



第一章 基本概念

计算机系网络所：张小平



主要内容

- 离散数学概述
- 图的基本概念及定义
- 图的代数表示



离散数学概述

- 离散数学：
 - 离散数学：Discrete Mathematics
 - Separate , discontinuous, 分离的, 不连续的
 - 另一面：“连续、无限可分”
 - 物质世界到底是离散的，还是连续的？
 - 研究分立的对象之间所形成的关系
 - 以研究离散量的结构和相互之间的关系为主要目标，其研究对象一般是有限个或可数个元素



离散数学概述

- 离散数学主要包括以下四个方面：
 - 数理逻辑、集合论、图论、代数结构
 - 离散数学充分契合了计算机科学的特点，是计算机科学重要的基础理论之一



图论与代数结构



图论概述

- 图论 [Graph Theory] 是数学的一个分支，它以图为研究对象。
- 世界上各事物之间，自然界内诸现象之间经常存在着某些必然的联系，需要人们通过研究分析，去揭示这些关系。
- 人们常把事物、现象用**结点**表示，用有向的或无向的**边**来表示它们之间的联系。这就构成了图论中所讨论的**图**。



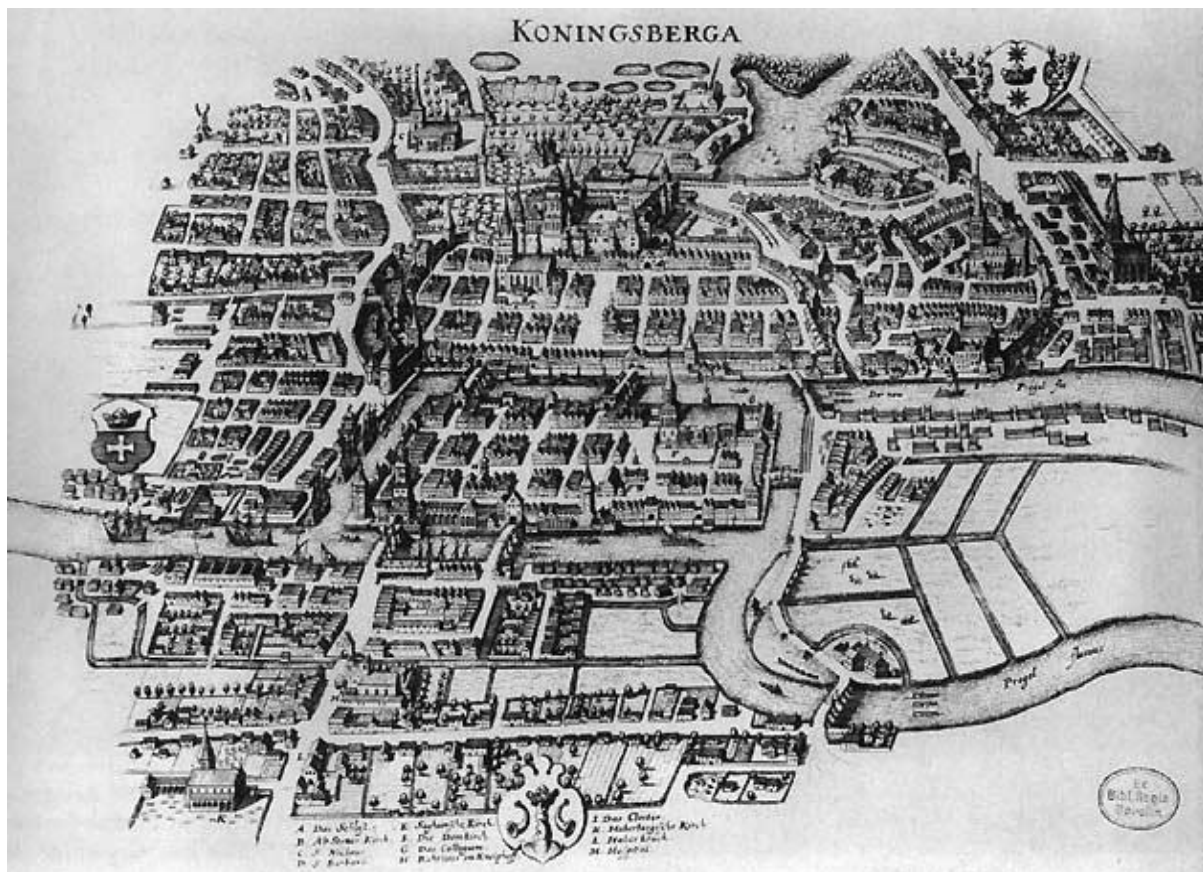
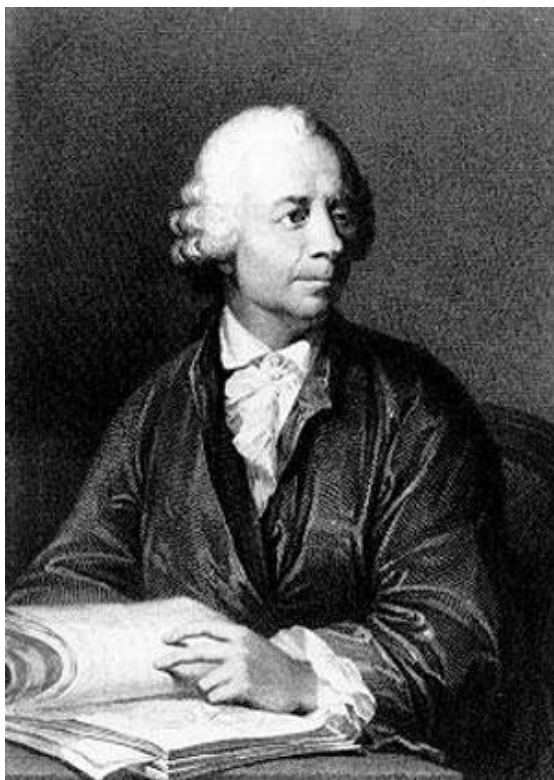
图论概述

- 历史上图论曾经被好多位数学家各自独立地建立过。关于图论的文字记载最早出现在欧拉1736年的论著中，其原始问题有很强的实际背景。
- 18世纪在哥尼斯堡城(今俄罗斯加里宁格勒)的普莱格尔河上有7座桥，将河中的两个岛和河岸连结。



图论概述

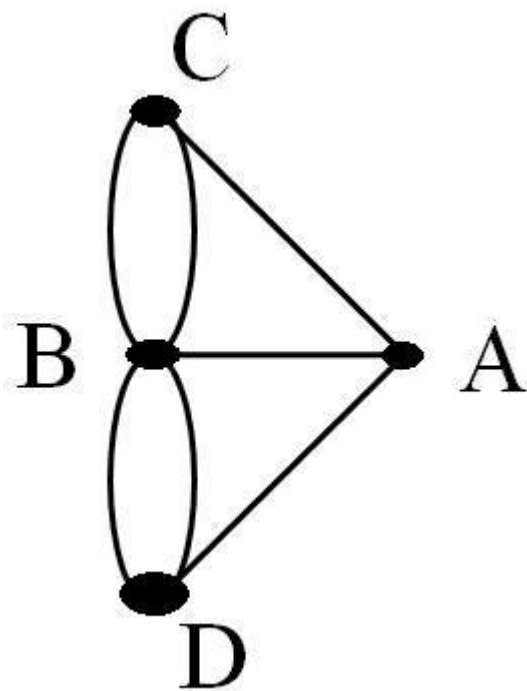
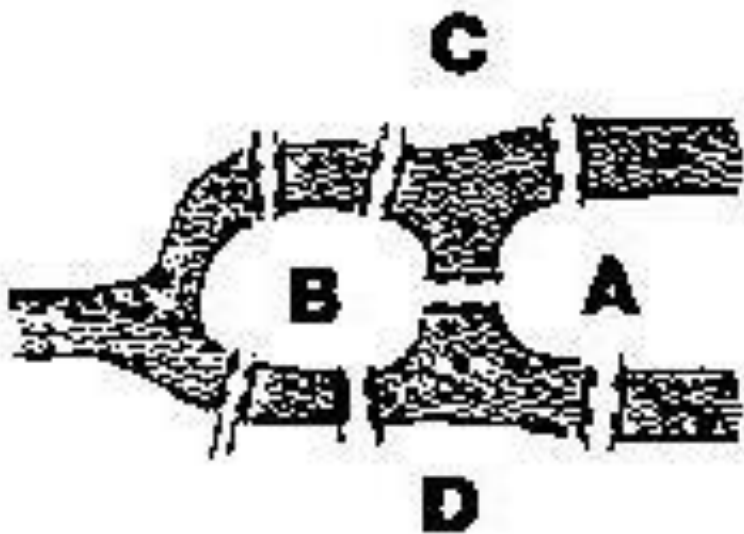
- 欧拉与哥尼斯堡城七桥问题





图论概述

- 欧拉对“七桥问题”的研究是图论研究的开始。





图论概述

- 早期的图论与数学游戏有密切的联系：
 - 周游世界问题、渡河问题、三家三井问题...
- 20世纪后，图论的应用渗透到其他学科领域，如物理、化学、运筹学、博弈论、计算机网络、社会学、语言学等等
- 对于基础图论来说，不要求事先掌握高深的数学工具，只需要有集合论和线性代数的基本概念，即可进行学习



图论概述

- 学习目标：
 - 掌握图论的基本概念、基本方法
 - 掌握图论的基本理论和重要定理、算法
 - 同时学习将实际问题转化为图论问题的能力

图论是一个非常有用的数学工具



图论概述

- 如何学好图论：
 - 注意图论中解题或证明的方法：与微积分不同，反证法、构造法是图论解题的主要方法。
 - 从基本概念入手充分掌握定义、定理，并重视定理的证明过程。
 - 认真听课！
 - 保质保量完成作业！



主要内容

- 图论概述
- 图的基本概念及定义
- 图的代数表示



图的基本概念及定义

- 定义1.1.1 二元组 $G=(V(G),E(G))$ 称为图。其中 $V(G)$ 是非空集，称为结点集； $E(G)$ 为 $V(G)$ 各结点之间边的集合，称为边集。
- 常用 $G=(V,E)$ 表示图。
- 当 V,E 都是有限集合时，称 G 为有限图。
- 当 V 或 E 是无限集合时，称 G 为无限图。
- 一般情况下，给定 $G=(V,E)$ ，如不加特殊说明，认为 $V=\{v_1,v_2,v_3,\cdots,v_n\}$ ， $E=\{e_1,e_2,e_3,\cdots,e_m\}$ ，即结点数 $|V|=n$ ，边数 $|E|=m$ 。



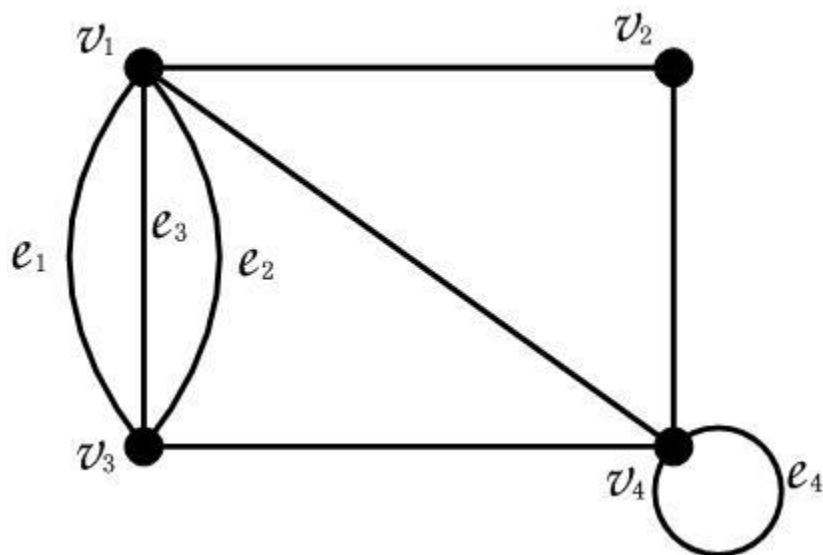
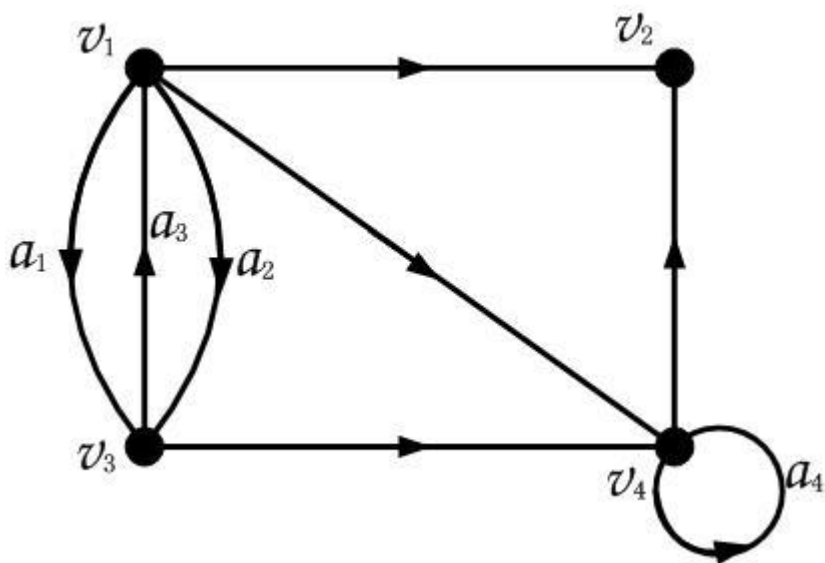
图的基本概念及定义

- 有向图和无向图：

- 若图中的边为有向的，则称为**有向图**

- 若图中的边为无向的，则称为**无向图**

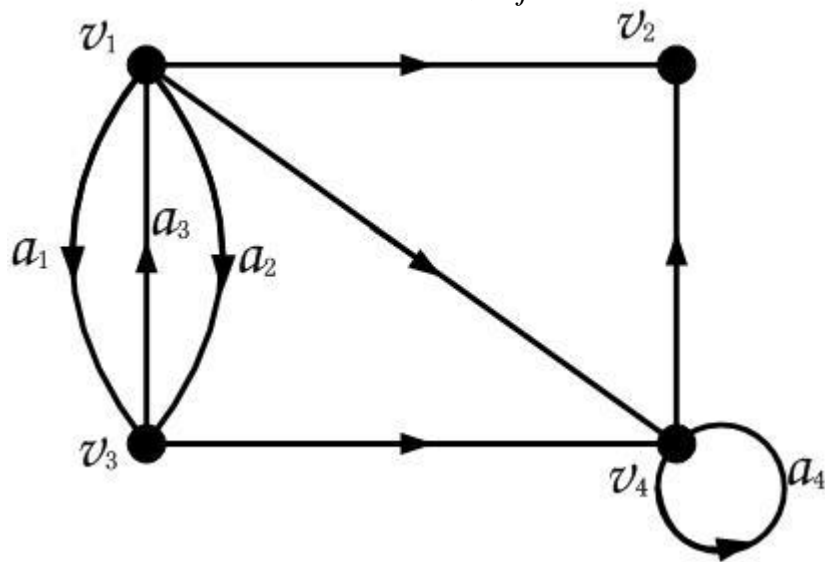
- 若图中既有有向边又有无向边，则称为**混合图**





图的基本概念及定义

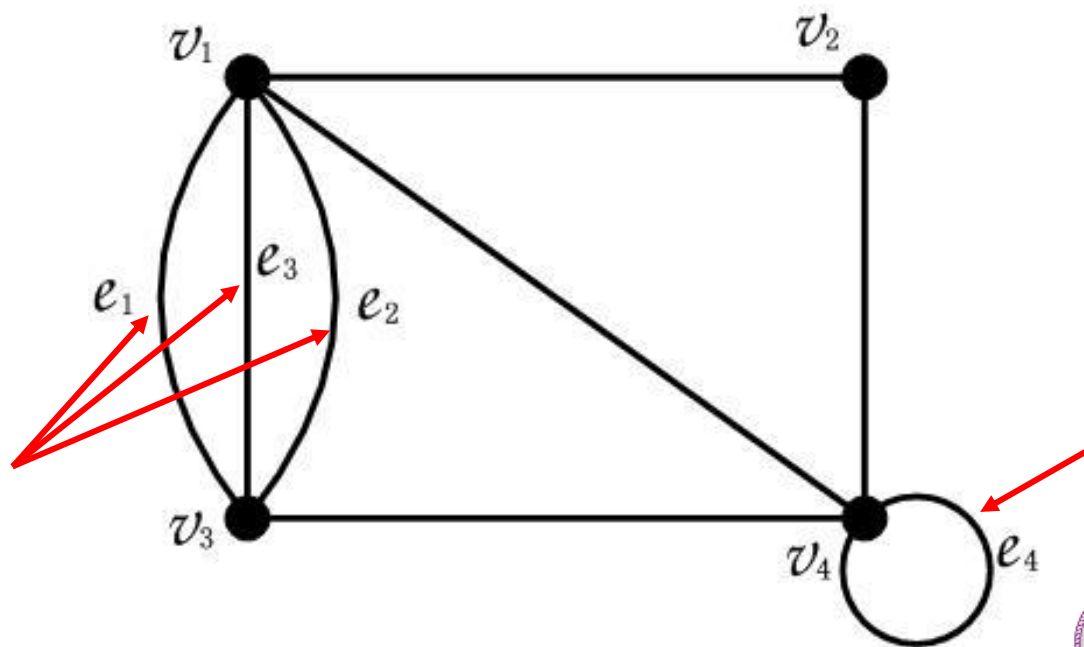
- 图的边可用 $e_k = (v_i, v_j)$ 表示
 - 称 v_i 与 v_j 是**相邻结点**
 - 称 e_k 分别与 v_i, v_j **相关联**
 - 如果 e_k 是有向边, 称 v_i 是 e_k 的**始点**, v_j 是 e_k 的**终点**, 并称 v_i 是 v_j 的**直接前驱**, v_j 是 v_i 的**直接后继**
 - 如果 e_k 是无向边, 称 v_i, v_j 是 e_k 的两个**端点**





图的基本概念及定义

- 只与一个结点相关联的边称为**自环**
- 在同一对结点之间可以存在多条边，称为**重边**
- 含有重边的图叫**多重图**





图的基本概念及定义

- 定义1.1.2 $G = (V, E)$ 的某结点所关联的边数称为该结点的度，用 $d(v)$ 表示。如果 v 带有自环，则自环对 $d(v)$ 的贡献为2。
- 有向图中：
 - 以结点 v 为始点的边数目称为 v 的正度，记为 $d^+(v)$
 - 以结点 v 为终点的边数目称为 v 的负度，记为 $d^-(v)$
- 显然，有 $d^+(v) + d^-(v) = d(v)$



图的基本概念及定义

- 定义1.1.3 任意两结点间最多只有一条边，且不存在自环的无向图称为**简单图**。
- 没有任何边的简单图叫**空图**，记为 N_n 。
- 任何两结点间都有边的简单图称为**完全图**，记为 K_n 。
 - K_n 中每个结点的度为 $n-1$



图的基本概念及定义

- 性质 1.1.1 设 $G = (V, E)$ 有 n 个结点， m 条边，
则

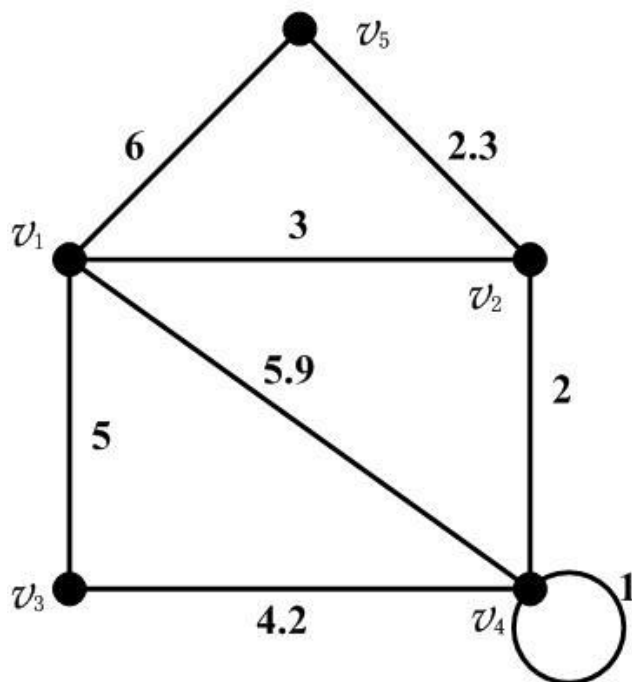
$$\sum_{v \in V(G)} d(v) = 2m$$

- 性质 1.1.2 G 中度为奇数的结点必为偶数个
- 性质 1.1.3 有向图 G 中正度之和等于负度之和
- 性质 1.1.4 K_n 中的边数为 $\frac{1}{2}n(n-1)$
- 性质 1.1.5 非空简单图中一定存在度相同的
结点



图的基本概念及定义

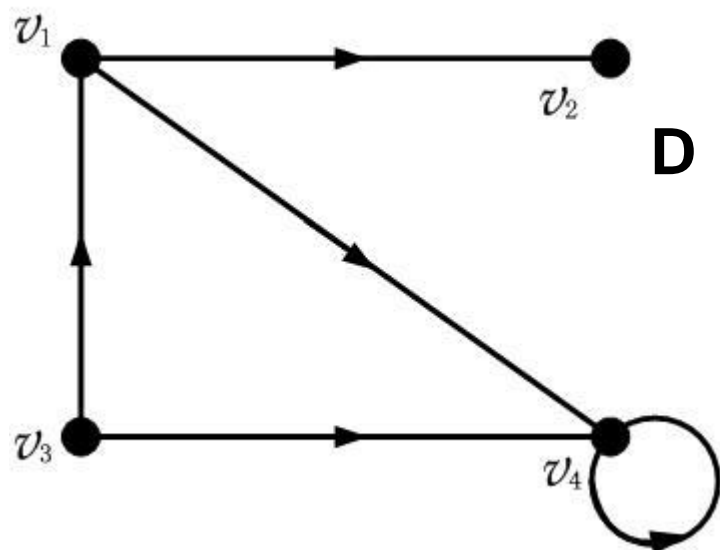
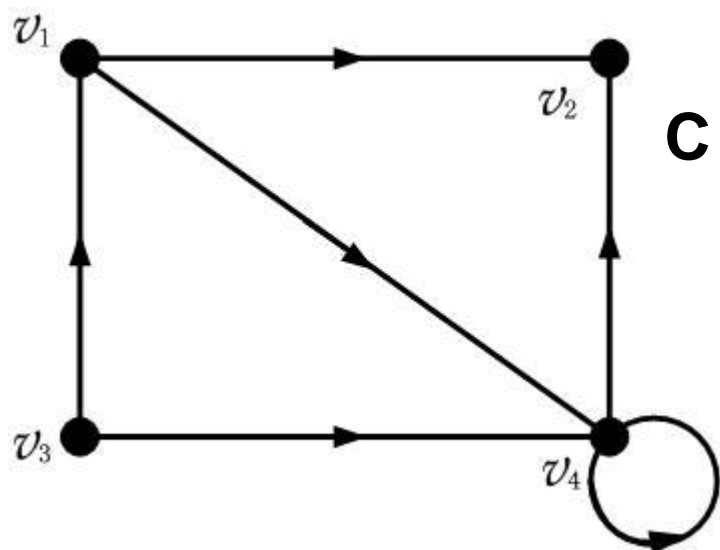
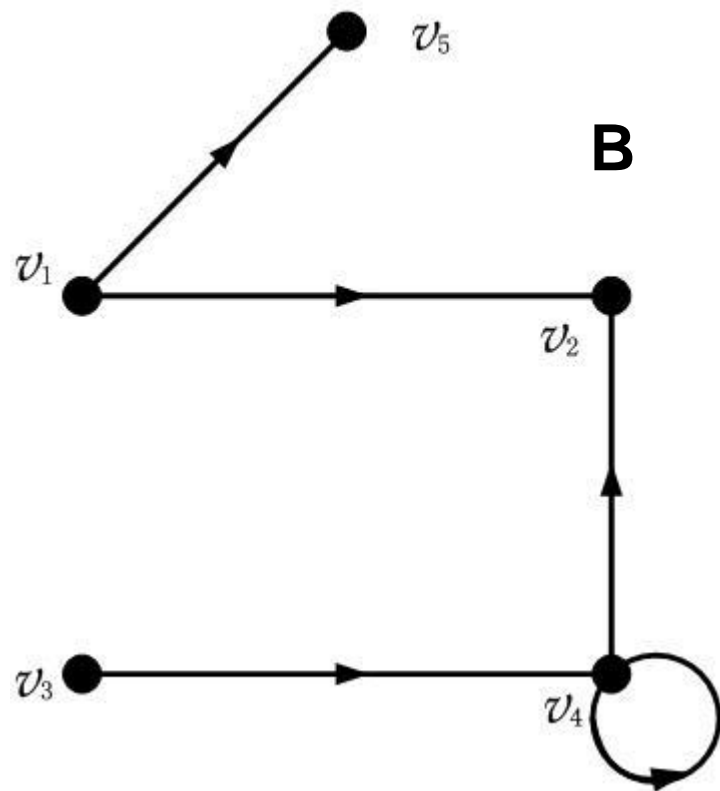
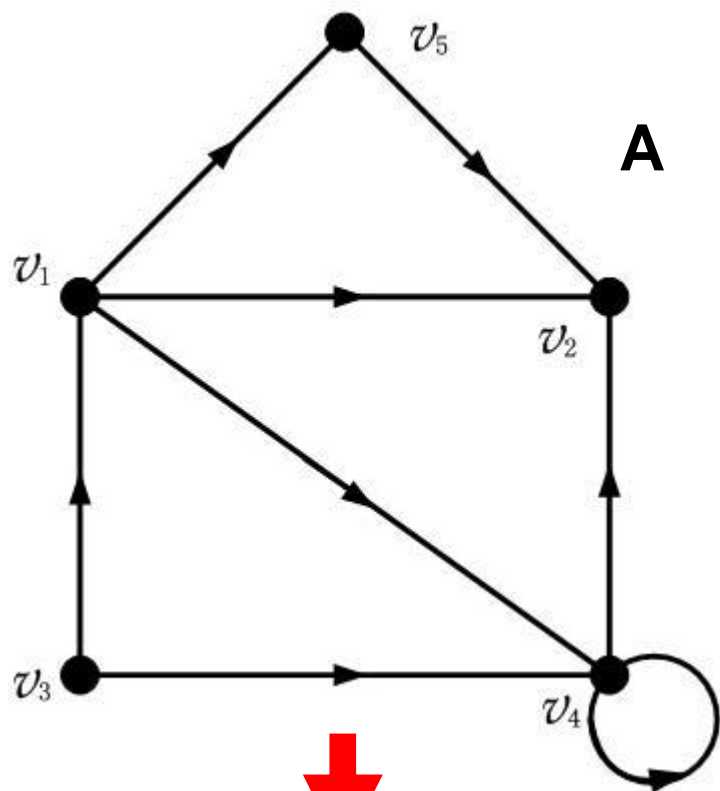
- 定义1.1.4 如果图 $G=(V,E)$ 的每条边 $e_k=(v_i, v_j)$ 都赋予一个实数 w_k 做为该边的权，则称 G 为赋权图。如果这些权都是正实数，就称 G 为正权图。





图的基本概念及定义

- 定义1.1.5 给定 $G=(V,E)$, 如果存在另外一个图 $G'=(V',E')$, 满足 V' 包含于 V , E' 包含于 E , 则称 G' 是 G 的一个子图
- 如果 $V'=V$, 就称 G' 是 G 的支撑子图
- 如果 V' 包含于 V , 且 E' 包含了在结点子集 V' 之间的所有边, 则称 G' 是 G 的导出子图





图的基本概念及定义

- 显然，根据上述定义，图 G 是自身的子图，支撑子图，导出子图
- 空图是图 G 的支撑子图
- 称原图 G 和空图都是图 G 的平凡子图



图的基本概念及定义

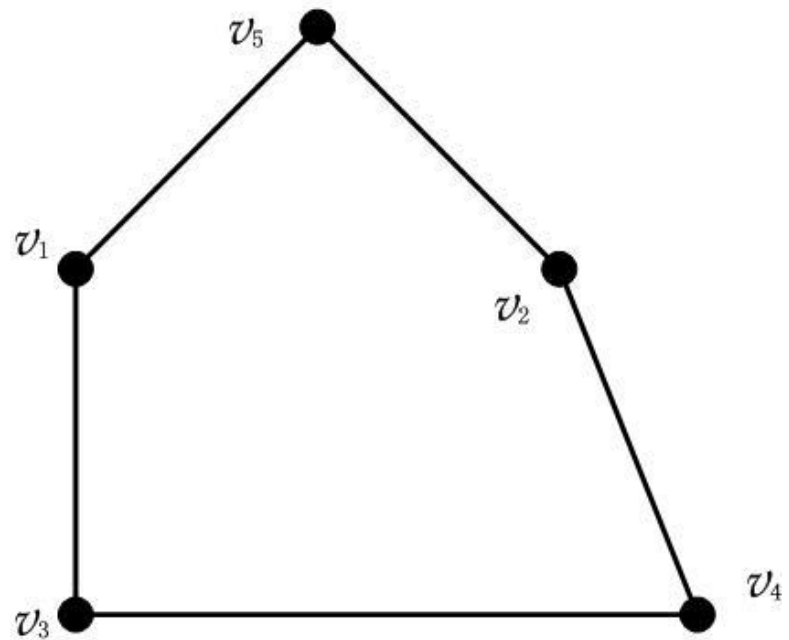
- 定义1.1.6 给定两个图 $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ 。
令

$$G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$$

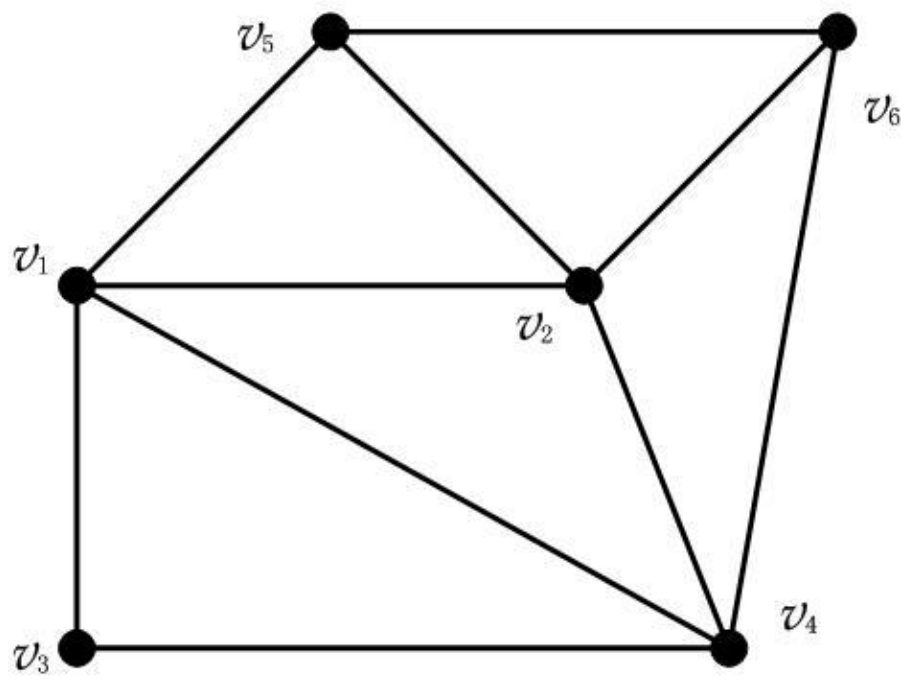
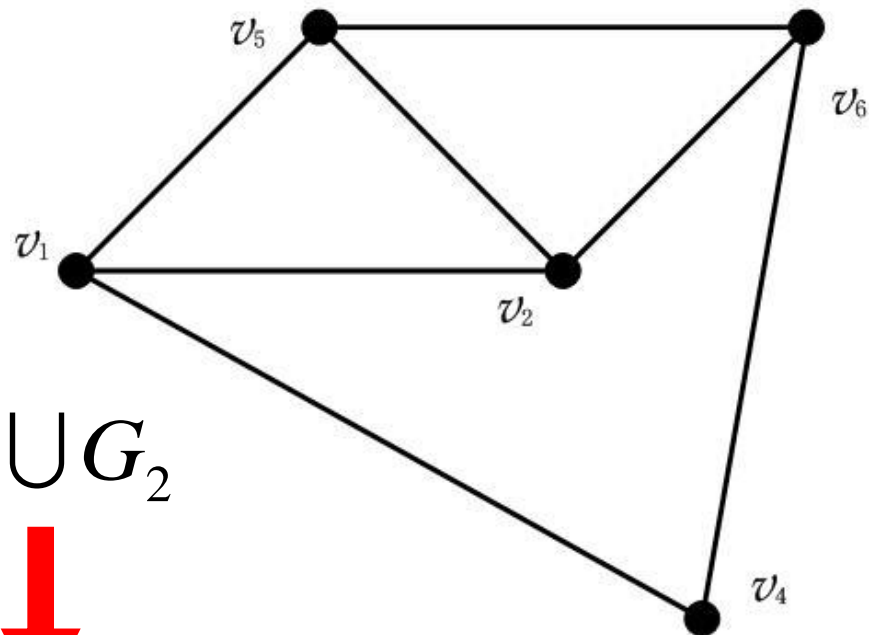
$$G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$$

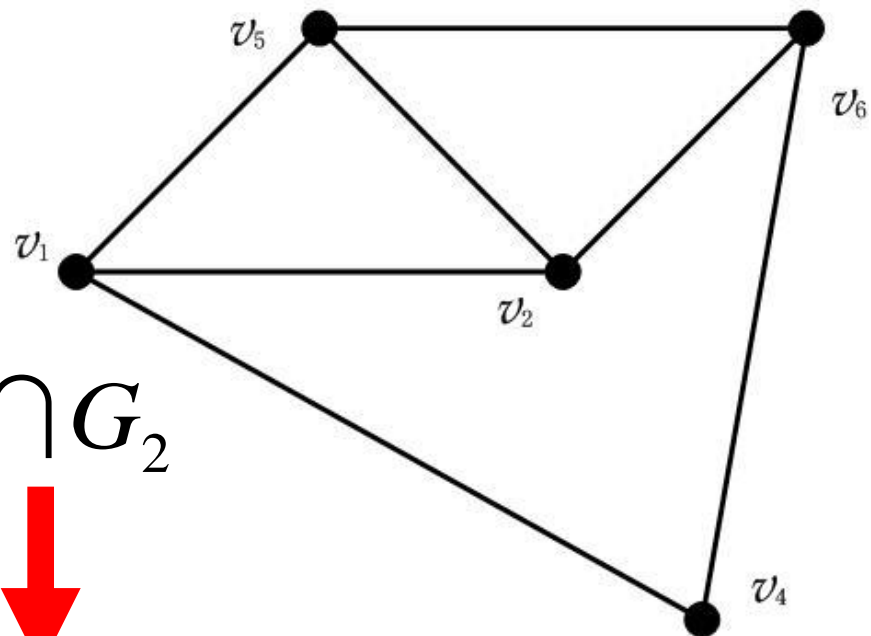
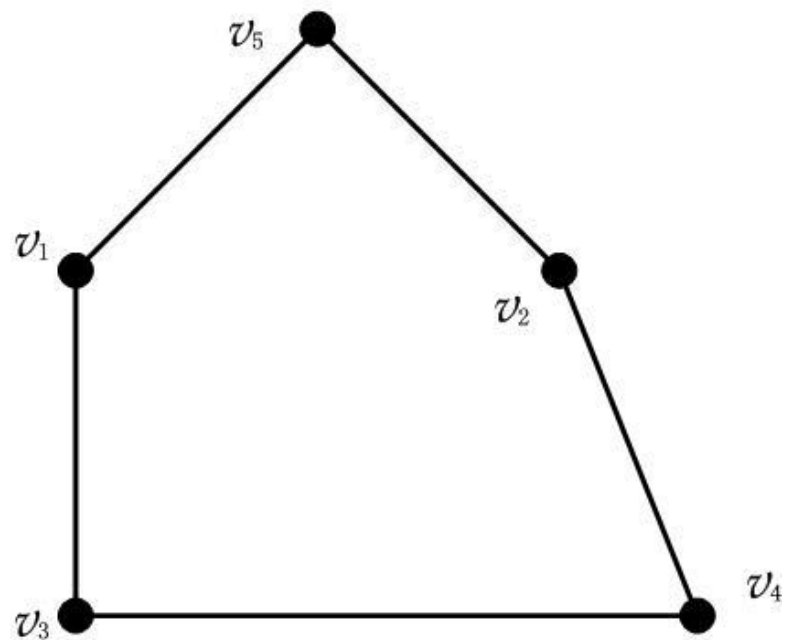
$$G_1 \oplus G_2 = (V_1 \cup V_2, E_1 \oplus E_2)$$

分别称为 G_1 和 G_2 的 并、交、对称差 去同存异

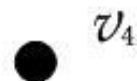
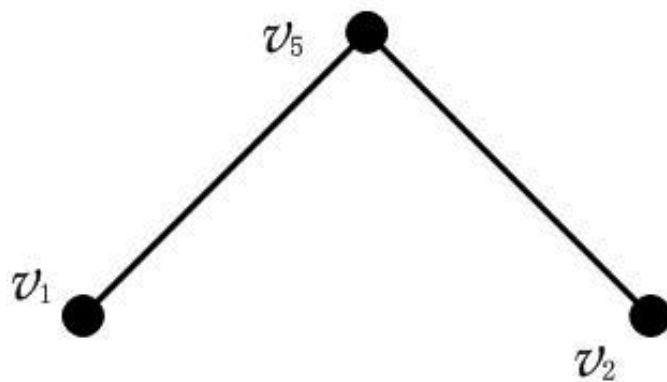


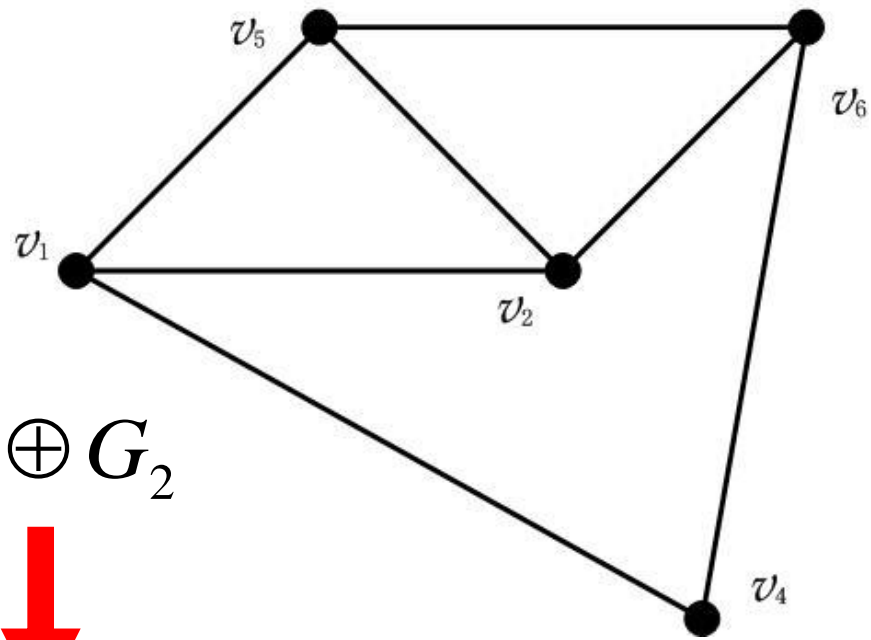
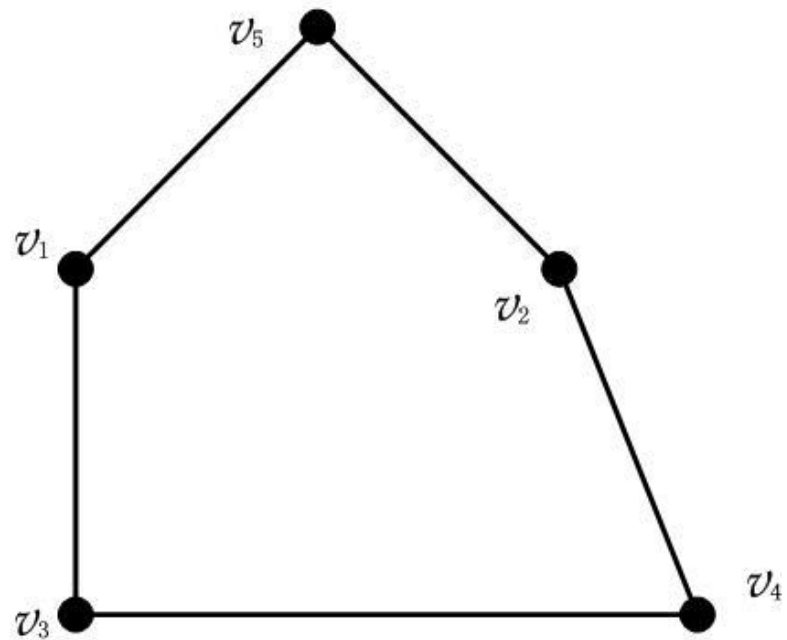
$$G_1 \cup G_2$$



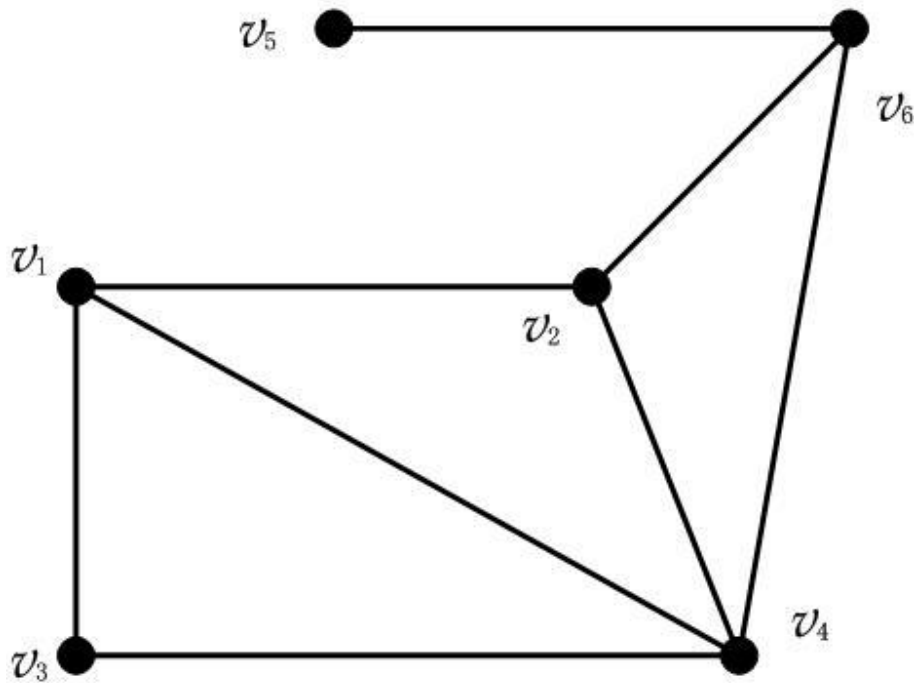


$$G_1 \cap G_2$$





$$G_1 \oplus G_2$$





图的基本概念及定义

- 在图 G 中删掉一个子图 H ，指删掉 H 中的各条边，记做 $G - H$ 。
- G 为简单图，称 $K_n - G$ 为 G 的补图，记做 \overline{G}
- 从 G 中删去某个结点 v 及其关联的边所得到的图记做 $G - v$
- 从 G 中删掉某条特定的边 e ，记做 $G - e$
- 显然， $G - v$ 是图 G 的导出子图， $G - e$ 是图 G 的支撑子图



图的基本概念及定义

- 定义1.1.7 如 G 为无向图，则

$\Gamma(v) = \{u \mid (v, u) \in E\}$ 称为 v 的邻点集

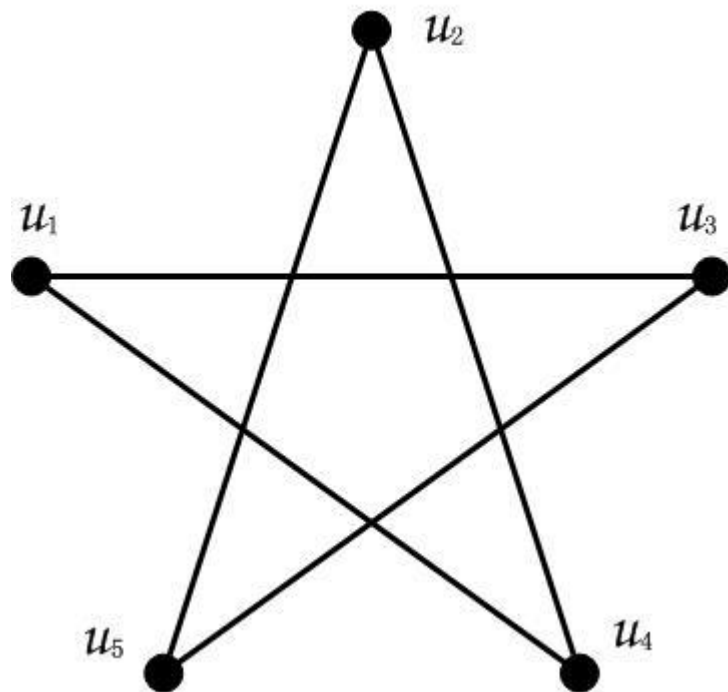
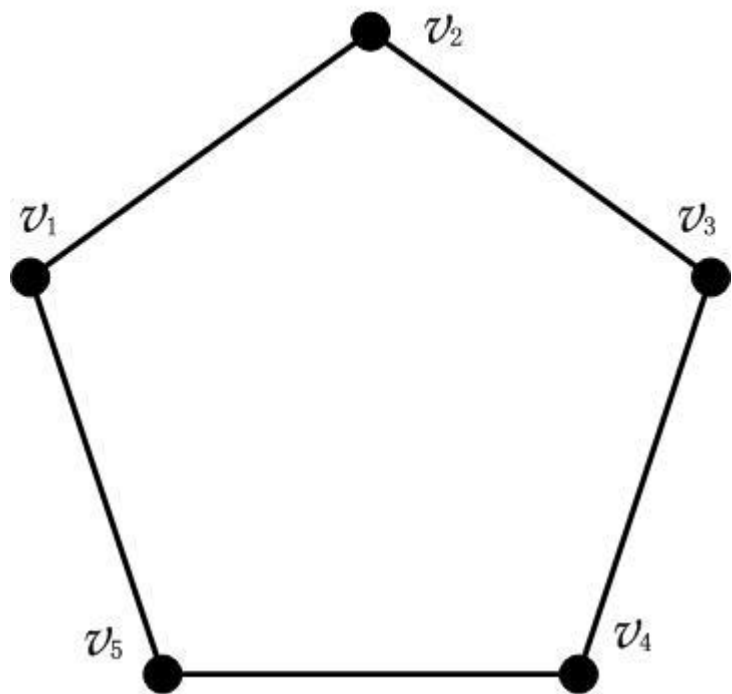
如 G 为有向图， v 是其一个结点，则

$$\Gamma^+(v) = \{u \mid (v, u) \in E\}$$

称为 v 的直接后继集或外邻集；

$$\Gamma^-(v) = \{u \mid (u, v) \in E\}$$

称为 v 的直接前趋集或内邻集





图的基本概念及定义

- 定义1.1.8 两个图 $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, 如果 V_1, V_2 之间存在双射 f , 而且 $(u, v) \in E_1$, 当且仅当 $(f(u), f(v)) \in E_2$ 时, 称 G_1 和 G_2 同构。

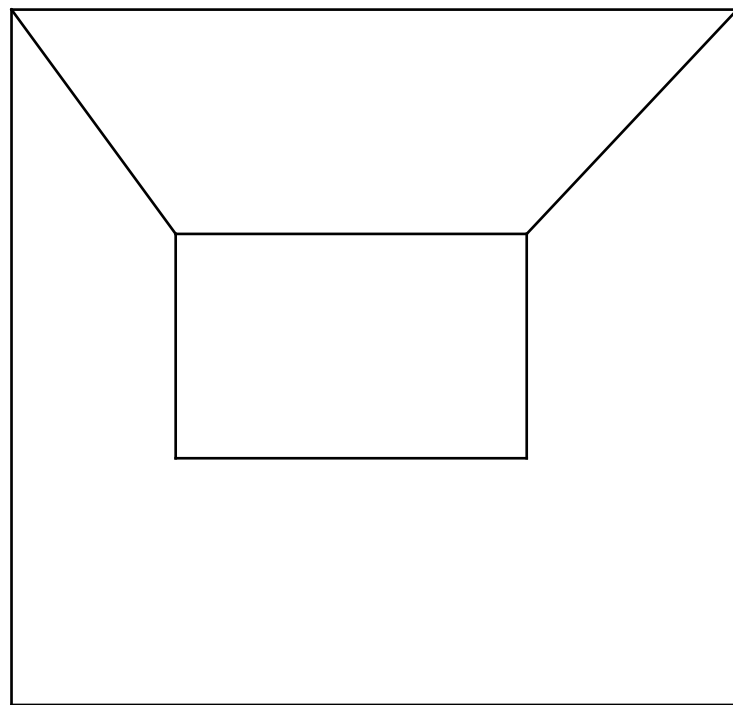
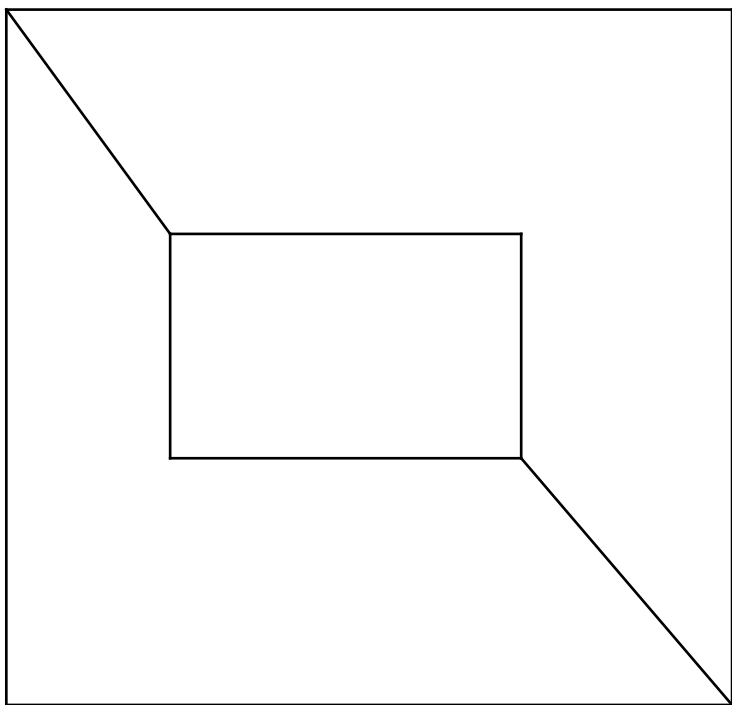
记做

$$G_1 \cong G_2$$



图的基本概念及定义

- 从定义知，若 $G_1 \cong G_2$ ，必须满足：
 - (1) $|V(G_1)| = |V(G_2)|, |E(G_1)| = |E(G_2)|$
 - (2) G_1 和 G_2 结点度的非增序列相同
 - (3) G_1 和 G_2 存在同构的导出子图
- 如何判定两图同构？





图的基本概念及定义

- 小结:

- 图、有限图、无限图
- 有向图、无向图、混合图
- 相邻结点、始点、终点、直接前驱、直接后继、端点
- 自环、重边、多重图
- 结点度、正度、负度
- 简单图、空图、完全图
- 权、赋权图、正权图
- 子图、支撑子图、导出子图、平凡子图
- 图的交、并、对称差、补图
- 图的同构



主要内容

- 图论概述
- 图的基本概念及定义
- 图的代数表示

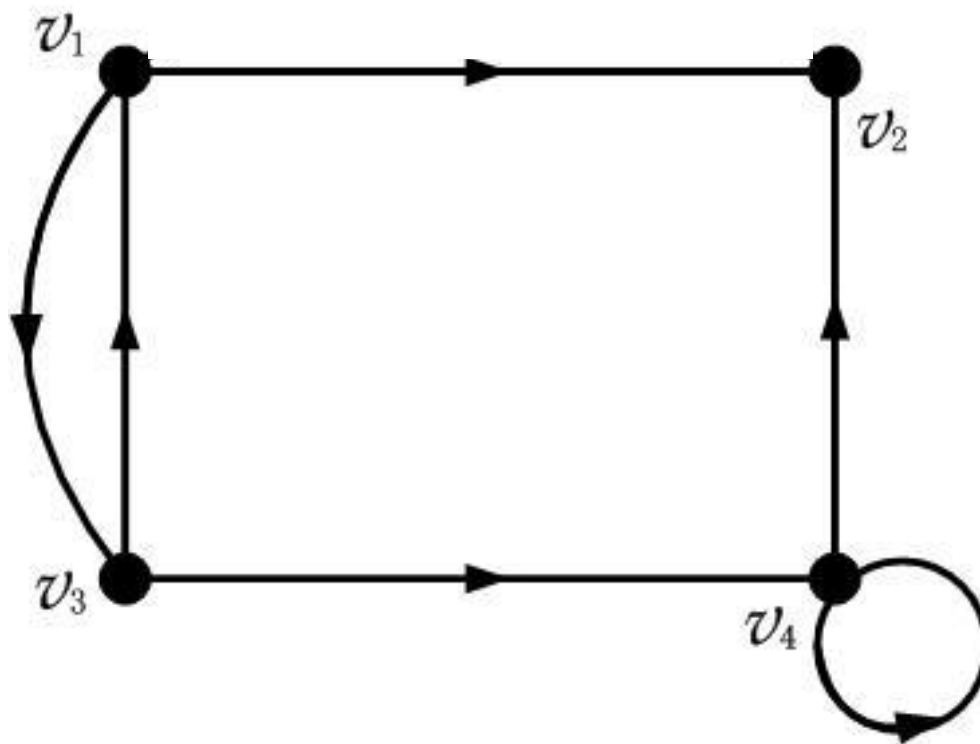


图的代数表示

- 如何对图进行描述或运算？
- 我们需要用代数的方法来表示图！



图的代数表示



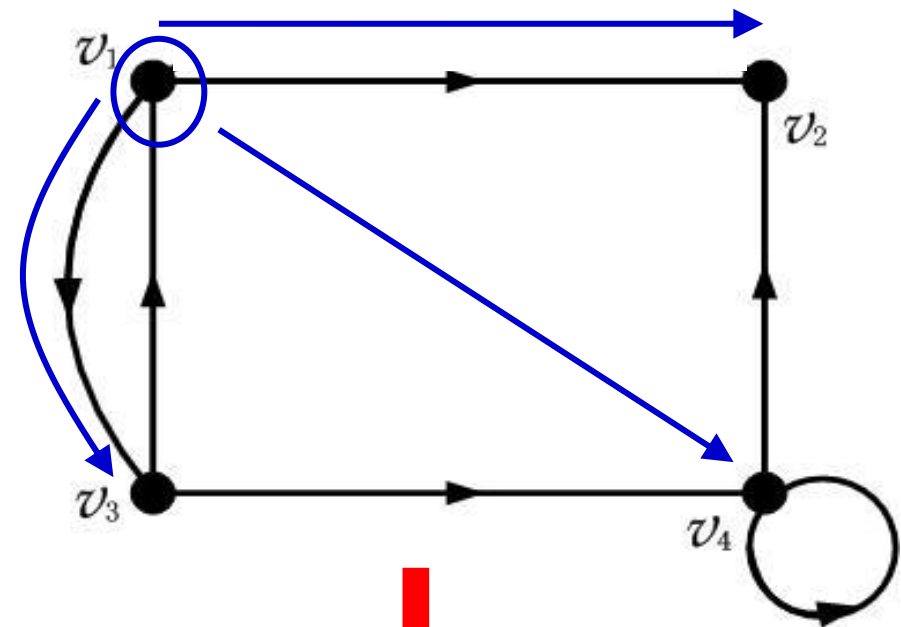


图的代数表示

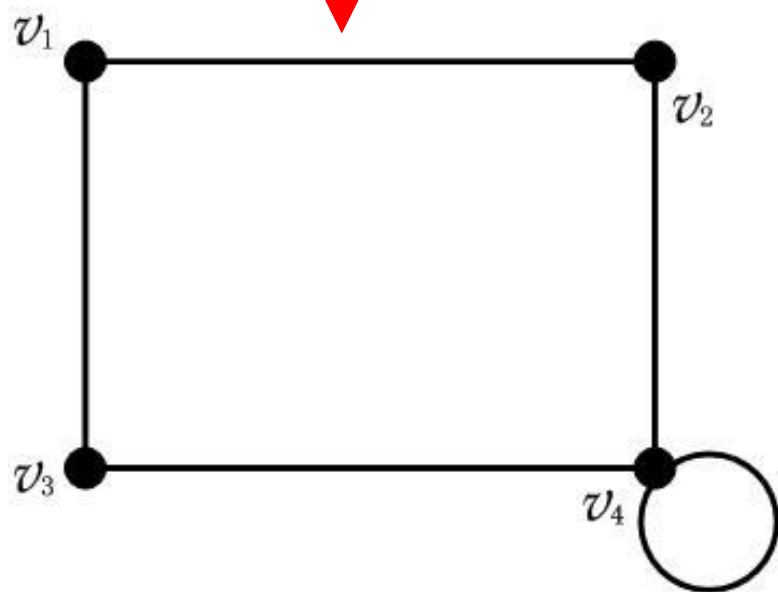
- **邻接矩阵**：表示了结点间的邻接关系
 - 有向图的邻接矩阵A是一个n阶方阵，其元素为：

$$A = [a_{ij}]_{n \times n} \quad a_{ij} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & \text{其它} \end{cases}$$

邻接矩阵可以表示自环，但无法表示重边



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



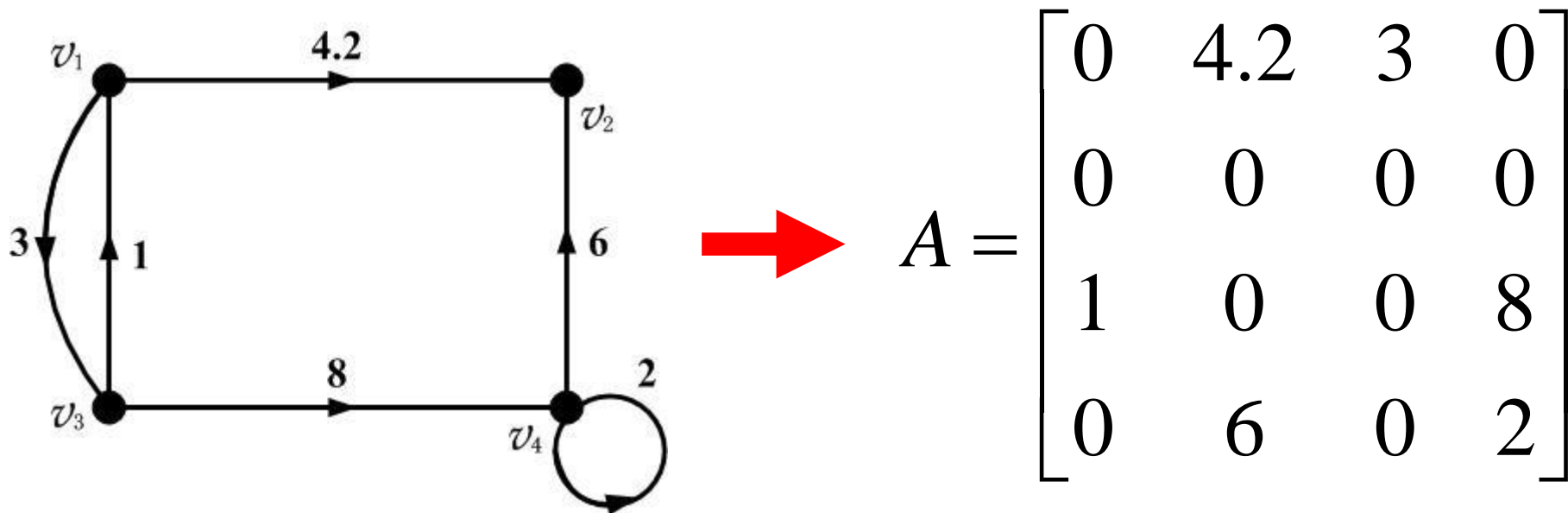
$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$



图的代数表示

- **权矩阵**：赋权图常用权矩阵 A 进行表示，其元素为：

$$A = [a_{ij}]_{n \times n} \quad a_{ij} = \begin{cases} w_{ij} & (v_i, v_j) \in E \\ 0 & \text{其它} \end{cases}$$



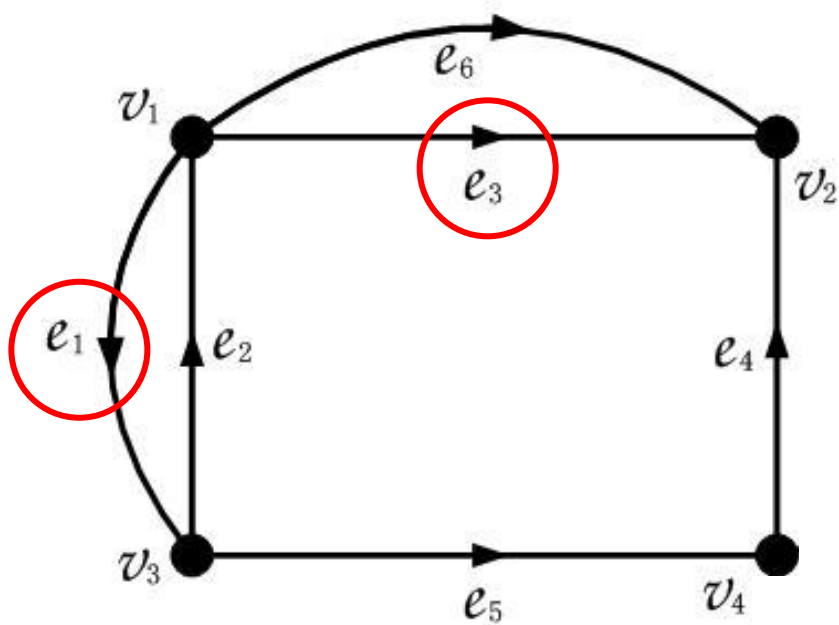
同样，权矩阵可以表示自环，但无法表示重边



图的代数表示

- **关联矩阵**：关联矩阵表示结点与边之间的关联关系。
 - 有向图 G 的关联矩阵 B 是 $n \times m$ 的矩阵，当给定结点和边的编号之后，其元素

$$B = [b_{ij}]_{n \times m} \quad b_{ij} = \begin{cases} 1 & e_j = (v_i, v_k) \in E \\ -1 & e_j = (v_k, v_i) \in E \\ 0 & \text{其它} \end{cases}$$



→ $B = \begin{bmatrix} v_1 & 1 & -1 & 1 & 0 & 0 & 1 & 0 \\ v_2 & 0 & 0 & -1 & -1 & 0 & -1 & 0 \\ v_3 & -1 & 1 & 0 & 0 & 1 & 0 & 0 \\ v_4 & 0 & 0 & 0 & 1 & -1 & 0 & ? \\ & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{bmatrix}$

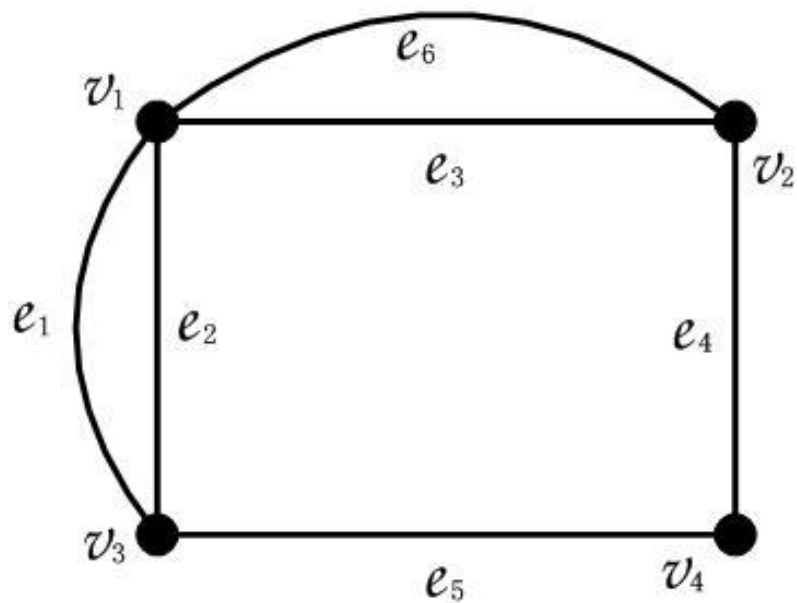
关联矩阵可以表示重边，但无法表示自环



图的代数表示

- 关联矩阵：关联矩阵表示结点与边之间的关联关系。
 - 无向图 G 的关联矩阵 B 是 $n \times m$ 的矩阵，当给定结点和边的编号之后，其元素

$$B = [b_{ij}]_{n \times m} \quad b_{ij} = \begin{cases} 1 & e_j \text{ 与 } v_i \text{ 关联} \\ 0 & \text{其它} \end{cases}$$



$$B = \begin{bmatrix} v_1 & 1 & 1 & 1 & 0 & 0 & 1 \\ v_2 & 0 & 0 & 1 & 1 & 0 & 1 \\ v_3 & 1 & 1 & 0 & 0 & 1 & 0 \\ v_4 & 0 & 0 & 0 & 1 & 1 & 0 \\ & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{bmatrix}$$



图的代数表示

- 关联矩阵的性质（有向图）：
 - 每列只有两个非零元：1、-1
 - 第 i 行非零元的数目恰为结点 V_i 的度，其中1的数目为其正度，-1的数目为其负度
- 关联矩阵的性质（无向图）：
 - 每列只有一个非零元：1
 - 第 i 行1的数目恰为结点 V_i 的度
- 能够表示重边，但不能表示自环



图的代数表示

- 优点：
 - 邻接矩阵、权矩阵和关联矩阵一旦写出代数表达式，则可得到确定图，且非常直观
- 缺点：
 - 不能表示重边或自环
 - 在计算机上存储邻接矩阵与关联矩阵时，将占据较大的存储空间并可能增加计算复杂度。
- 因此引入边列表、正向表、逆向表、邻接表等。



图的代数表示

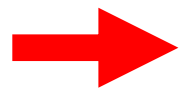
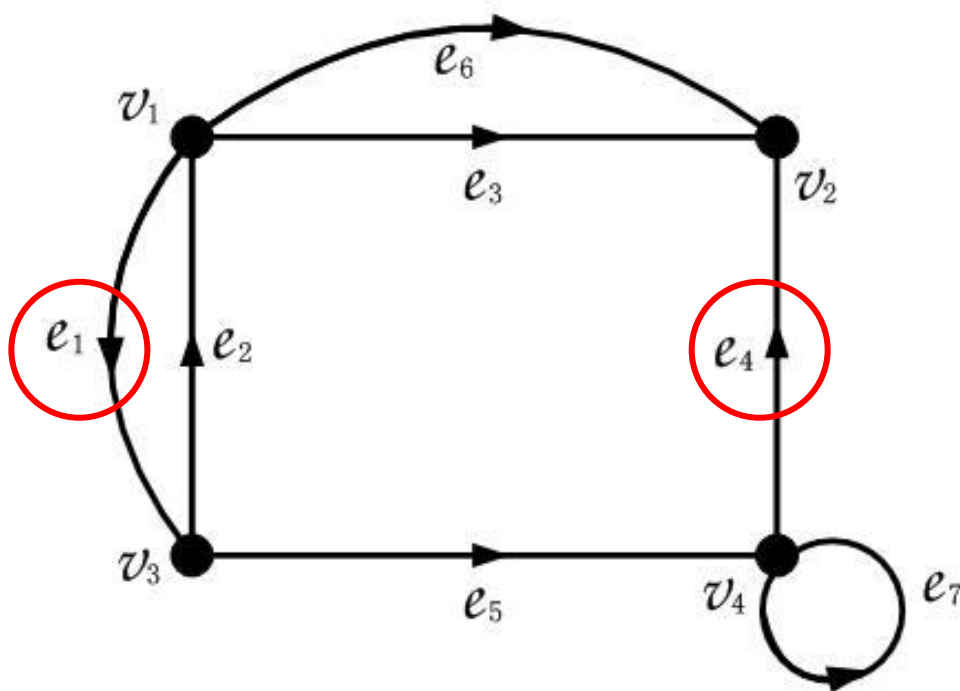
• 边列表

由两个 m 维向量 A 和 B 组成。

若 $e_k = (v_i, v_j)$, 则 $A(k) = i$, $B(k) = j$

如 G 为赋权图, 则再增加一个 m 维变量 Z ,

若 e_k 的权值为 w_k , 则令 $Z(k) = w_k$



$$A: (1 \quad 3 \quad 1 \quad 4 \quad 3 \quad 1 \quad 4)$$

$$B: (3 \quad 1 \quad 2 \quad 2 \quad 4 \quad 2 \quad 4)$$

$e_1 \quad e_2 \quad e_3 \quad e_4 \quad e_5 \quad e_6 \quad e_7$

边列表实质是关联矩阵的压缩形式并克服了其缺点



图的代数表示

- 赋权图只需增加权值向量 Z 即可，不详述
- 类似，可以得到无向图的边列表表示

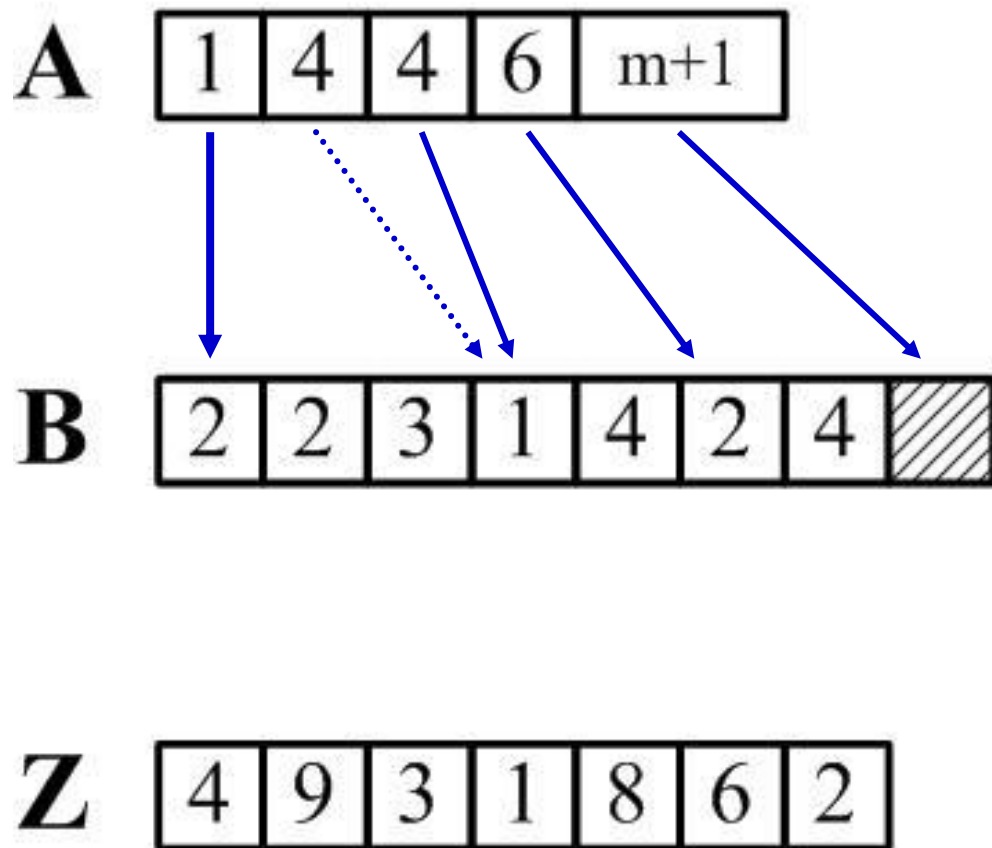
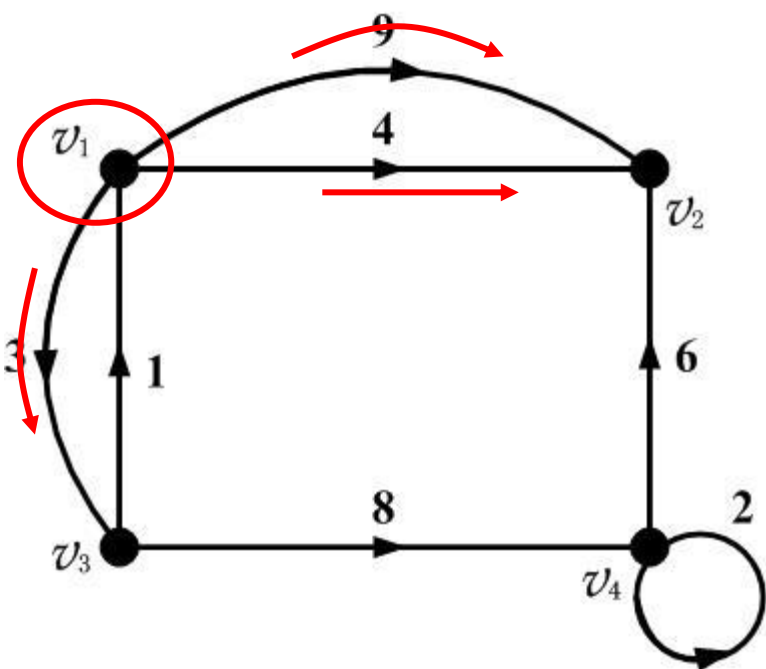


图的代数表示

- **正向表**：当对 G 的结点与边进行编号后，正向表将每个节点的直接后继集中在一起存放。
- 有向图的正向表由一个 $(n+1)$ 维向量 A ，一个 m 维向量 B 组成。 $A(i)$ 表示结点 v_i 的第一个后继在 B 中的地址， B 中存放这些后继结点的编号

$$A(n+1) = m+1$$

- 对赋权图，用 m 维向量 Z 存放权 $Z(k) = w_k$





图的代数表示

- 正向表存在下述关系：

- 1. $d^+(v_i) = A(i+1) - A(i)$

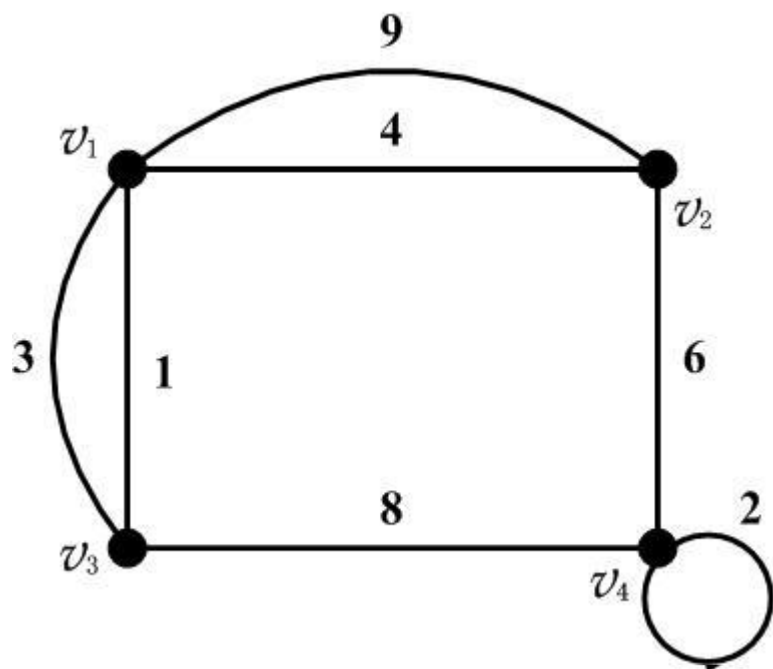
- 2. $A(i) = \sum_{j=1}^{i-1} d^+(v_j) + 1$

- 3. 从 $B(A(i))$ 到 $B(A(i+1)-1)$ 的任意一个值，都是 v_i 的直接后继



图的代数表示

- 无向图的正向表结构：
- B 向量中存放的是相应邻结点的编号，因此为 $2m$ 维
- 相应地， Z 向量也变为 $2m$ 维



A

1	4	4	6	m+1
---	---	---	---	-----

B

2	2	3	1	4	2	4	
---	---	---	---	---	---	---	--

Z

4	9	3	1	8	6	2
---	---	---	---	---	---	---

A

1	5	8	11	m+1
---	---	---	----	-----

B

2	2	3	3	1	1	4	1	1	4	2	3	4	4	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Z

4	9	3	1	4	6	9	1	8	3	2	8	6	2
---	---	---	---	---	---	---	---	---	---	---	---	---	---



图的代数表示

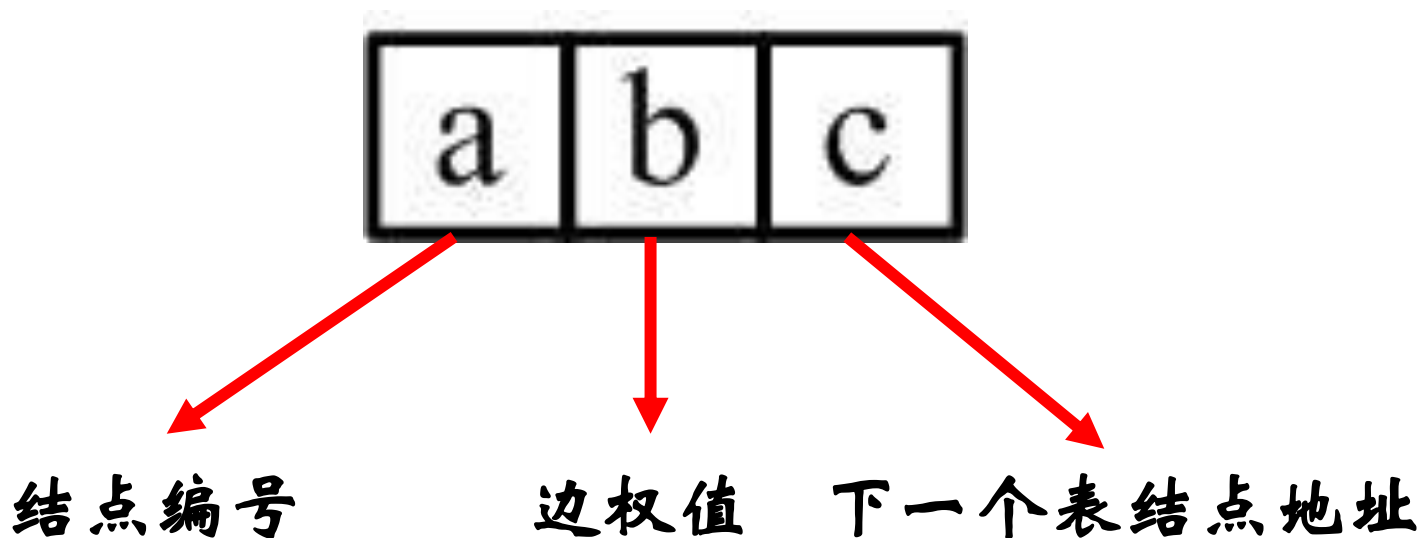
- **逆向表**：与正向表相反，逆向表是将每个结点的直接前趋集中在一起存放。
- 逆向表实质是对有向图邻接矩阵的列进行压缩的结果。

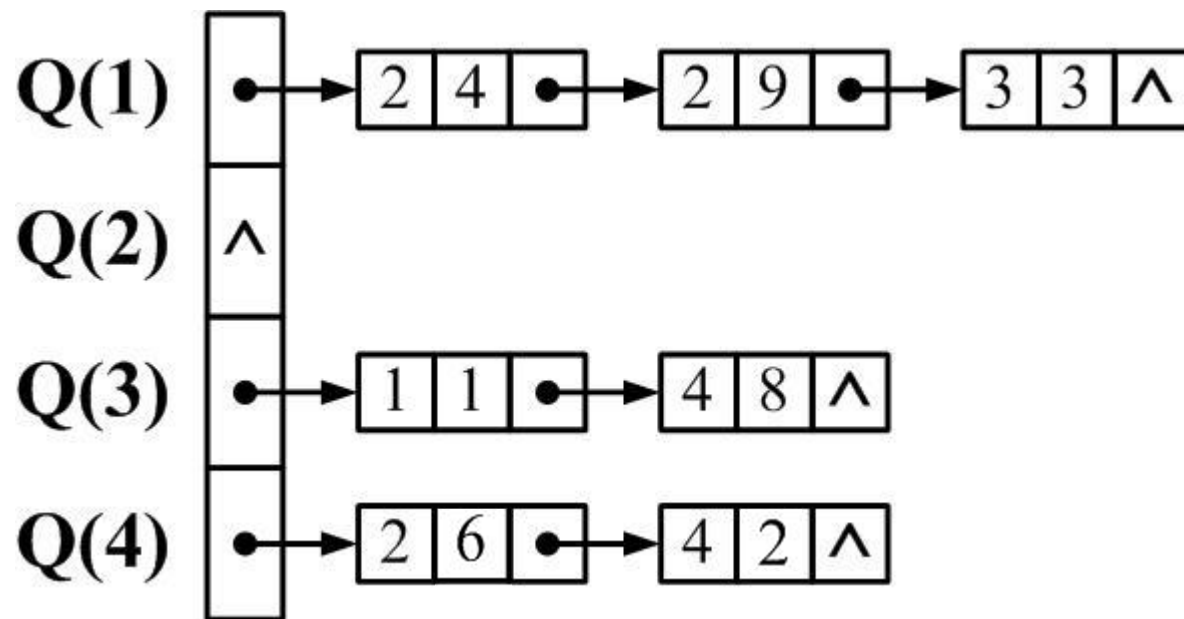
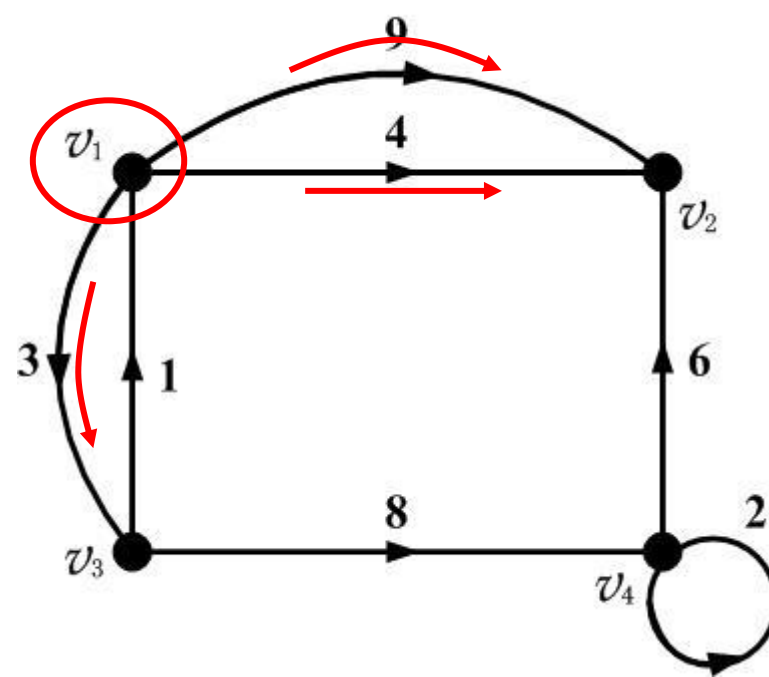
思考：正（逆）向表的优缺点是什么？



图的代数表示

- **邻接表**：采用单链表结构表示一个图。其基本单元为**表结点**，如下所示：







图的代数表示—小结

- 图的代数表示方法：
 - 邻接矩阵
 - 权矩阵
 - 关联矩阵
 - 边列表
 - 正向表
 - 逆向表
 - 邻接表
- 思考：各种表示方法如何相互转换



主要内容

- 图论概述
- 图的基本概念及定义
- 图的代数表示



作业

- 课后习题
 - 2、3、8
- 选作：
 - 1, 4