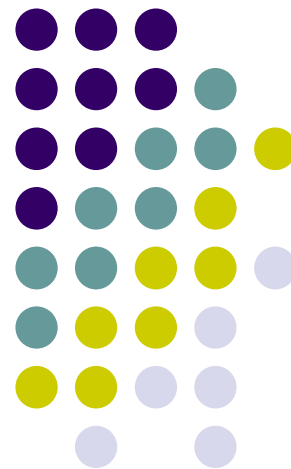


基本概念

离散数学—图论初步

南京大学计算机科学与技术系



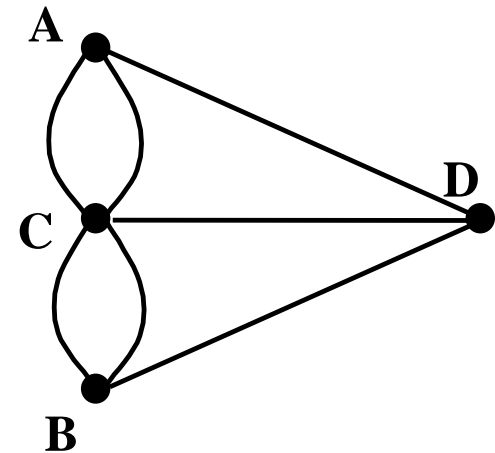
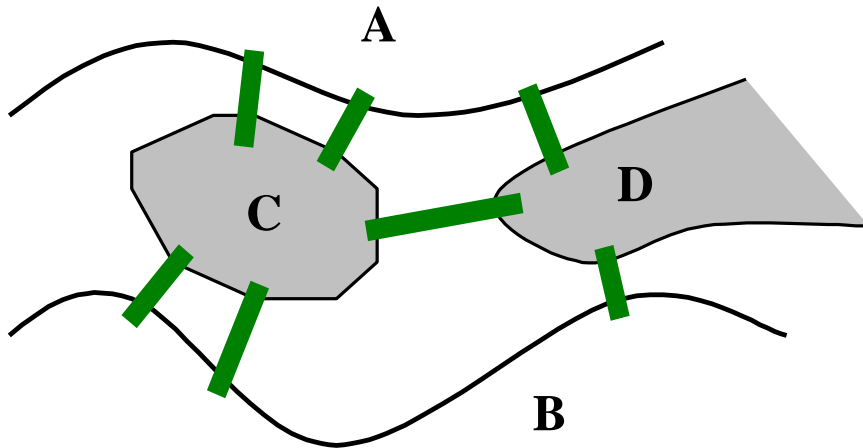
内容提要

- 图的定义
- 用图建模
- 图的表示
- 图的运算
- 图的同构



Königsberg七桥问题

- 问题的抽象：
 - 用顶点表示对象-“地块”
 - 用边表示对象之间的关系-“有桥相连”

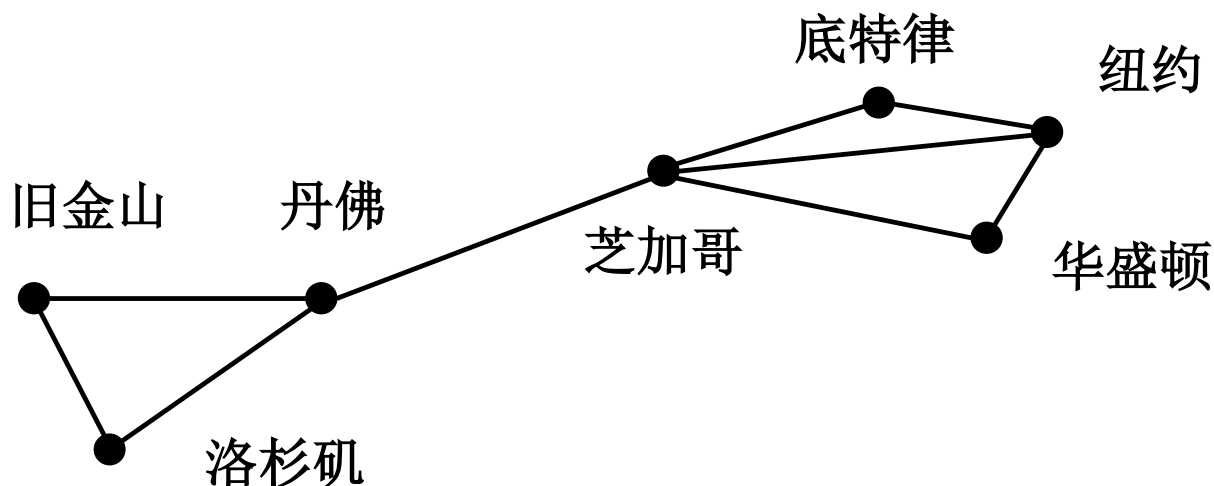




图的定义 Graph

φ 常常省略，写作：
 $G = (V, E)$

- 图 G 是一个三元组： $G = (V, E, \varphi)$
 - V 是**非空**顶点集， E 是边集，且 $V \cap E = \phi$;
 - $\varphi: E \rightarrow P(V)$, 且 $\forall e \in E. 1 \leq |\varphi(e)| \leq 2$. $\varphi(e)$ 称为边 e 的端点集.
- 举例（数据中心、通信链接）



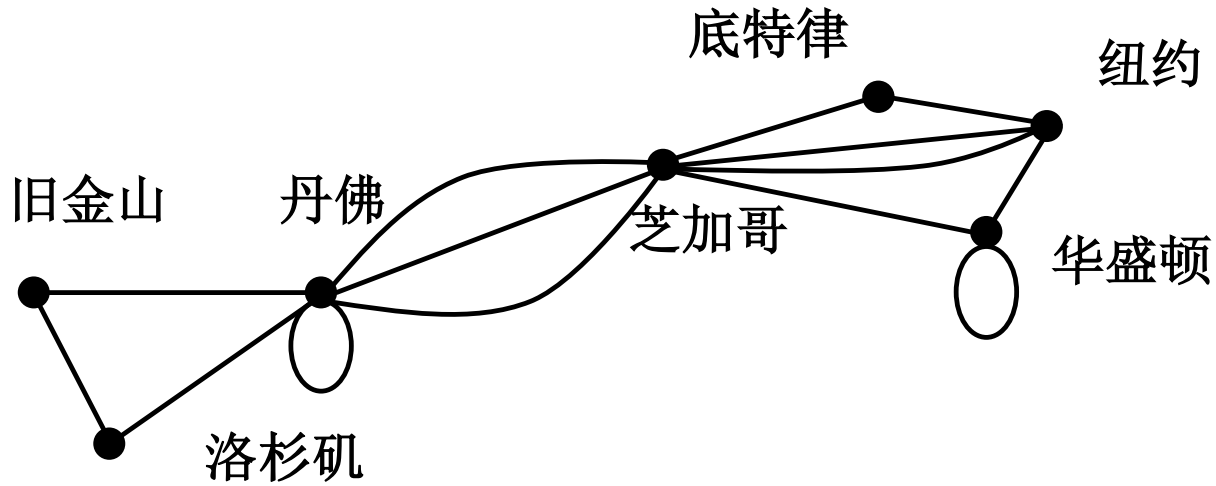


图的定义（续）

- 图 $G = (V, E, \varphi)$ 是**简单图**，如果
 - 每条边有2个端点，即： $\forall e \in E. |\varphi(e)| = 2$ ， **并且**
 - 不同边有不同端点集，即： 如果 $e_1 \neq e_2$ ， 则 $\varphi(e_1) \neq \varphi(e_2)$
- 图 $G = (V, E, \varphi)$ 是**伪图**， 如果
 - 存在一条只有1个端点的边， 即： $\exists e_0 \in E. |\varphi(e_0)| = 1$ ， **或者**
 - 有两条边具有相同的端点集， 即： $\exists e_1 \neq e_2. \varphi(e_1) = \varphi(e_2)$

图的定义（续）

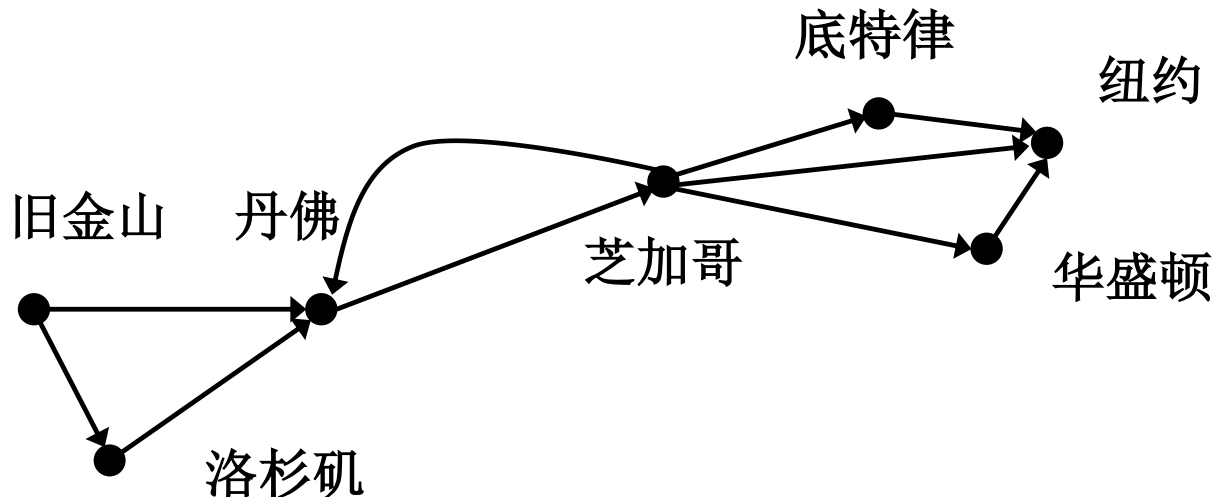
- 伪图（包含环或者多重边）示例





图的定义（有向图）

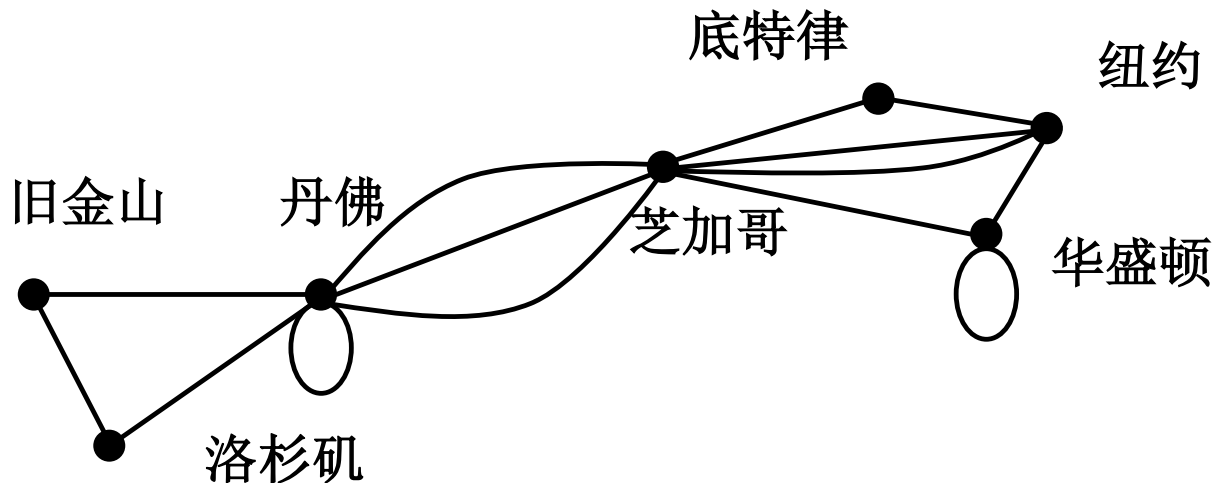
- 有向图 G 是一个三元组： $G = (V, E, \varphi)$
 - V 是非空顶点集， E 是有向边（弧）集，且 $V \cap E = \phi$ ；
 - $\varphi: E \rightarrow V \times V$, 若 $\varphi(e) = (u, v)$, 则 u 和 v 分别称为 e 的起点和终点.
- 举例（简单有向图）





图的术语

- 无向图 $G = (V, E, \varphi)$, $\varphi(e) = \{u, v\}$
 - u 和 v 在 G 里邻接（相邻）
 - e 关联（连接）顶点 u 和 v
- 图 G 中顶点 v 的度, $\text{deg}(v)$, $d_G(v)$
 - 与该顶点关联的边数, 环为顶点的度做出双倍贡献。





握手定理

- 无向图G有m条边，n个顶点 v_1, \dots, v_n .

$$\sum_{i=1}^n d(v_i) = 2m$$

- 推论：无向图中奇数度顶点必是偶数个。



图的术语（续）

- 有向图 $G = (V, E, \varphi)$, $\varphi(e) = (u, v)$
 - u 是 e 的起点, v 是 e 的终点
 - 假设 $u \neq v$, u 邻接到 v , v 从 u 邻接
- 有向图中顶点的出度和入度
 - $d_G^+(v)$ = 以 v 为始点的边的条数, $\text{deg}^+(v)$
 - $d_G^-(v)$ = 以 v 为终点的边的条数, $\text{deg}^-(v)$
- 有向图中各顶点的出度之和等于入度之和。

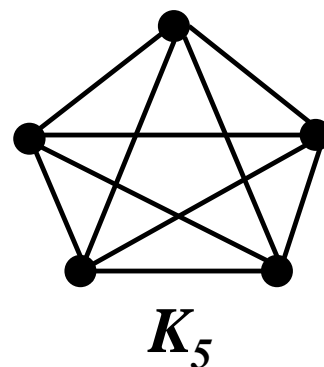
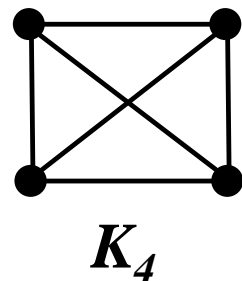
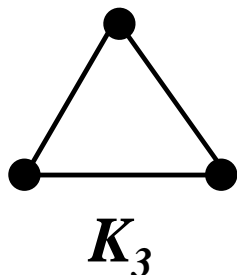
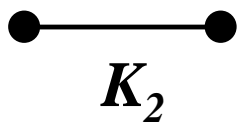
$$\sum_{v \in V} \text{deg}^+(v) = \sum_{v \in V} \text{deg}^-(v) = |E|$$

- 有向图的底图



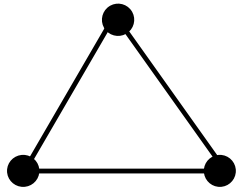
特殊的简单图（完全图）

- 若简单图 G 中任意两点均相邻，则称为完全图。记为 K_n ，其中 n 是图中顶点数。
 - K_n 中每个顶点皆为 $n-1$ 度，总边数为 $n(n-1)/2$ 。
 - 边数达到上限的简单图。

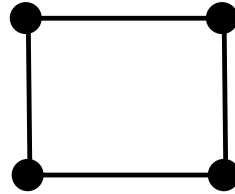


特殊的简单图（圈图与轮图）

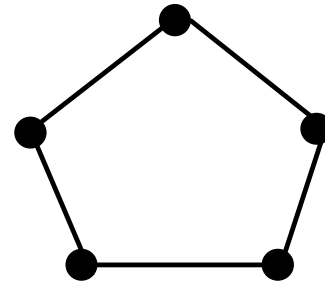
Cycle



C_3

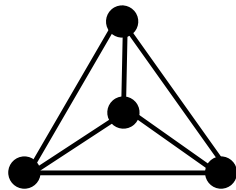


C_4

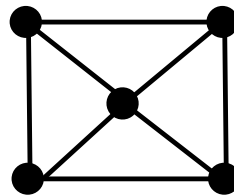


C_5

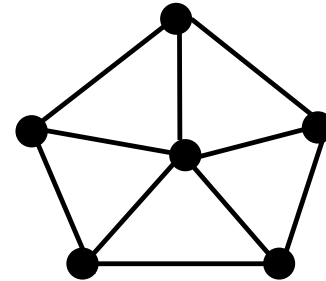
Wheel



W_3



W_4

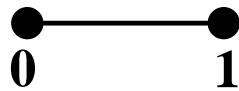


W_5

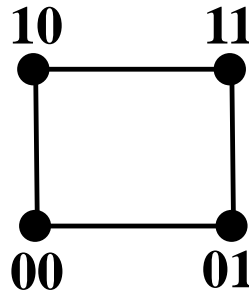


特殊的简单图（立方体图）

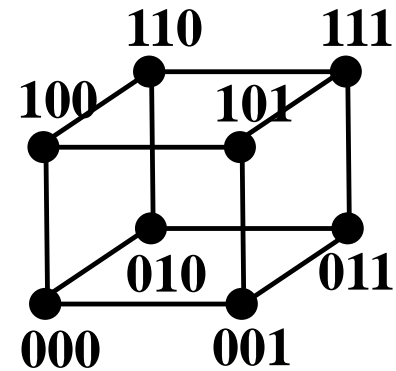
n-cube



Q_1



Q_2



Q_3

正则图：顶点度相同的简单图



子图

- 设 $G=\langle V,E\rangle$, $G'=\langle V',E'\rangle$, 如果 $V'\subseteq V$, $E'\subseteq E$, 则称 G' 是 G 的子图。
- 如果 $V'\subset V$, 或者 $E'\subset E$, 则称为真子图。
- 诱导(导出)子图: 可以由顶点集的子集, 或者由边集的子集导出一个子图。



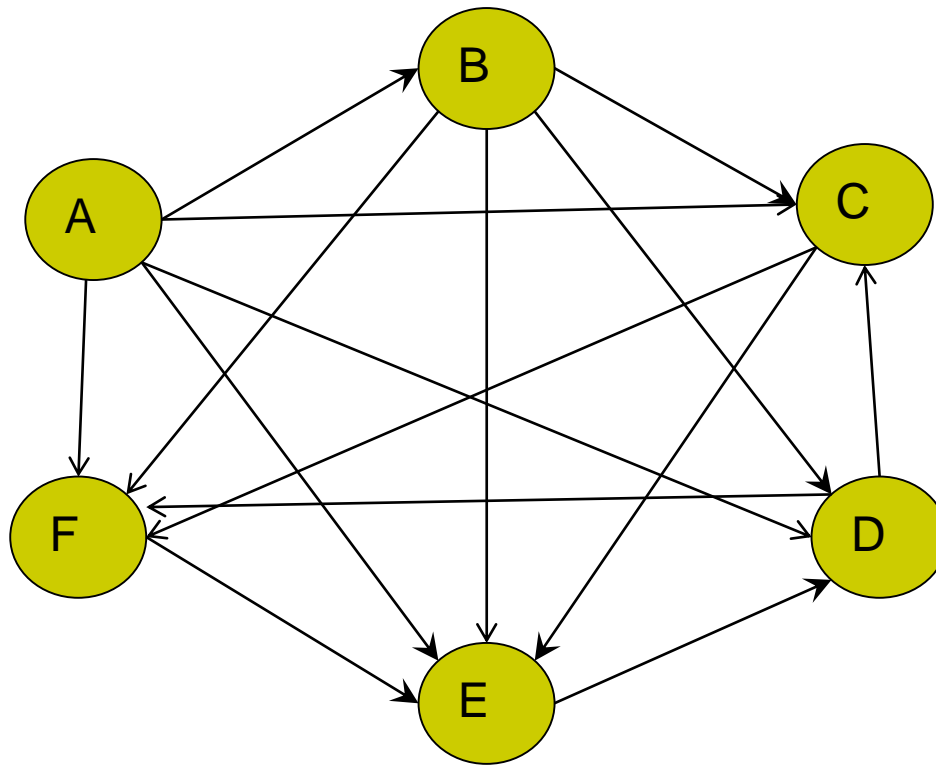
用图建模



图模型

- 交通网络
 - 航空、公路、铁路
- 信息网络
 - 万维网图 (Web Graph)
 - 引用图 (Citation Graph)
- 社会网络
 - 熟人关系图
 - 合作图, 好莱坞图
 - 呼叫图
- 体育 (循环赛的图模型)

循环赛的冠军是哪个队？



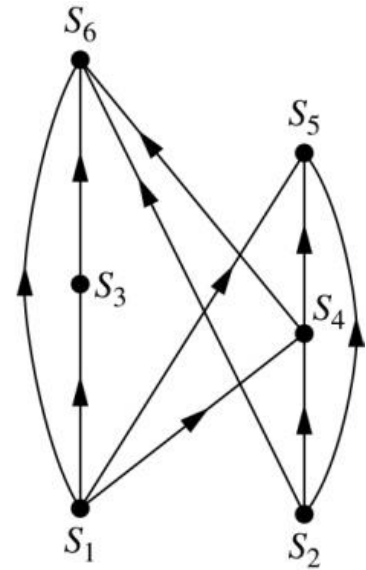
是什么思考帮助我们建模？
问题的答案是什么？

优先图和程序并发执行

- 右边的程序有没有办法执行快一点？

$s_1 || s_2; s_3 || s_4; s_5 || s_6$

$S_1 \quad a := 0$
 $S_2 \quad b := 1$
 $S_3 \quad c := a + 1$
 $S_4 \quad d := b + a$
 $S_5 \quad e := d + 1$
 $S_6 \quad e := c + d$



是什么思考帮助我们建模？
问题的答案是什么？



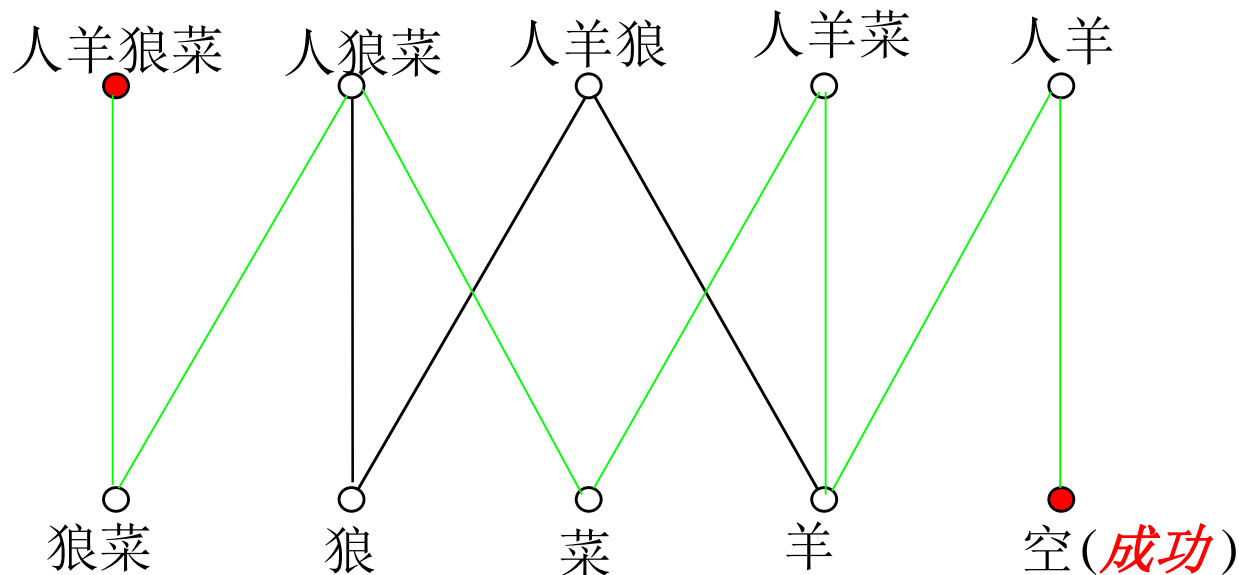
“巧渡河”问题

- 问题：人、狼、羊、菜用一条只能同时载两位的小船渡河，“狼羊”、“羊菜”不能在无人在场时共处，当然只有人能架船。
- 图模型：顶点表示“原岸的状态”，两点之间有边当且仅当一次合理的渡河“操作”能够实现该状态的转变。
- 起始状态是“人狼羊菜”，结束状态是“空”。
- 问题的解：找到一条从起始状态到结束状态的尽可能短的通路。



“巧渡河”问题的解

- 注意：在“人狼羊菜”的16种组合种允许出现的只有10种。





考试时间编排问题

- 问题：排考试时间，一方面要总时间尽可能短(假设教室没问题)，另一方面一个同学所选的任意两门课不能同时间。
- 图模型：每门课程对应一个顶点。任意两点相邻当且仅当对应的两门课程有相同的选课人。
- 解：用不同颜色给顶点着色。相邻的点不能同颜色。则最少着色数即至少需要的考试时间段数(可以将颜色相同的点所对应的课程安排在同一时间)。

中国邮递员问题（管梅谷，1960）



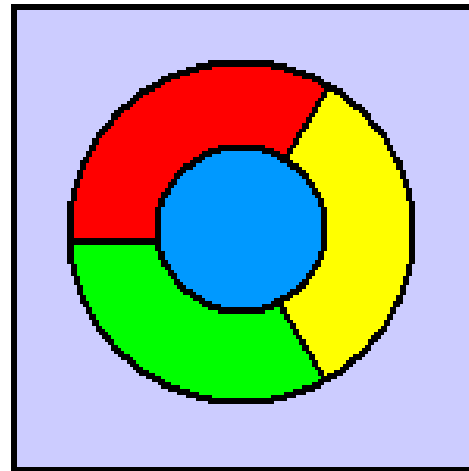
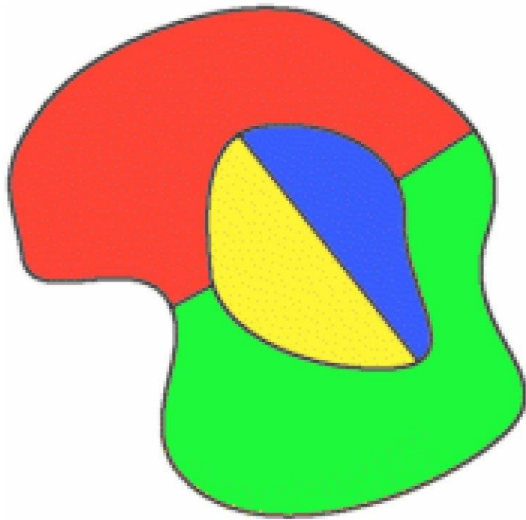
- 邮递员从邮局出发，走过辖区内**每条街道至少一次**，再回邮局，如何选择最短路线？
- **Euler**回路？添加重复边（权和最小）。



旅行商(TSP)问题

- n 个城市间均有道路，但距离不等，旅行商从某地出发，走过其它 $n-1$ 个城市，且只经过一次，最后回到原地，如何选择最短路线？
- 最短Hamilton回路。

地图与平面图着色（四色定理）





图的表示



关联矩阵 (*incidence matrix*)

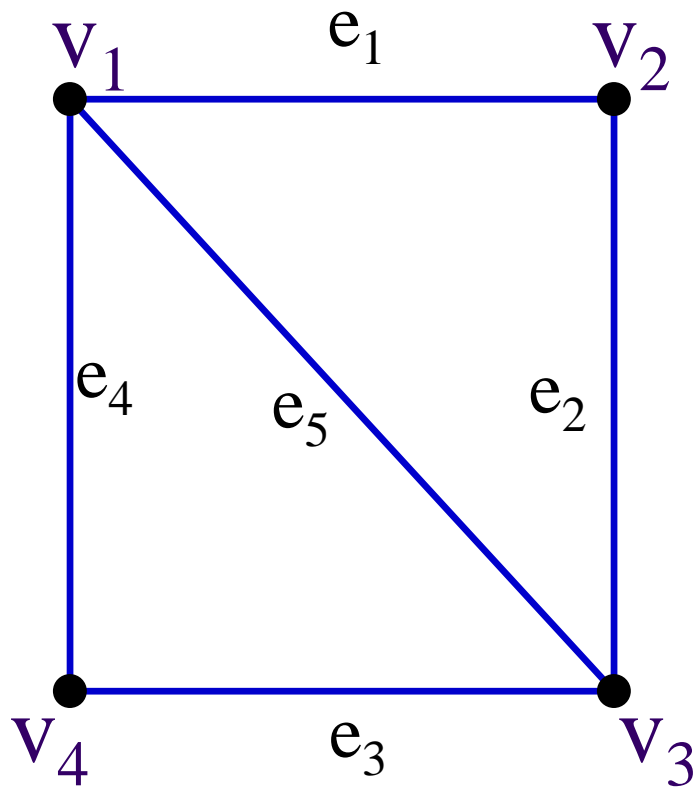
- 无向图 $G = (V, E, \varphi)$ ，不妨设 $V = \{v_1, \dots, v_n\}$ ， $E = \{e_1, \dots, e_m\}$ 。
- $M(G) = [m_{ij}]$ 称为 G 的关联矩阵 ($n \times m$ 阶矩阵)，其中

$$m_{ij} = \begin{cases} 1 & \text{如果 } e_j \text{ 关联 } v_i \\ 0 & \text{否则} \end{cases}$$

$v_i \in \varphi(e_j)$

- 无向图 G 可以是伪图(含环或多重边)。

举例（关联矩阵）



$$M(G) = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

关联矩阵表示法 **不适合于有向图**



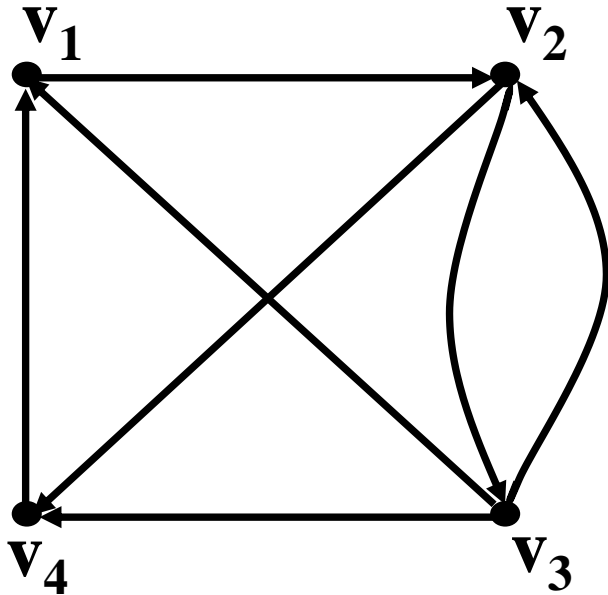
邻接矩阵 (*adjacency matrix*)

- 简单有向图 $G = (V, E, \varphi)$ ，设 $V = \{v_1, \dots, v_n\}$ ， $E = \{e_1, \dots, e_m\}$ 。
- $A(G) = [a_{ij}]$ 称为 G 的邻接矩阵 ($n \times n$ 阶矩阵)，其中

$$a_{ij} = \begin{cases} 1 & \text{如果 } v_i \text{ 邻接到 } v_j \\ 0 & \text{否则} \end{cases}$$

$\exists e \in E. \varphi(e) = (v_i, v_j)$

举例（邻接矩阵）

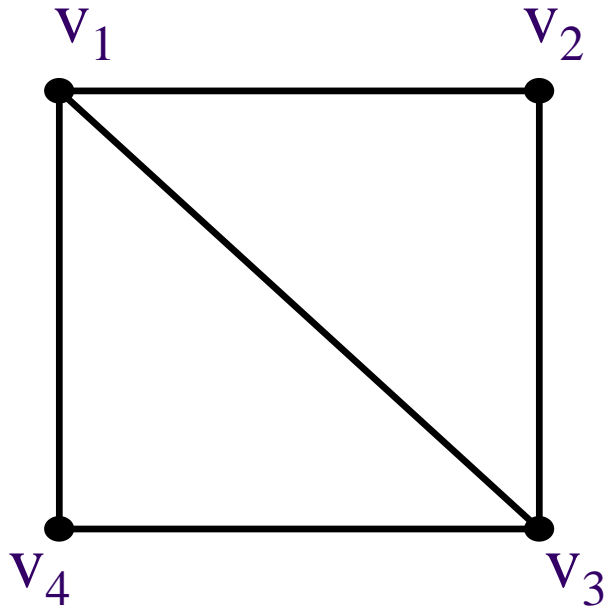


$$A(G) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

可推广到简单无向图



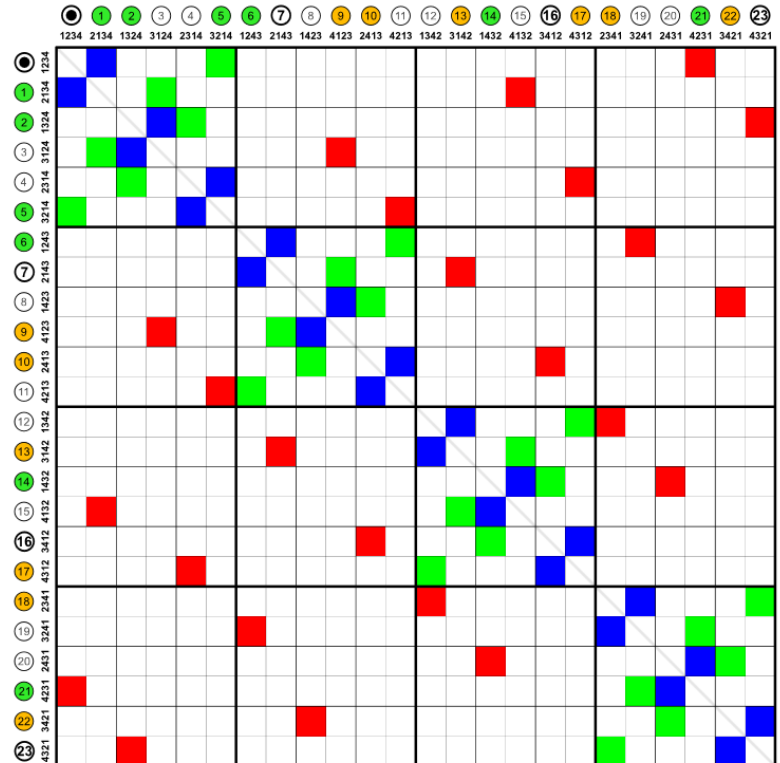
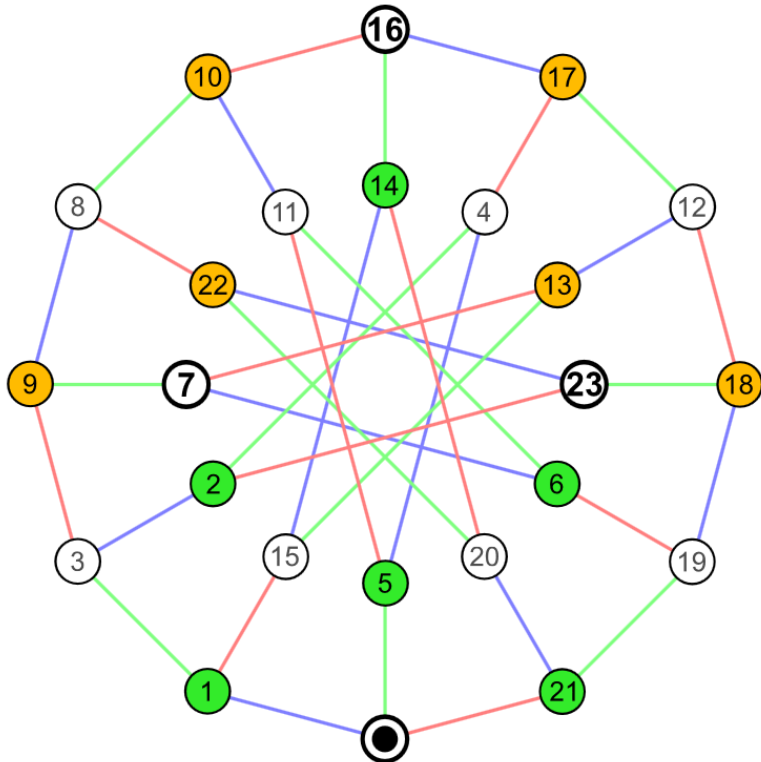
举例（邻接矩阵）



$$A(G) = \begin{Bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{Bmatrix}$$

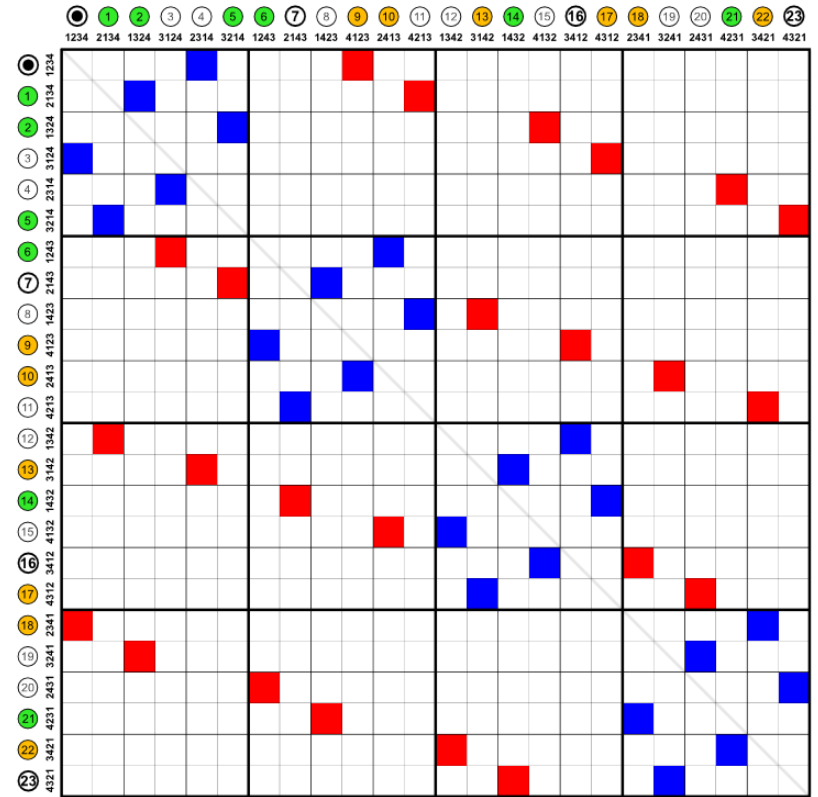
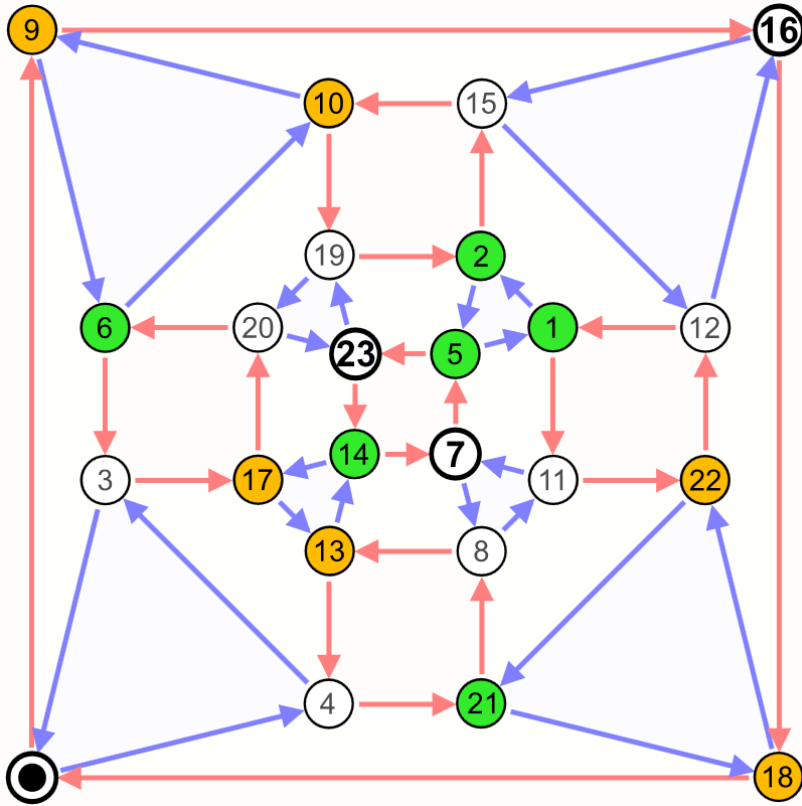
简单无向图的邻接矩阵是对称矩阵

举例（邻接矩阵）



The Nauru graph, from Wikipedia

举例（邻接矩阵）

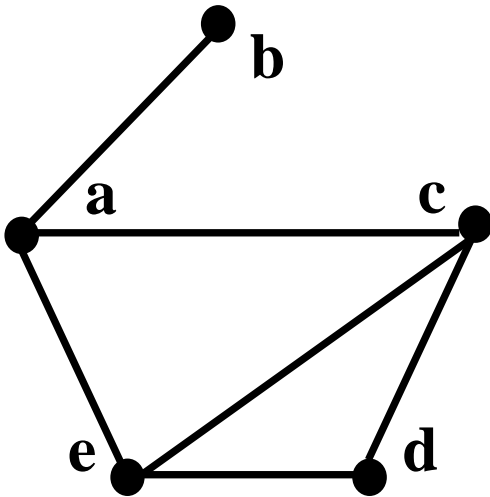


Directed Cayley graph of S_4 , from Wikipedia

邻接表

ϕ 是单射

- 若图 $G = (V, E, \phi)$ 没有多重边，列出这个图的所有边。对每个顶点，列出与其邻接的顶点。

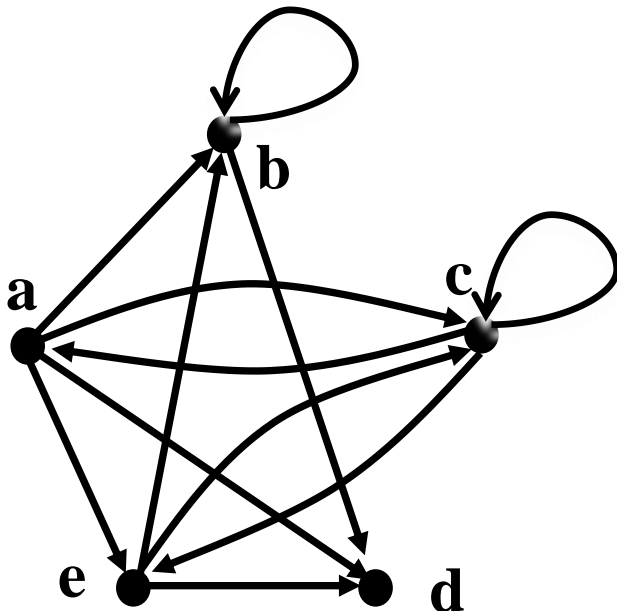


顶点	相邻顶点
a	b, c, e
b	a
c	a, d, e
d	c, e
e	a, c, d

邻接表（有向图）

ϕ 是单射

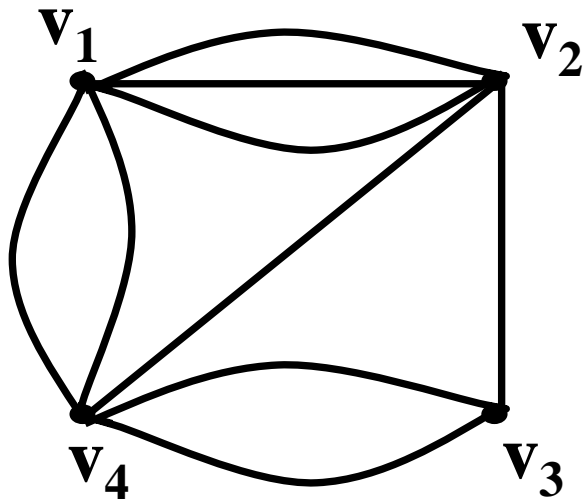
- 若图 $G = (V, E, \phi)$ 没有多重边，列出这个图的所有边。对每个顶点，列出与其邻接的顶点。



顶点	相邻顶点
a	b, c, d, e
b	b, d
c	a, c, e
d	
e	b, c, d

关于邻接矩阵

- 通常，邻接矩阵中的元素为0和1，称为布尔矩阵。
- 邻接矩阵也可表示包含多重边的图，此时的矩阵不是布尔矩阵。



$$A = \begin{pmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{pmatrix}$$



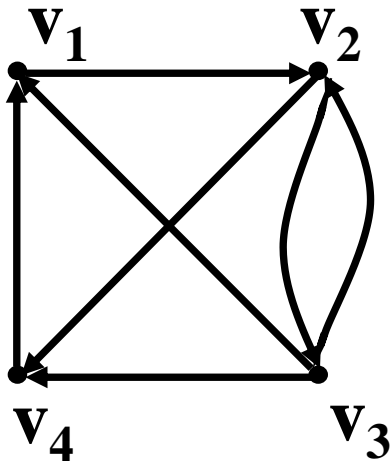
关于邻接矩阵

- 当有向图中的有向边表示关系时，邻接矩阵就是关系矩阵。无向图的邻接矩阵是对称的。
- 图G的邻接矩阵中的元素的次序是无关紧要的，只要进行和行、列和列的交换，则可得到相同的矩阵。
 - 若有二个简单有向图，则可得到二个对应的邻接矩阵，若对某一矩阵进行行和行、列和列之间的交换后得到和另一矩阵相同的矩阵，则此二图同构。

邻接矩阵的运算

- 顶点的度

- 行中1的个数就是行中相应结点的出度
- 列中1的个数就是列中相应结点的入度



$$A = \begin{Bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{Bmatrix}$$

$$\text{Deg}^+(1)=1, \text{Deg}^-(1)=2$$

$$\text{Deg}^+(2)=2, \text{Deg}^-(2)=2$$

$$\text{Deg}^+(3)=3, \text{Deg}^-(3)=1$$

$$\text{Deg}^+(4)=1, \text{Deg}^-(4)=2$$



邻接矩阵的运算

- 逆图（转置矩阵）

- 设 G 的邻接矩阵为 A ，则 G 的逆图的邻接矩阵是 A 的转置矩阵，用 A^T 表示。

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

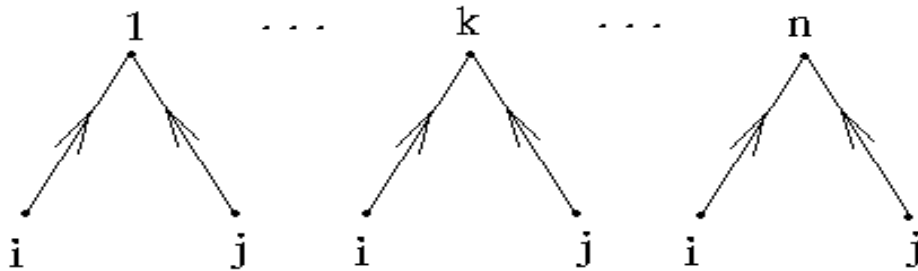
$$A^T = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$



邻接矩阵的运算

$$A \times A^T = B = [b_{ij}]$$

$$b_{ij} = \sum_{k=1}^n a_{ik} \times a_{jk} = a_{i1} \times a_{j1} + a_{i2} \times a_{j2} + \cdots + a_{in} \times a_{jn}$$



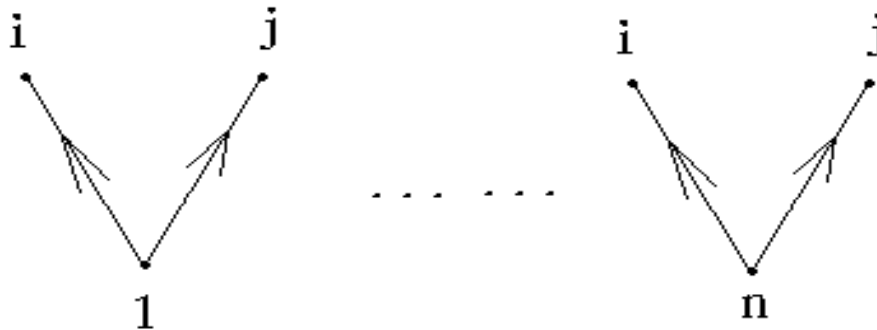
- b_{ij} 表示结点 i 和结点 j 均有边指向的那些结点的个数；
- 若 $i=j$ ，则 b_{ii} 表示结点 i 的出度。



邻接矩阵的运算

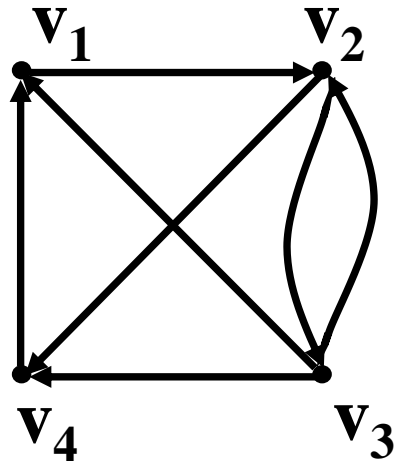
$$A^T \times A = C = [C_{ij}]$$

$$C_{ij} = \sum_{k=1}^n a_{ki} \times a_{kj} = a_{1i} \times a_{1j} + a_{2i} \times a_{2j} + \cdots + a_{ni} \times a_{nj}$$



- C_{ij} 表示同时有边指向结点*i*和结点*j*的那些结点的个数；
- 若*i=j*，则 C_{ii} 表示结点*i*的入度。

邻接矩阵的运算



$$A = \begin{Bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{Bmatrix}$$

$$A \times A^T = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 1 & 0 \\ 1 & 1 & 3 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

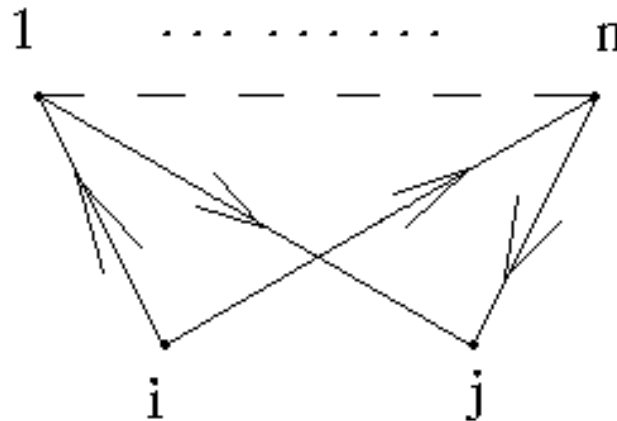
$$A^T \times A = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 2 \end{bmatrix}$$



邻接矩阵的运算

$$A \times A = A^2 = D = [d_{ij}]$$

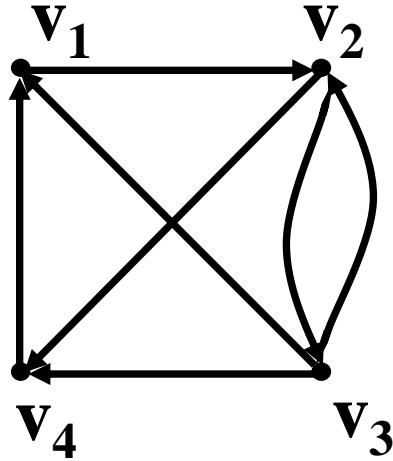
$$d_{ij} = \sum_{k=1}^n a_{ik} \times a_{kj} = a_{i1} \times a_{1j} + \dots + a_{in} \times a_{nj}$$



- 若 $a_{ik} \times a_{kj} = 1$ ，则表示有 $i \rightarrow k \rightarrow j$ 长度为2的有向边；
- d_{ij} 表示 i 和 j 之间具有长度为2的通路个数。



邻接矩阵的运算



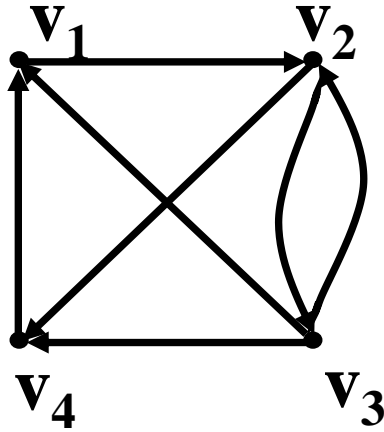
$$A = \begin{Bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{Bmatrix}$$

$$A^2 = A \times A = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$A^3 = A^2 \times A = \begin{bmatrix} 2 & 1 & 0 & 1 \\ 1 & 2 & 1 & 1 \\ 2 & 2 & 1 & 2 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

□ 从 $v_2 \rightarrow v_1$, 有二条长度为2的通路; 有一条长度为3的通路

邻接矩阵的运算



$$A = \begin{Bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{Bmatrix}$$

$$B_4 = A^1 + A^2 + A^3 + A^4 = \begin{bmatrix} 3 & 4 & 2 & 3 \\ 5 & 5 & 4 & 6 \\ 7 & 7 & 4 & 7 \\ 3 & 2 & 1 & 2 \end{bmatrix}$$

□ 长度不大于k的通路个数

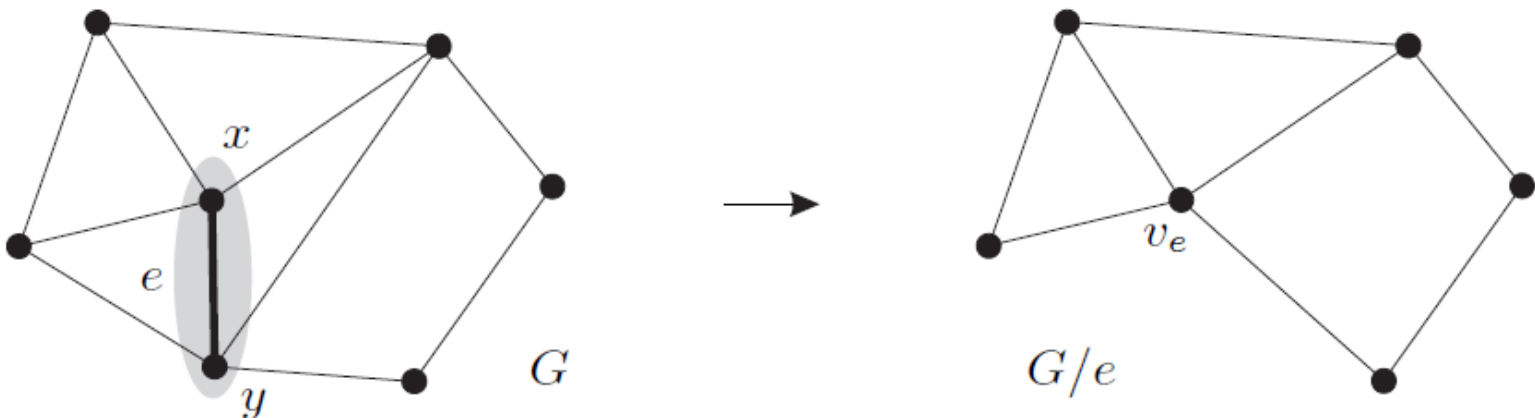
邻接矩阵的运算

- 回顾：[Warshall算法](#)



图的运算

- 加新边: $G+e$
- 减边或边集: $G-e$
- 减点或点集: $G-v$ (同时删除与 v 关联的边)
- 边的收缩: G/e





图的运算

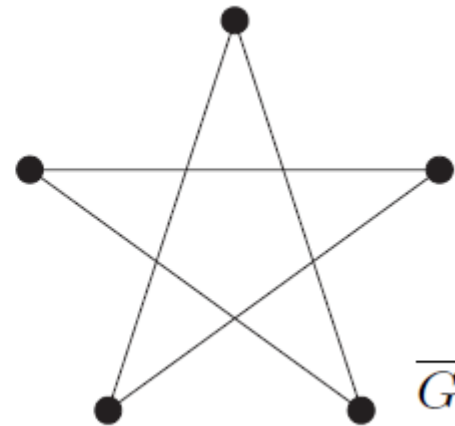
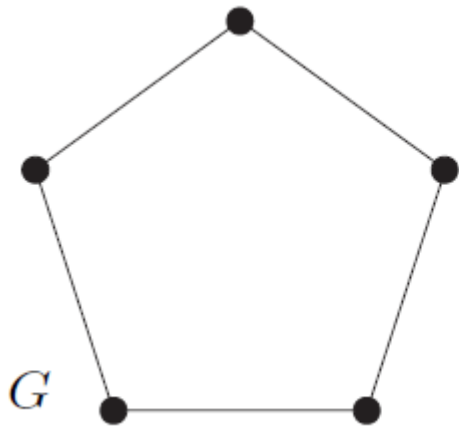
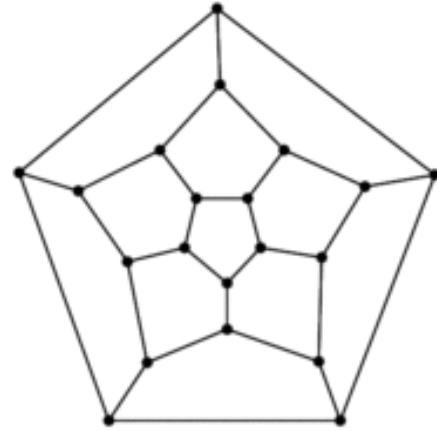
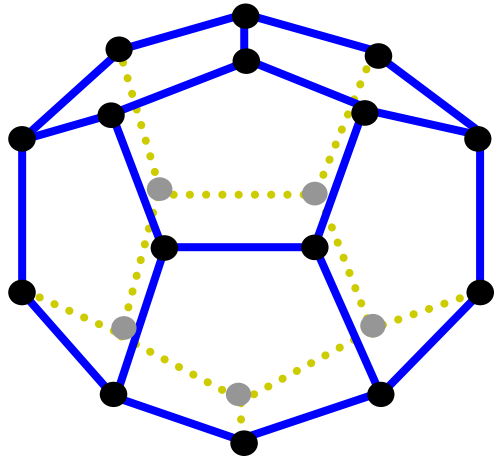
- $G \cup G'$: 以 $V(G) \cup V(G')$ 中的顶点组成的集合为顶点集, 以 $E(G) \cup E(G')$ 为边集。 // 简单图的并
- 假设 G 和 G' 是不交的无向图, 定义 $G * G'$ 如下:
 - 以 $V(G) \cup V(G')$ 为顶点集
 - 以 $E(G) \cup E(G') \cup \{\{x, y\} \mid x \in V(G), y \in V(G')\}$ 为边集
- 举例, $K_2 * K_3 = K_5$.
- 简单图 G 的补图 (complement graph), 记为 G
 - $G=(V, E)$ 的补图定义为 $(V, [V]^2 \setminus E)$



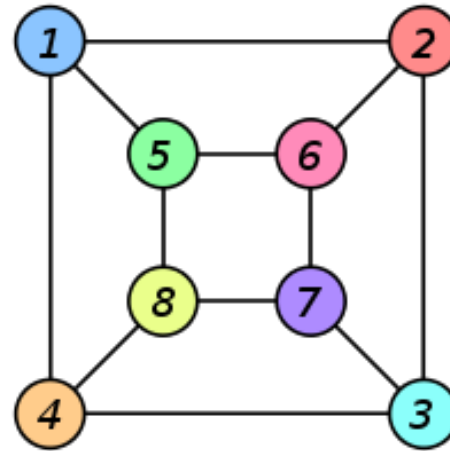
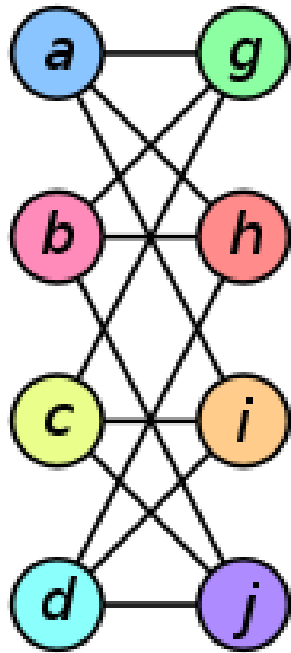
图的同构

- 图同构的定义
 - 设 $G_1=(V_1, E_1, \varphi_1)$ 和 $G_2=(V_2, E_2, \varphi_2)$ 是两个简单无向图。若存在双射 $f: V_1 \rightarrow V_2$, u 和 v 在 G_1 中相邻当且仅当 $f(u)$ 和 $f(v)$ 在 G_2 中相邻。此时称 f 是一个同构函数。
 - 设 $G_1=(V_1, E_1, \varphi_1)$ 和 $G_2=(V_2, E_2, \varphi_2)$ 是两个无向图。若存在双射 $f: V_1 \rightarrow V_2$, $g: E_1 \rightarrow E_2$, $\forall e \in E_1$, $\varphi_1(e)=\{u,v\}$, 当且仅当 $g(e) \in E_2$, 且 $\varphi_2(g(e))=\{f(u),f(v)\}$ 。

图同构的例子



图同构的例子



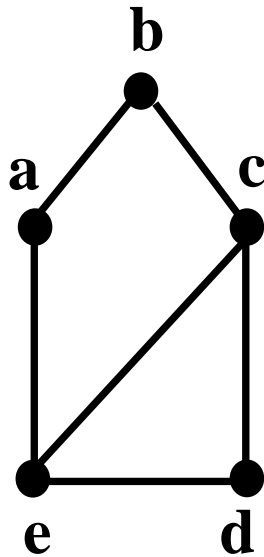


检测两个简单图是否同构

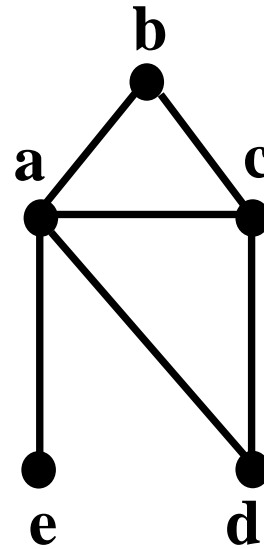
- 邻接矩阵表示: $n!$ 个
- 现有最好算法在最坏情况下的时间复杂性是指数级。
- (在最坏情况下) 时间复杂性为多项式的算法?

检测两个简单图是否同构

- 图同构下保持的性质称为图不变的
 - 顶点数、度序列、...
- 利用图不变的性质（**没有保持**）来推断出**不同构**

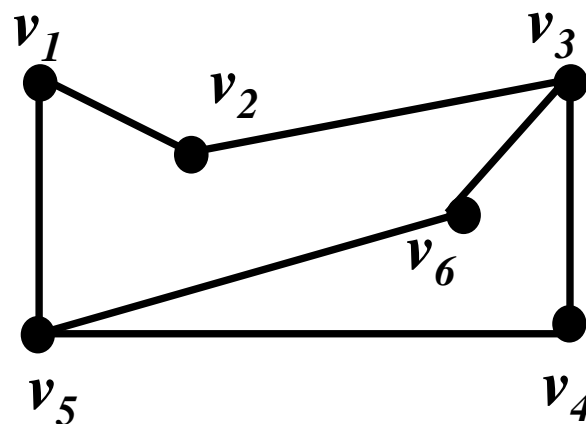
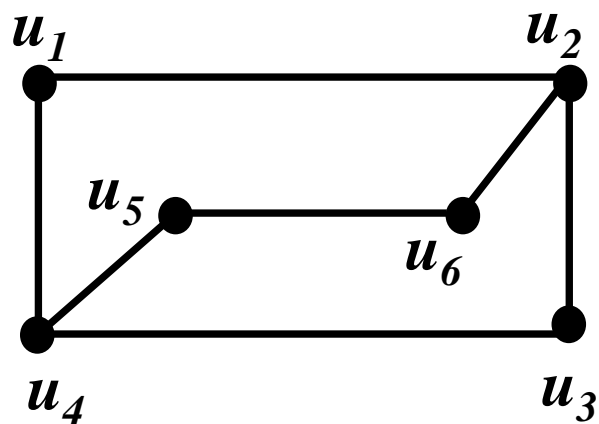
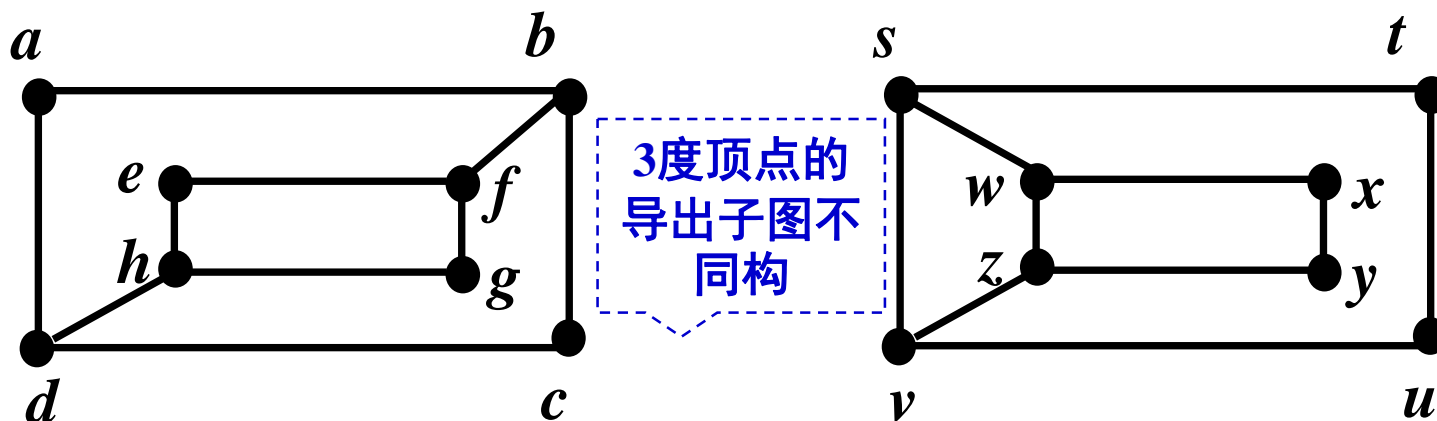


图G



图H

检测两个简单图是否同构



作业

- 见课程网站

