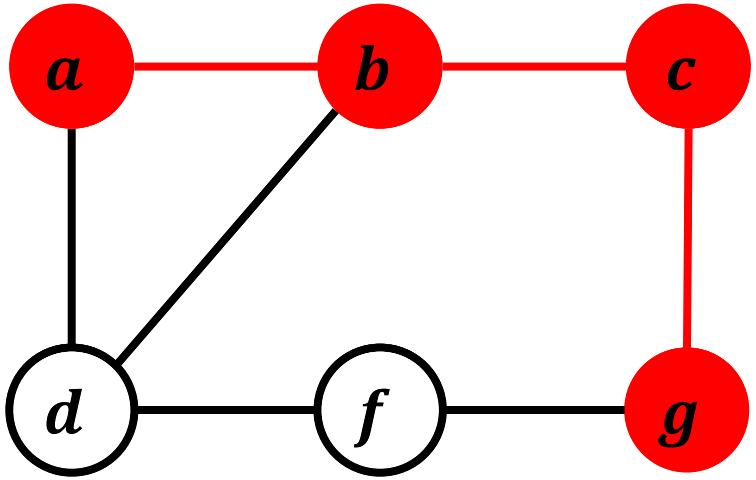
图的概念：路径

- 路径(Path)
  - 图中一个的顶点序列$< v_0, v_1, \dots, v_k >$称为v_0到v_k的路径
  - 路径包含顶点$v_0, v_1, \dots, v_k$和边$(v_0, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k)$
  - 存在路径$< v_0, v_1, \dots, v_k >$，则v_0可达v_k
  - 如果$v_0, v_1, \dots, v_k$互不相同，则该路径是**简单的**

图的概念：路径

- 路径(Path)
  - 图中一个的顶点序列$< v_0, v_1, \dots, v_k >$称为v_0到v_k的路径
  - 路径包含顶点$v_0, v_1, \dots, v_k$和边$(v_0, v_1), (v_1, v_2), \dots, (v_{k-1}, v_k)$
  - 存在路径$< v_0, v_1, \dots, v_k >$，则v_0可达v_k
  - 如果$v_0, v_1, \dots, v_k$互不相同，则该路径是简单的

- 路径$P =< a, b, c, g >$，顶点a可达顶点g



图的概念：环路

- **环路(Cycle)**
  - 如果路径$<v_0, v_1, \dots, v_k>$中$v_0 = v_k$且至少包含一条边，则该路径构成**环路**

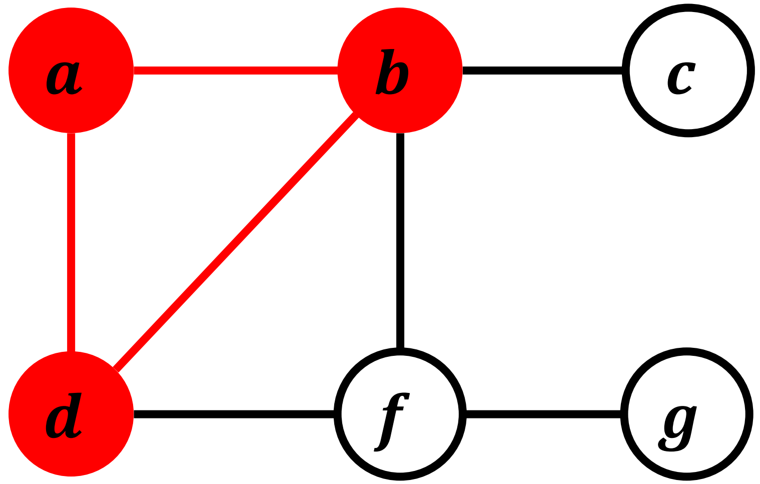
图的概念：环路

- **环路(Cycle)**
  - 如果路径$< v_0, v_1, \dots, v_k >$中$v_0 = v_k$且至少包含一条边，则该路径构成环路
  - 如果$v_1, v_2, \dots, v_k$互不相同，则该环路是**简单的**

图的概念：环路

- **环路(Cycle)**
  - 如果路径$< v_0, v_1, \dots, v_k >$中$v_0 = v_k$且至少包含一条边，则该路径构成**环路**
  - 如果$v_1, v_2, \dots, v_k$互不相同，则该环路是简单的

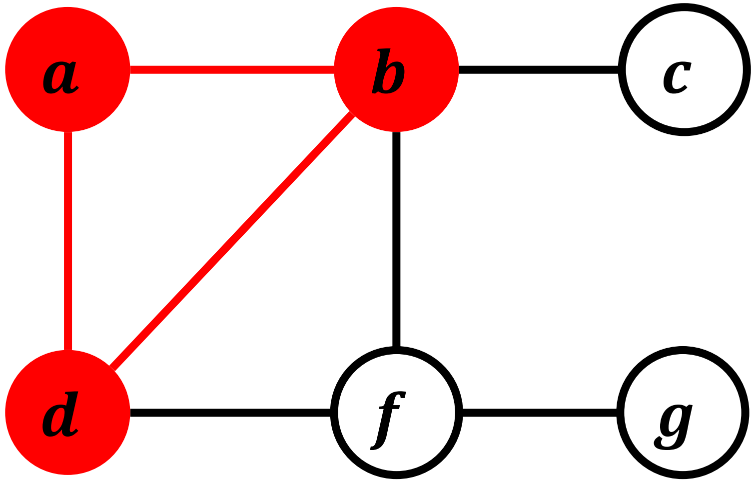
- **环路$C =< a, b, d, a >$**



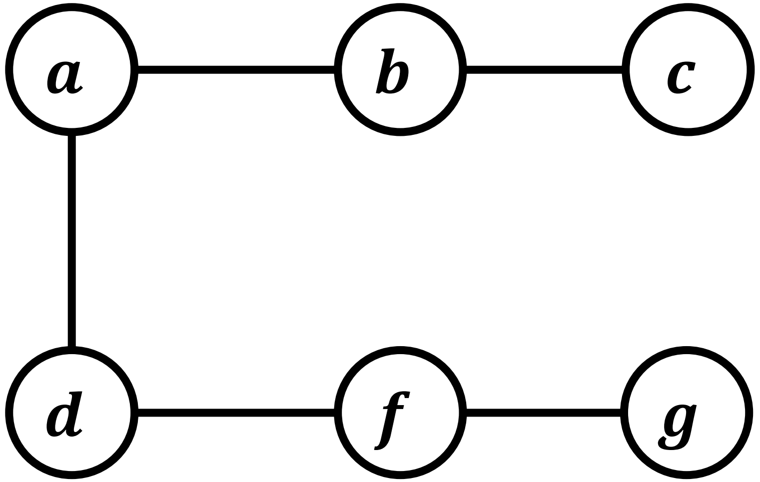
图的概念：环路

- 环路(Cycle)
  - 如果路径$< v_0, v_1, \ldots, v_k >$中$v_0 = v_k$且至少包含一条边，则该路径构成**环路**
  - 如果$v_1, v_2, \ldots, v_k$互不相同，则该环路是简单的

- 无环图(Acyclic Graph)：图中不存在环路



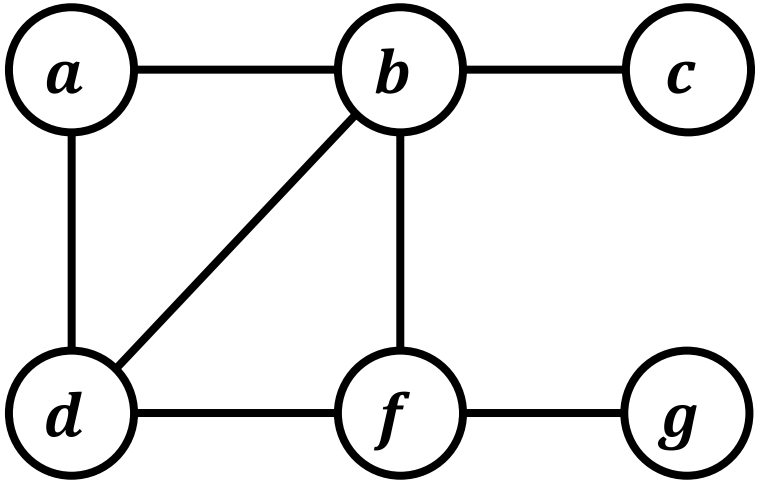
有环图



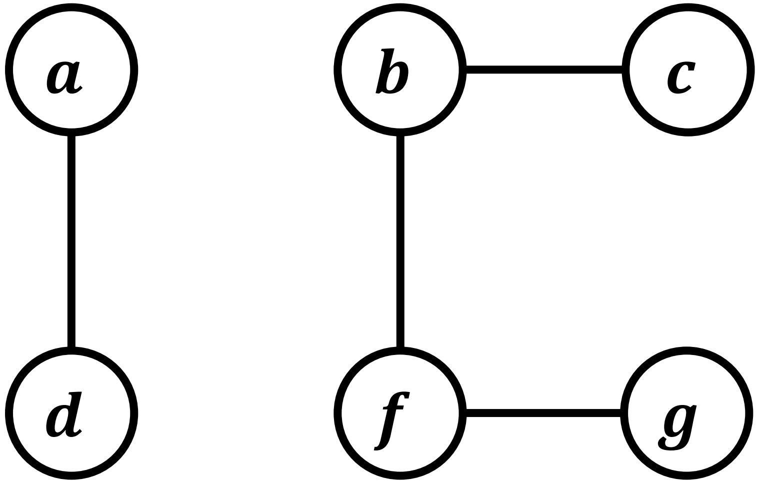
无环图

图的概念：连通

- 连通(Connectivity)
  - 如果图的任意顶点互相可达，则称该图是**连通的**，反之称为非连通



连通图

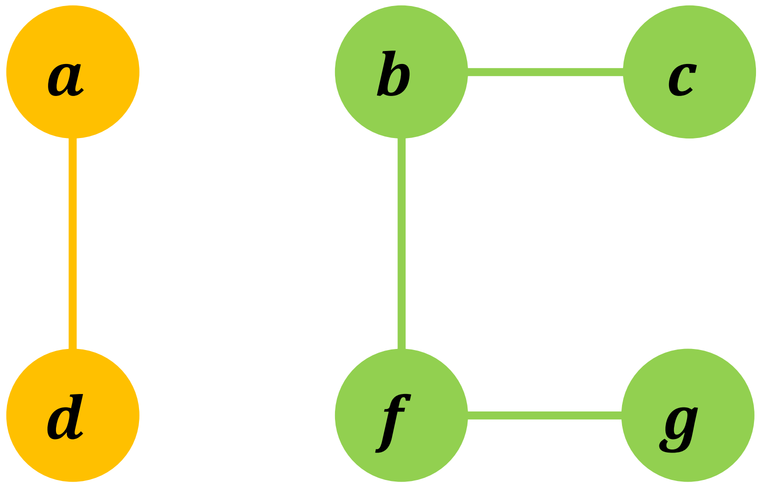


非连通图

图的概念：连通

- 连通(Connectivity)
  - 如果图的任意对顶点互相可达，则称该图是连通的，反之称为非连通

- 连通分量(Connected Components)
  - 根据是否连通将顶点进行分组，相互可达的顶点集称为**连通分量**

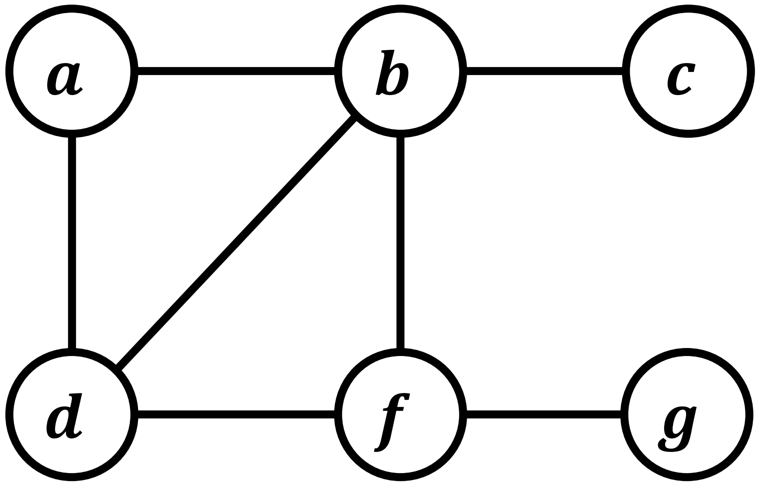


连通分量1　　连通分量2

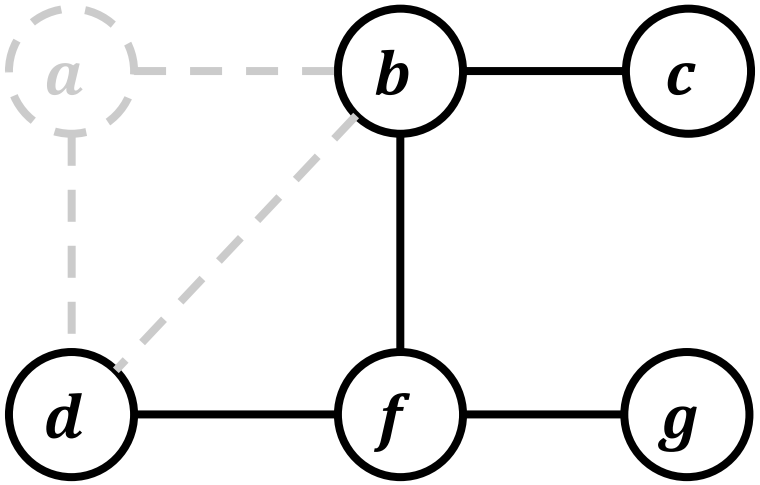
非连通图

图的概念：子图

- 子图(Subgraph)
  - 如果$V' \subseteq V, E' \subseteq E$，则称图$G' = < V', E' >$是图$G$的一个子图



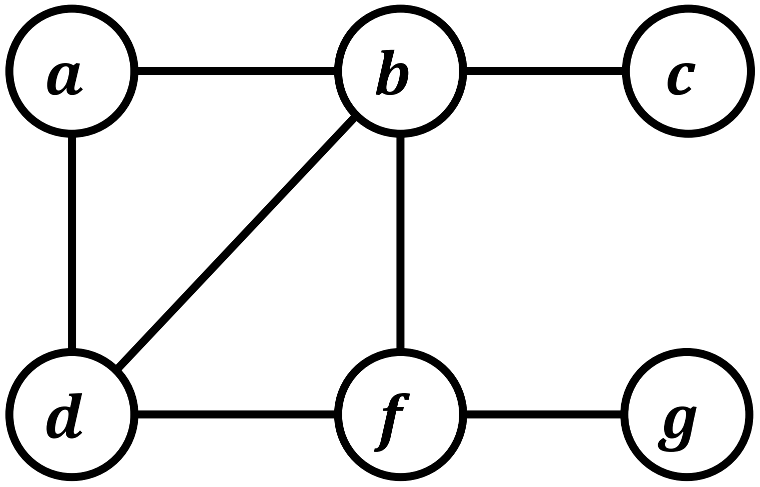
图G



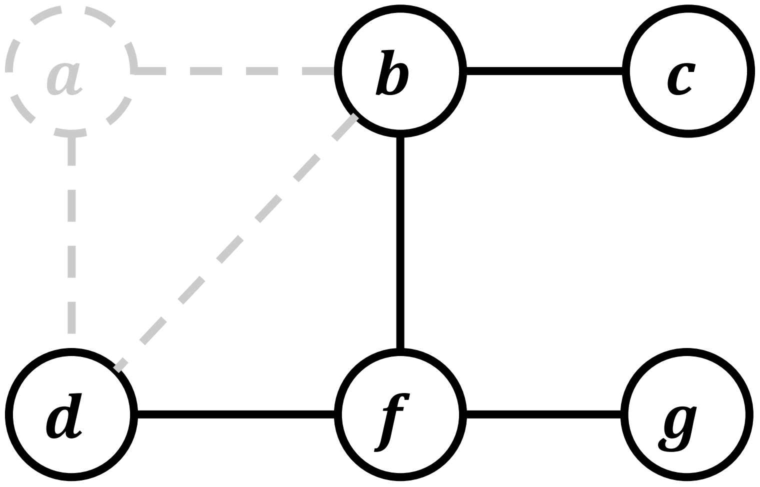
G的子图

图的概念：子图

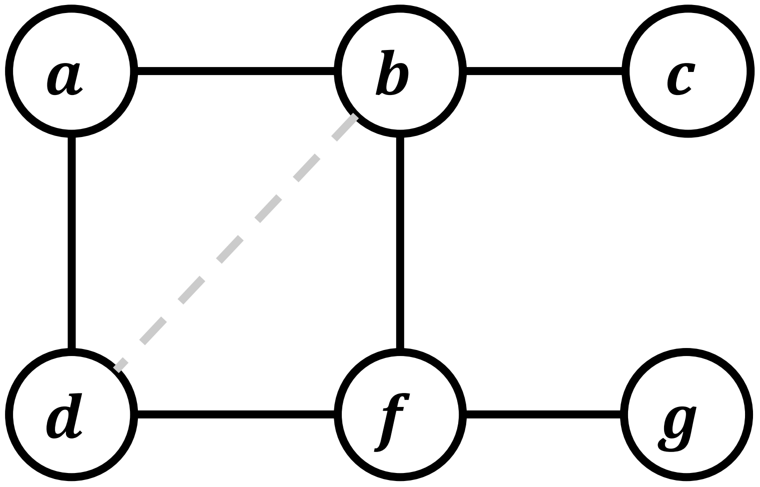
- 子图(Subgraph)
  - 如果$V' \subseteq V, E' \subseteq \mathrm{E}$，则称图$G' =< V', E' >$是图$G$的一个子图
- 生成子图(Spanning Subgraph)
  - 如果$V' = V, E' \subseteq \mathrm{E}$，则称图$G' =< V', E' >$是图$G$的一个生成子图



图G



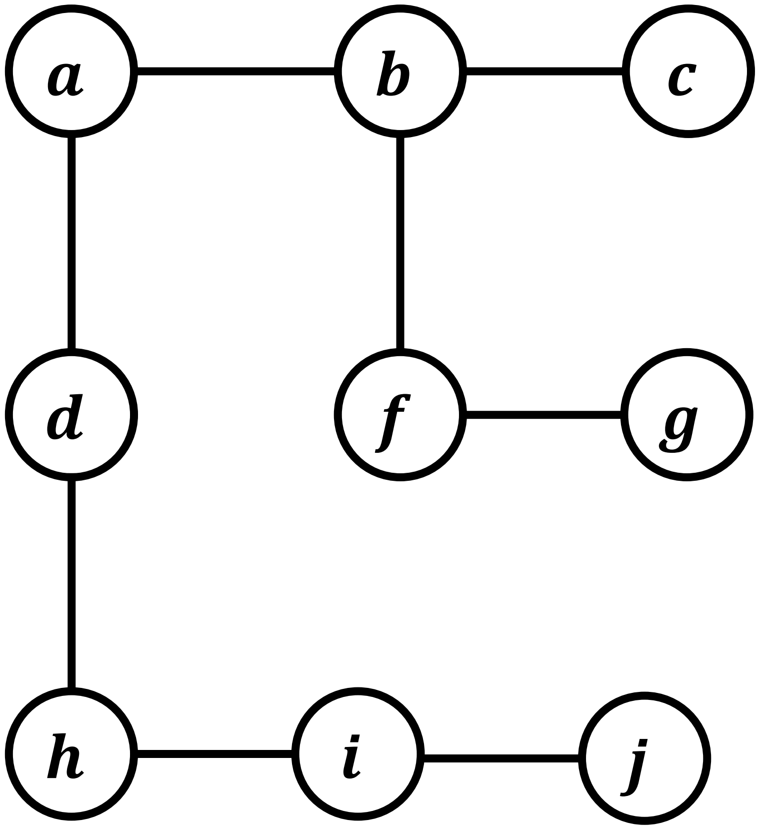
G的子图



G的生成子图

图的概念：树

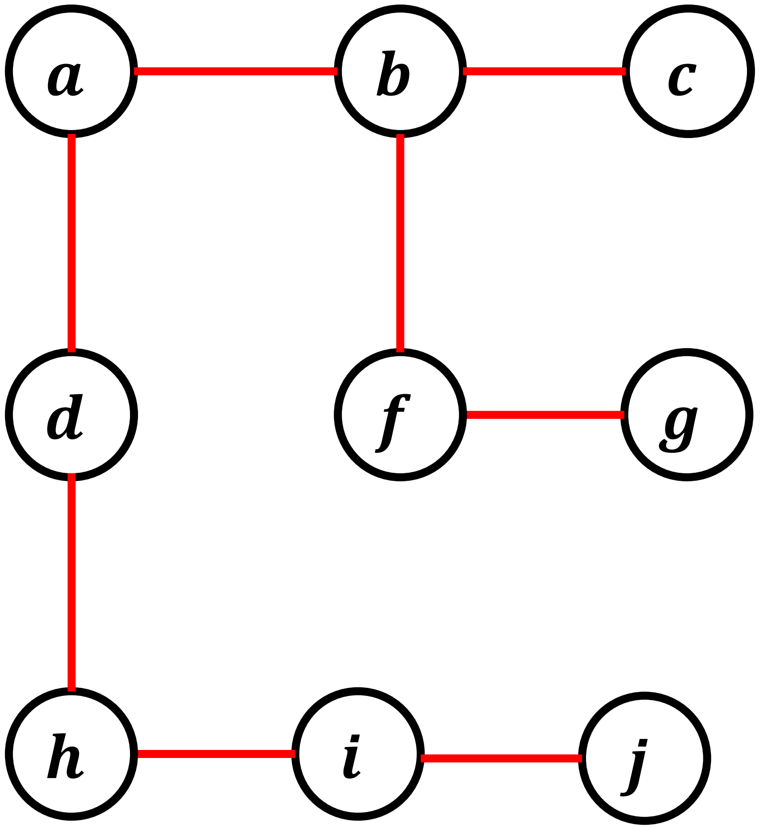
- **树(Tree)**
  - 连通、无环图$T=<V_T,E_T>$



树

图的概念：树

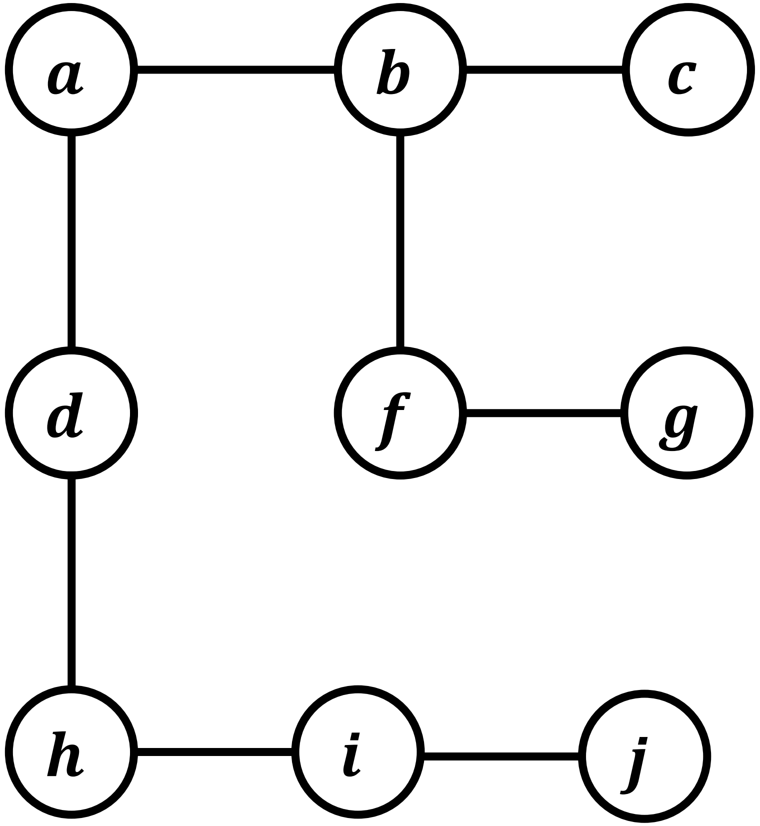
- 树(Tree)
  - 连通、无环图$T=<V_T,E_T>$，树有$|V_T|-1$条边



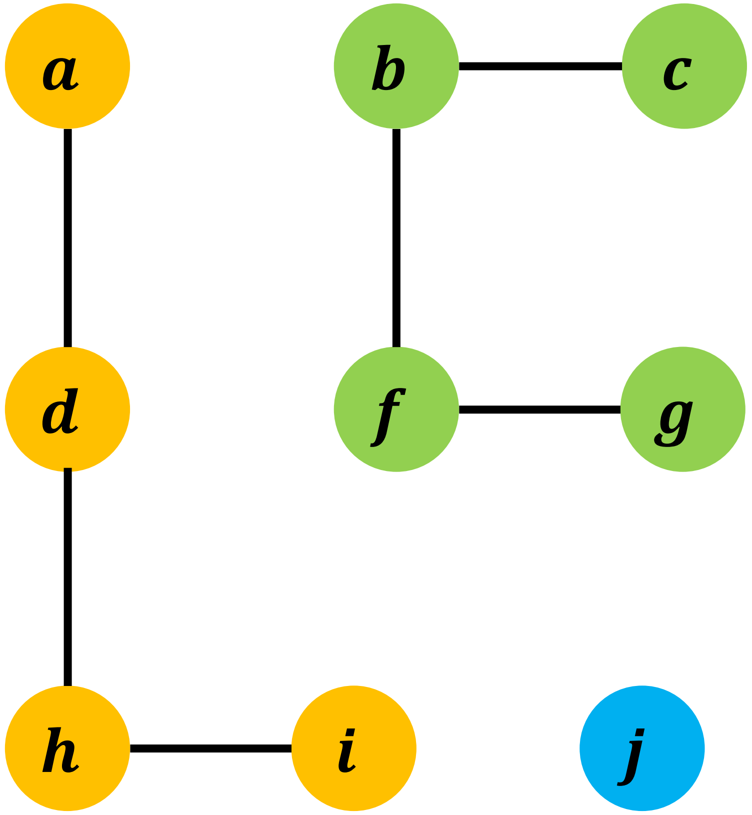
树

图的概念：树

- 树(Tree)
  - 连通、无环图$T=<V_T,E_T>$，树有$|V_T|-1$条边
- 森林(Forest)
  - 一至多棵树组成的无环图



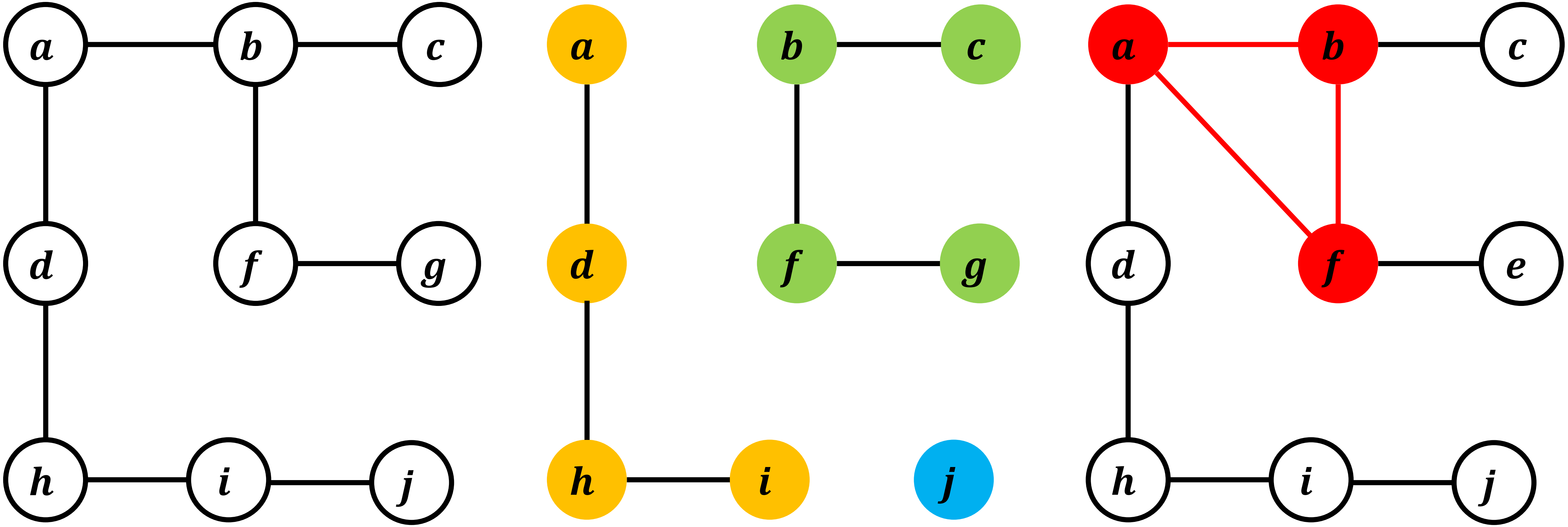
树



森林

图的概念：树

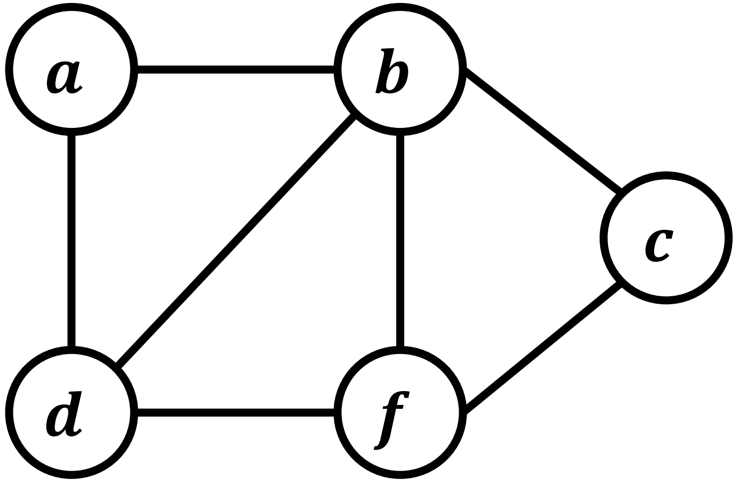
- 树(Tree)
  - 连通、无环图$T=<V_T,E_T>$，树有$|V_T|-1$条边
- 森林(Forest)
  - 一至多棵树组成的无环图



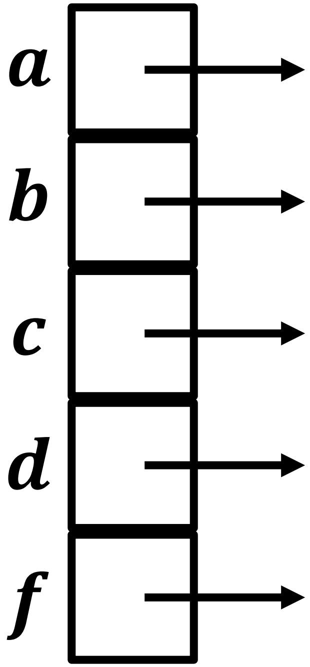
树　　　　森林　　　　非树、非森林

图的表示：邻接链表

- 图$G=<V,E>$，其邻接链表由$|V|$条链表的数组构成

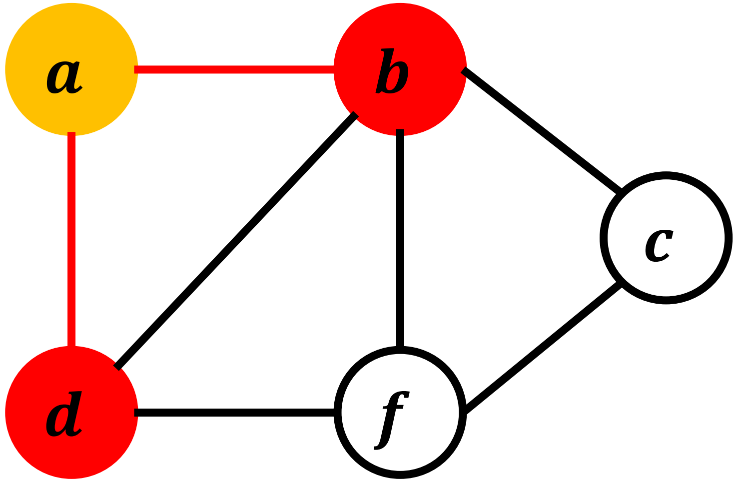


图G

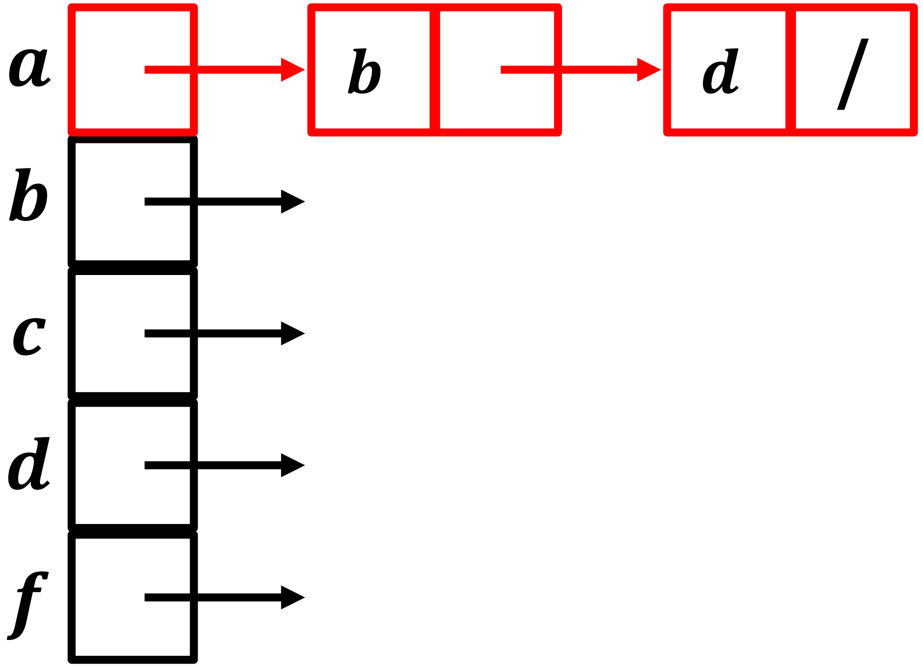


图的表示：邻接链表

- 图$G=<V,E>$，其邻接链表由$|V|$条链表的数组构成
- 每个顶点有一条链表，包含所有与其相邻的顶点
  - $Adj[a]=\{b,d\};$

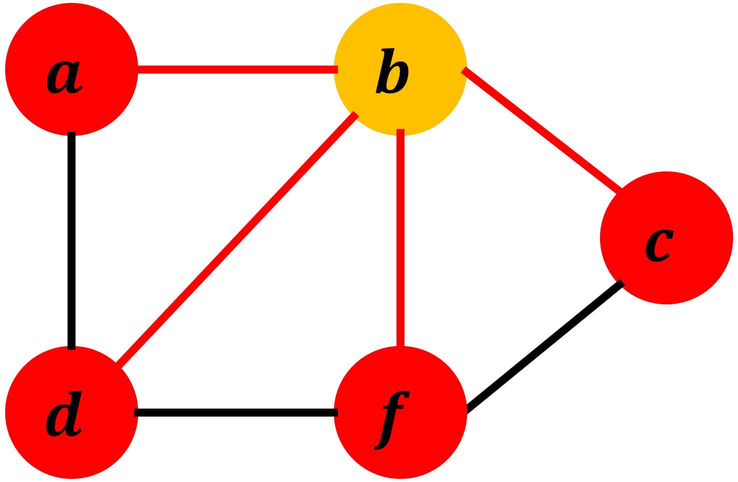


图G

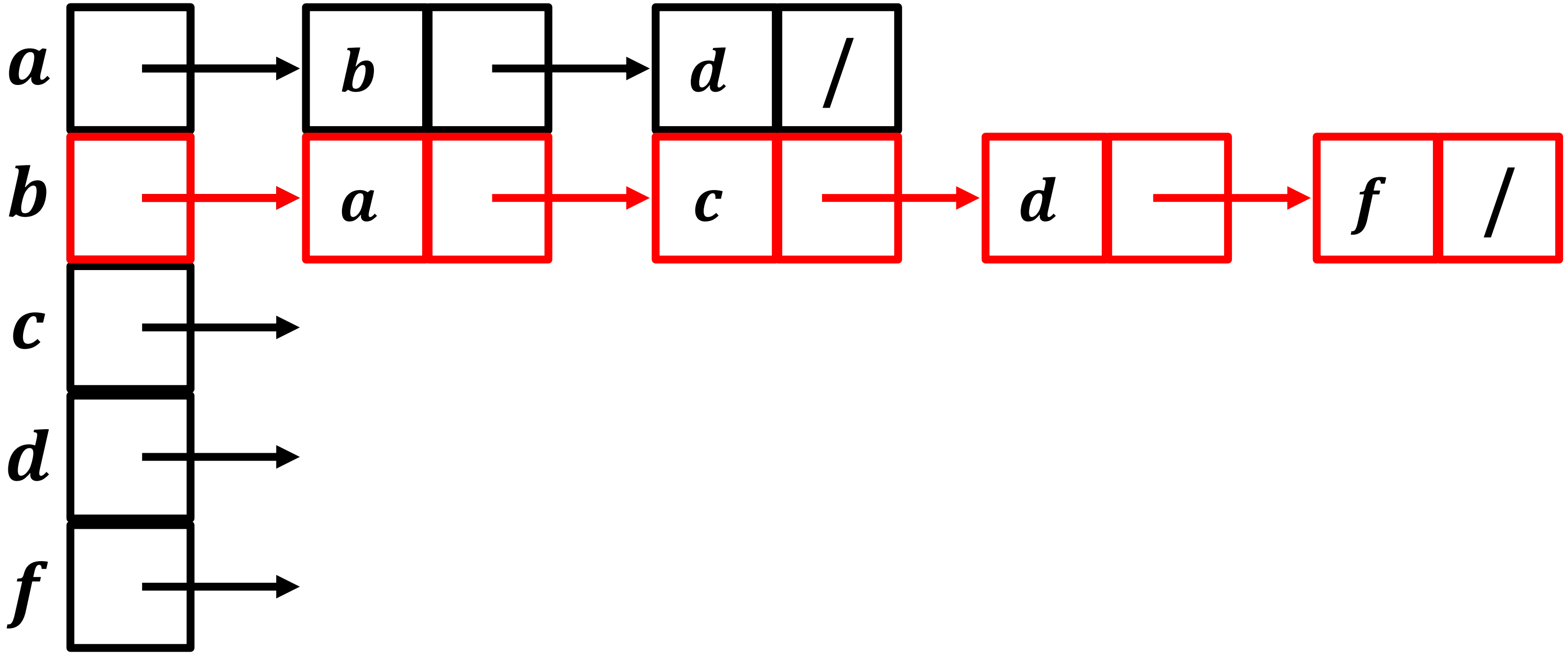


图的表示：邻接链表

- 图$G=<V,E>$，其邻接链表由$|V|$条链表的数组构成
- 每个顶点有一条链表，包含所有与其相邻的顶点
  - $Adj[a]=\{b,d\};Adj[b]=\{a,c,d,f\};$

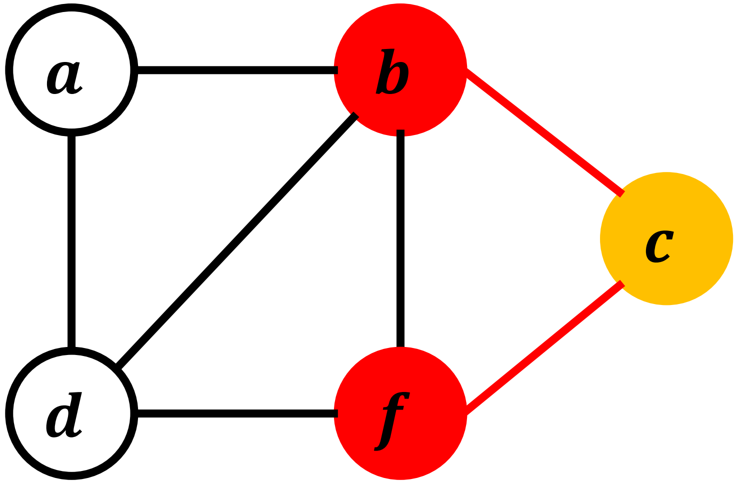


图G

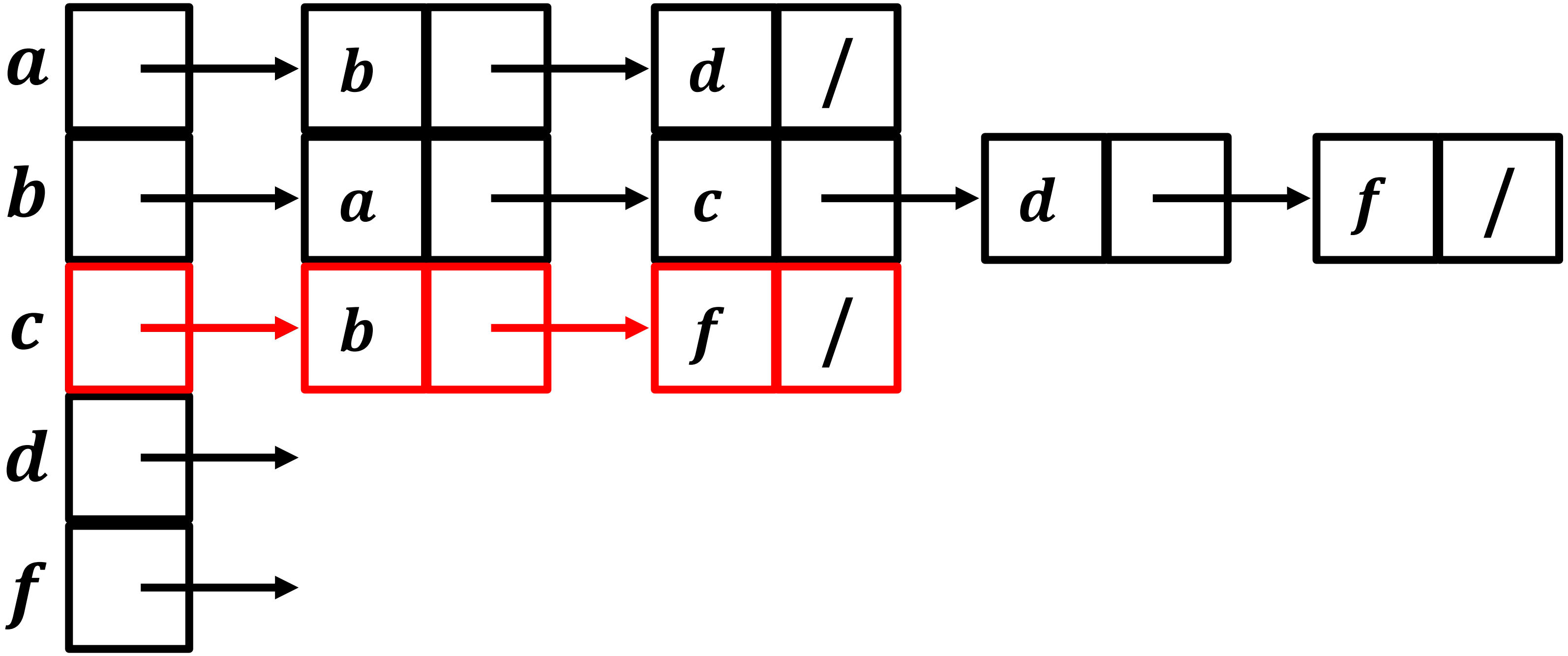


图的表示：邻接链表

- 图$G =< V, E >$，其邻接链表由$|V|$条链表的数组构成
- 每个顶点有一条链表，包含所有与其相邻的顶点
  - $Adj[a] = \{b, d\}; Adj[b] = \{a, c, d, f\}; Adj[c] = \{b, f\};$

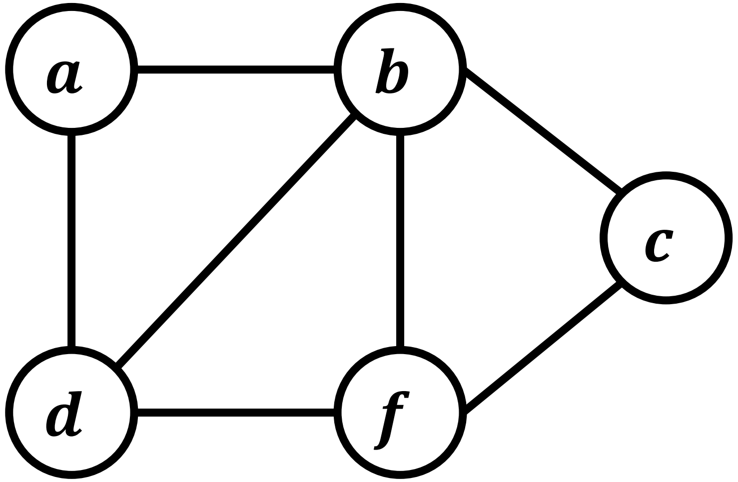


图G

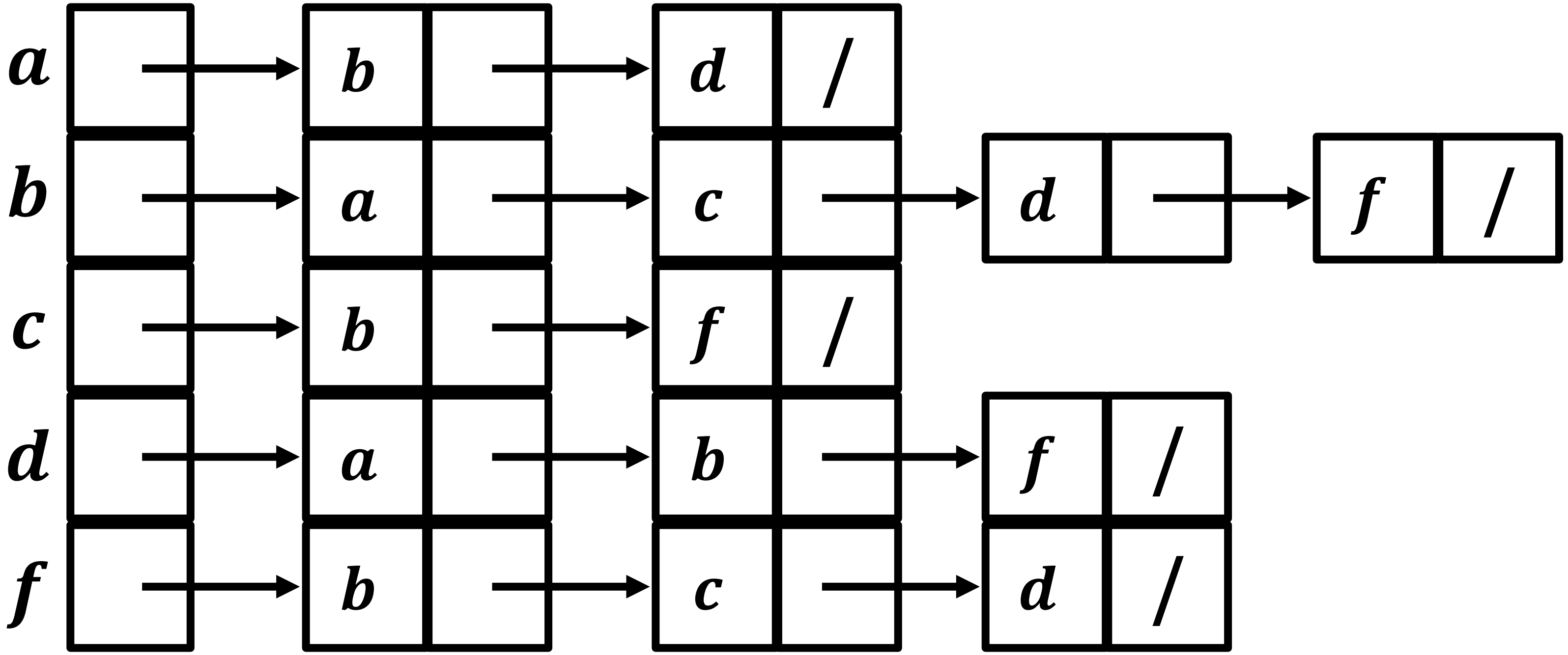


图的表示：邻接链表

- 图$G=<V,E>$，其邻接链表由$|V|$条链表的数组构成
- 每个顶点有一条链表，包含所有与其相邻的顶点
  - $Adj[a]=\{b,d\}; Adj[b]=\{a,c,d,f\}; Adj[c]=\{b,f\}; \ldots$

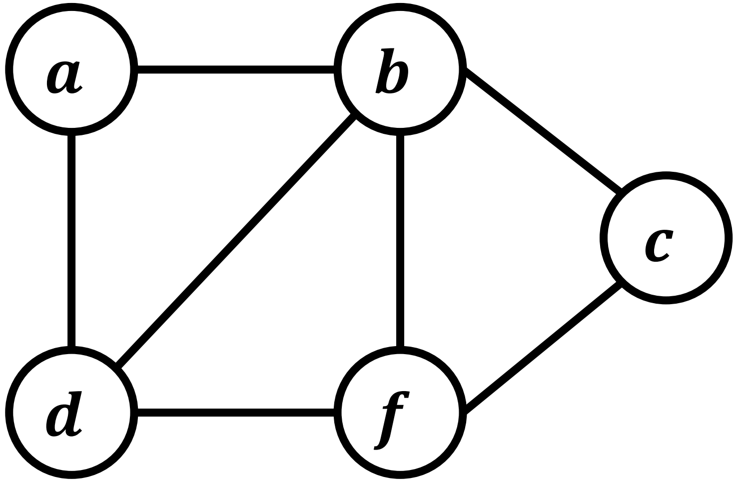


图G

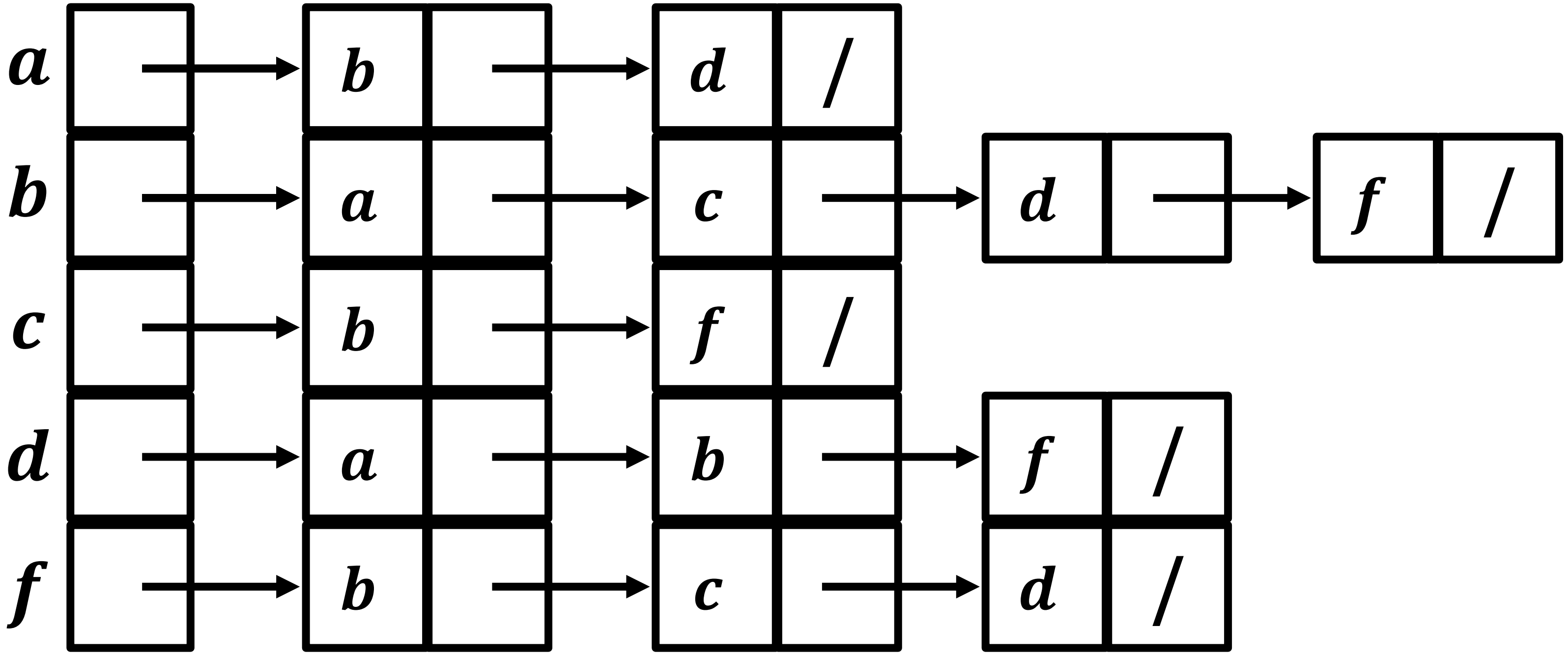


图的表示：邻接链表

- 图$G =< V, E >$，其邻接链表由$|V|$条链表的数组构成
- 每个顶点有一条链表，包含所有与其相邻的顶点
  - $Adj[a] = \{b, d\}; Adj[b] = \{a, c, d, f\}; Adj[c] = \{b, f\}; \dots$
- 空间大小$O(|V| + |E|)$



图G



图的表示：邻接矩阵

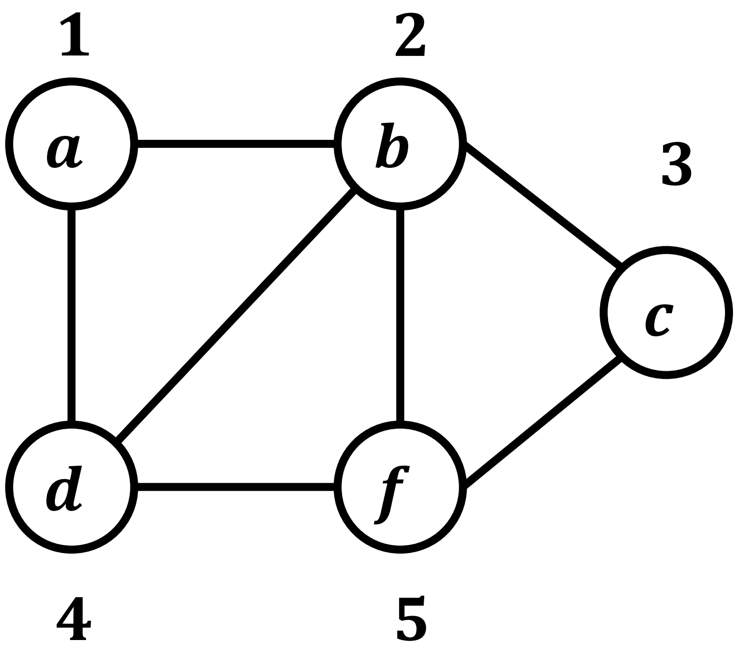
- 图$G=< V,E >$的邻接矩阵由$|V|\times|V|$的二维数组A构成，满足:

$$A_{ij}=\begin{cases}1, & (i,j)\in E\\ 0, & (i,j)\notin E\end{cases}$$

图的表示：邻接矩阵

- 图$G=<V,E>$的邻接矩阵由$|V|\times|V|$的二维数组A构成，满足:

$$A_{ij}=\begin{cases}1, & (i,j)\in E\\ 0, & (i,j)\notin E\end{cases}$$

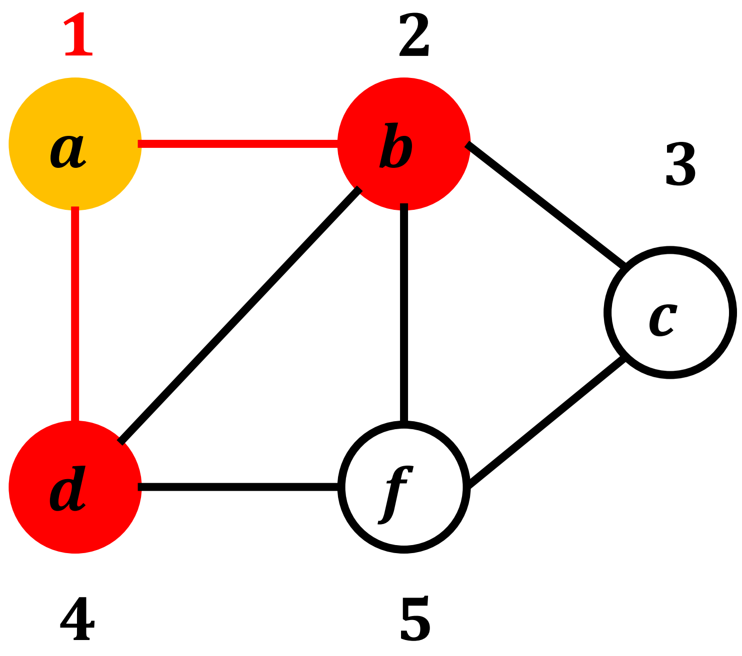


		a	b	c	d	f
		1	2	3	4	5
a	1					
b	2					
c	3					
d	4					
f	5					

图的表示：邻接矩阵

- 图$G=<V,E>$的邻接矩阵由$|V|\times|V|$的二维数组A构成，满足:

$$A_{ij}=\begin{cases}1, & (i,j)\in E\\ 0, & (i,j)\notin E\end{cases}$$

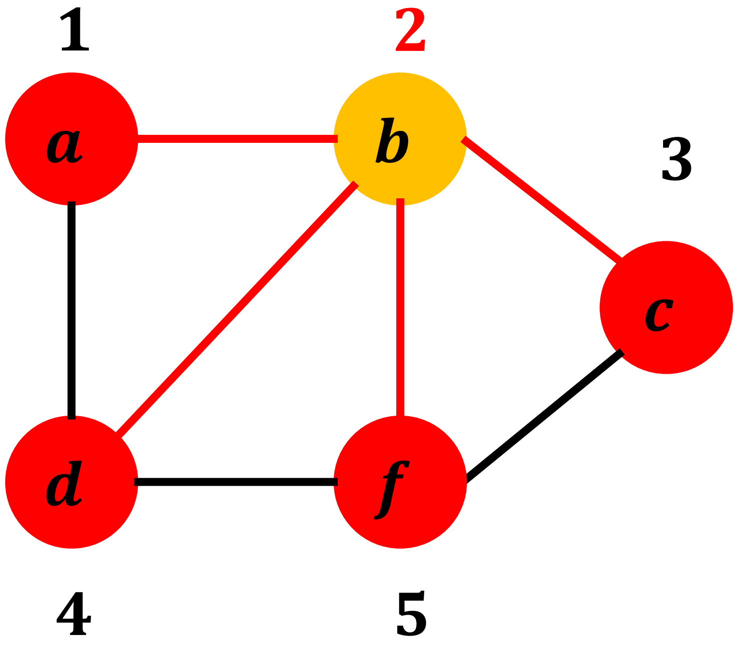


		a	b	c	d	f
		1	2	3	4	5
a	1	0	1	0	1	0
b	2					
c	3					
d	4					
f	5					

图的表示：邻接矩阵

- 图$G =< V, E >$的邻接矩阵由$|V| \times |V|$的二维数组A构成，满足:

$$A_{ij} = \begin{cases} 1, & (i,j) \in E \\ 0, & (i,j) \notin E \end{cases}$$

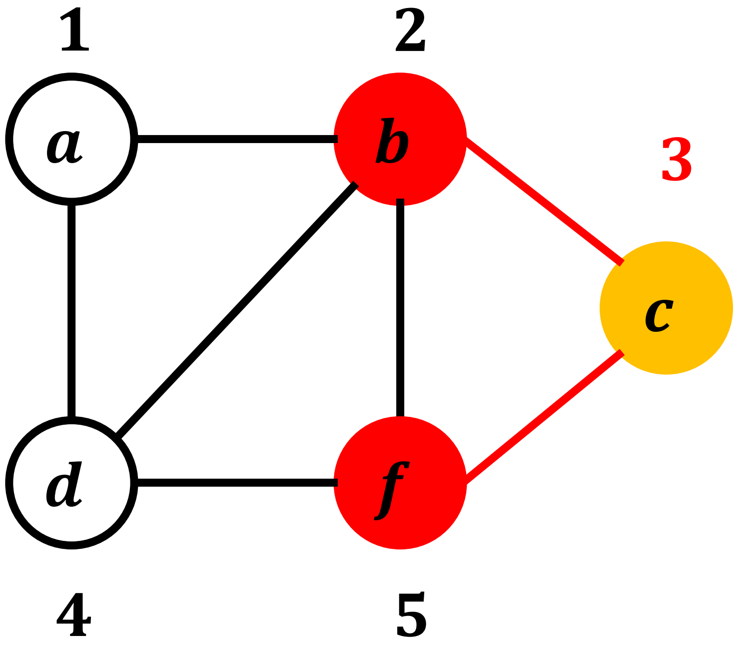


		a	*b*	*c*	*d*	*f*
		1	2	3	4	5
a	1	0	1	0	1	0
b	2	1	0	1	1	1
c	3					
d	4					
f	5					

图的表示：邻接矩阵

- 图$G=<V,E>$的邻接矩阵由$|V|\times|V|$的二维数组A构成，满足:

$$A_{ij}=\begin{cases}1, & (i,j)\in E\\ 0, & (i,j)\notin E\end{cases}$$



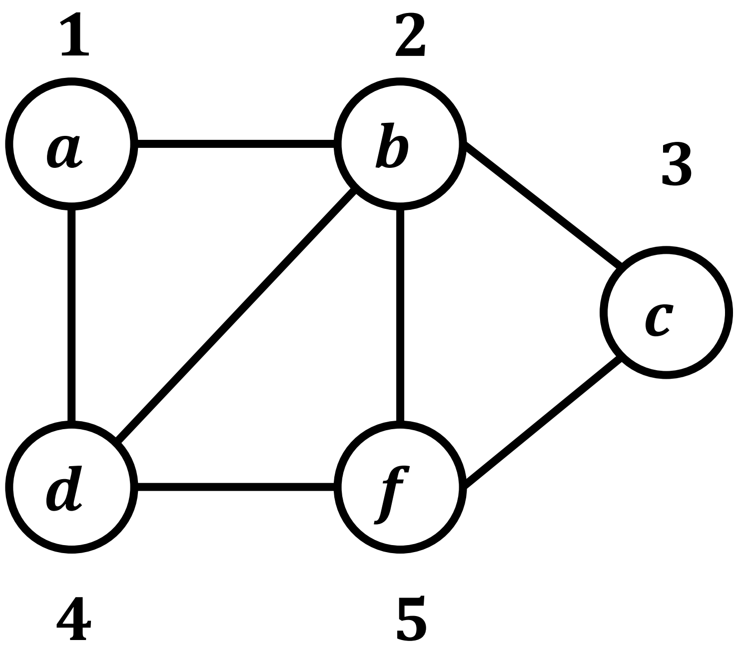
		a	b	c	d	f
		1	2	3	4	5
a	1	0	1	0	1	0
b	2	1	0	1	1	1
c	3	0	1	0	0	1
d	4					
f	5					

图的表示：邻接矩阵

- 图$G=<V,E>$的邻接矩阵由$|V|\times|V|$的二维数组A构成，满足:

$$A_{ij}=\begin{cases}1, & (i,j)\in E\\ 0, & (i,j)\notin E\end{cases}$$

- 空间大小$O(|V|^2)$， $O(1)$判断是否有边



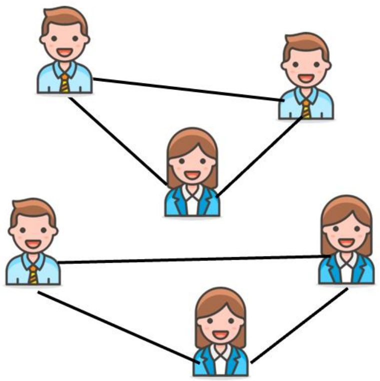
		a	b	c	d	f
		1	2	3	4	5
a	1	0	1	0	1	0
b	2	1	0	1	1	1
c	3	0	1	0	0	1
d	4	1	1	0	0	1
f	5	0	1	1	1	0

小结

- 图的概念
  - 图的定义、相邻与关联

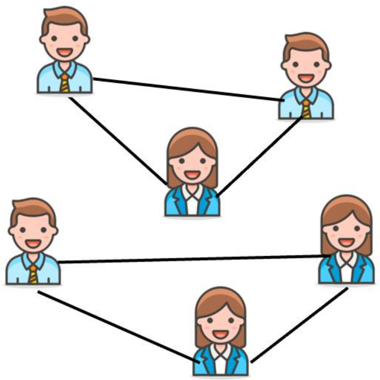
小结

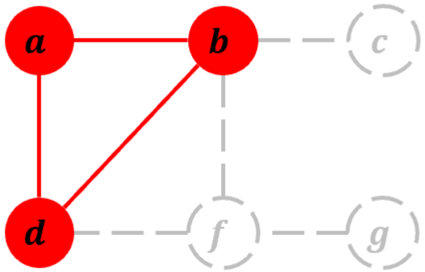
- 图的概念
  - 图的定义、相邻与关联
  - 顶点的度与图的度、握手定理



小结

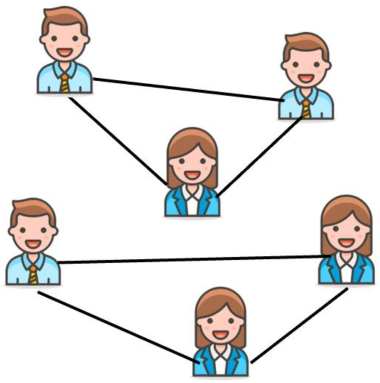
- 图的概念
  - 图的定义、相邻与关联
  - 顶点的度与图的度、握手定理
  - 路径与环路

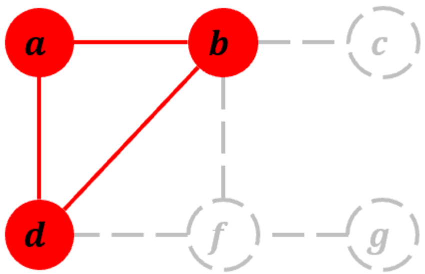


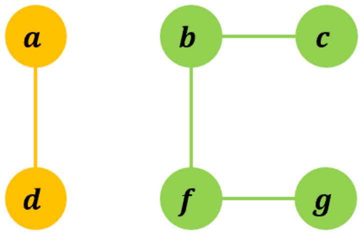


小结

- 图的概念
  - 图的定义、相邻与关联
  - 顶点的度与图的度、握手定理
  - 路径与环路
  - 连通、连通分量





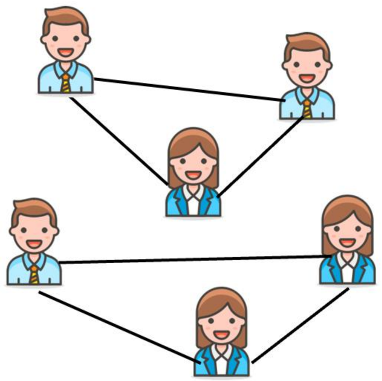


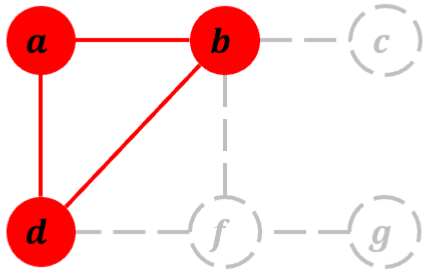
连通分量1　　连通分量2

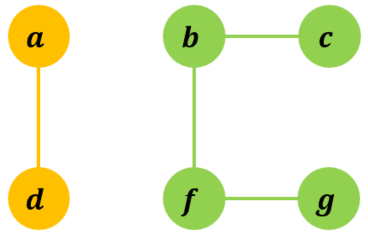
非连通图

小结

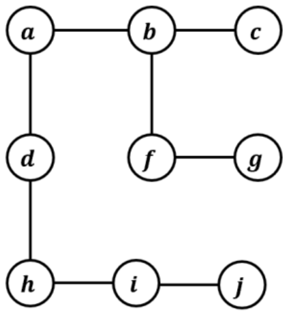
- 图的概念
  - 图的定义、相邻与关联
  - 顶点的度与图的度、握手定理
  - 路径与环路
  - 连通、连通分量
  - 子图、生成子图、树







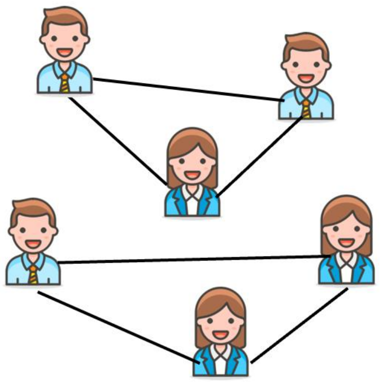
非连通图

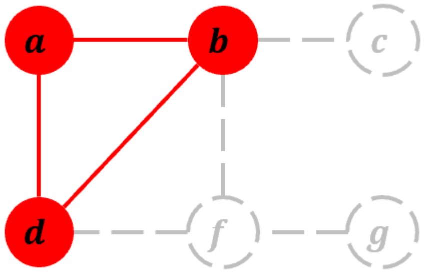


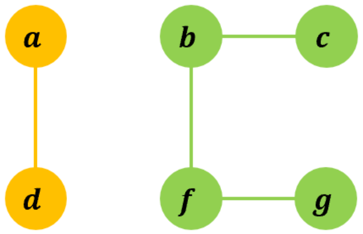
树

小结

- 图的概念
  - 图的定义、相邻与关联
  - 顶点的度与图的度、握手定理
  - 路径与环路
  - 连通、连通分量
  - 子图、生成子图、树
- 图的表示
  - 邻接链表与邻接矩阵

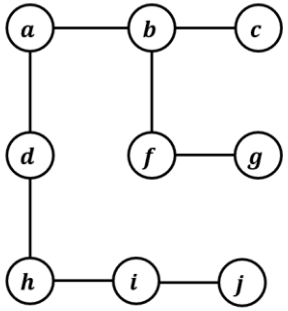




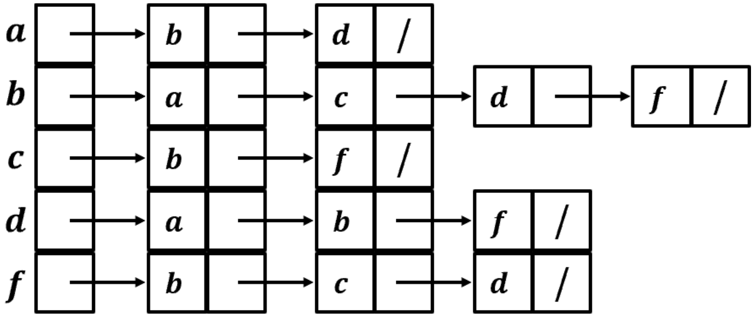


连通分量1 连通分量2

非连通图



树



		a	*b*	*c*	*d*	*f*
		1	2	3	4	5
a	1	0	1	0	1	0
b	2	1	0	1	1	1
c	3	0	1	0	0	1
d	4	1	1	0	0	1
f	5	0	1	1	1	0

谢谢

BDA

Beihang University