# 面向计算机科学的离散数学

# 图论基本概念

**苏航**
suhangss@mail.tsinghua.edu.cn
**清华大学 计算机系**

# 锻炼：表达能力与勇气

- **从斯坦福到印度学生**
  - 你善于做高数作业还是随机游走？
  - 因为英文不好，所以难于提问？
  - 《硅谷被印度人统治》？
- **鼓励积极参与课堂和回答问题**
  - 附加分☺

# 本节课主要内容

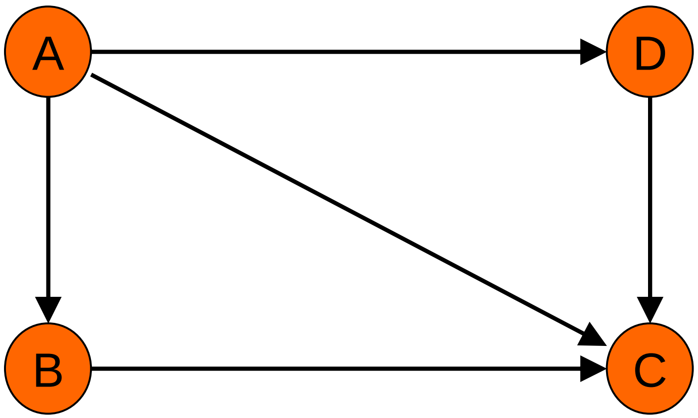
- 图的基本概念和定义
- 图的代数表示

# 图的基本概念

- 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、现象用顶点（或结点）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

# 图的基本概念(2)

- 例1
  - A、B、C、D四个队进行比赛，请用图论建模
  - 可用点代表队
  - 有向边代表输赢关系



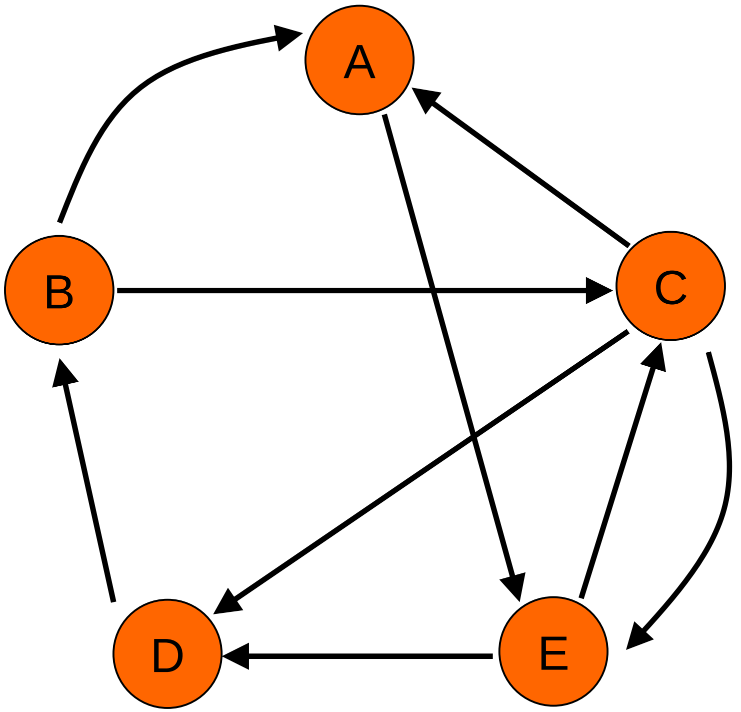

表示A胜B、C、D，B胜C，D胜C，B和D还未比赛

# 图的基本概念(3)

- **例2**
  - 有5个仓库A-E，其中A可以到E；B可以到A，C；C可以到A、D、E；D可以到B；E可以到C，D。问:
  - 1）E能否到B？
  - 2）任意两个仓库之间是否都有道路可达？

如何判断任意两个仓库之间是否都有道路可达？

**AECDBA**



# 图的基本概念(4)

- 例3 （缅怀金庸）
  - 未婚的段誉总是痴痴地看着王语嫣，王语嫣却眼不离慕容复，可惜慕容复已经结婚了。
  - 是否有个未婚者在看已婚者？

段誉 → 王 → 慕

点着色：未婚着红色；已婚着蓝色

段誉 → 王 → 慕

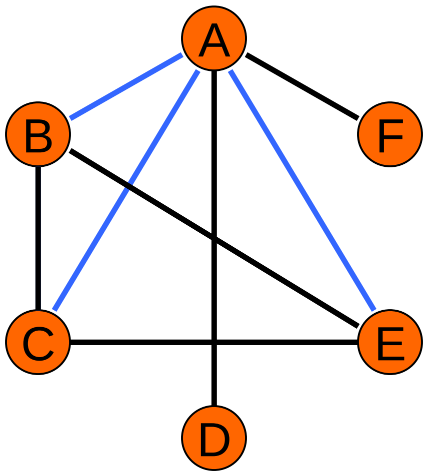
未婚　　如何着色?　　已婚

题目转换为：是否存在红到蓝的边?

# 图的基本概念(5)

- **例4**
  - 试证明：6个人中如果没有3个人互相认识，则至少有3个人相互不认识。

图论建模？




- **若两个人认识，用蓝色边连接，否则用黑色边连接**
- 如果没有3个人互相认识（不存在蓝色三角形）
- 至少有3个人相互不认识（存在黑色三角形）

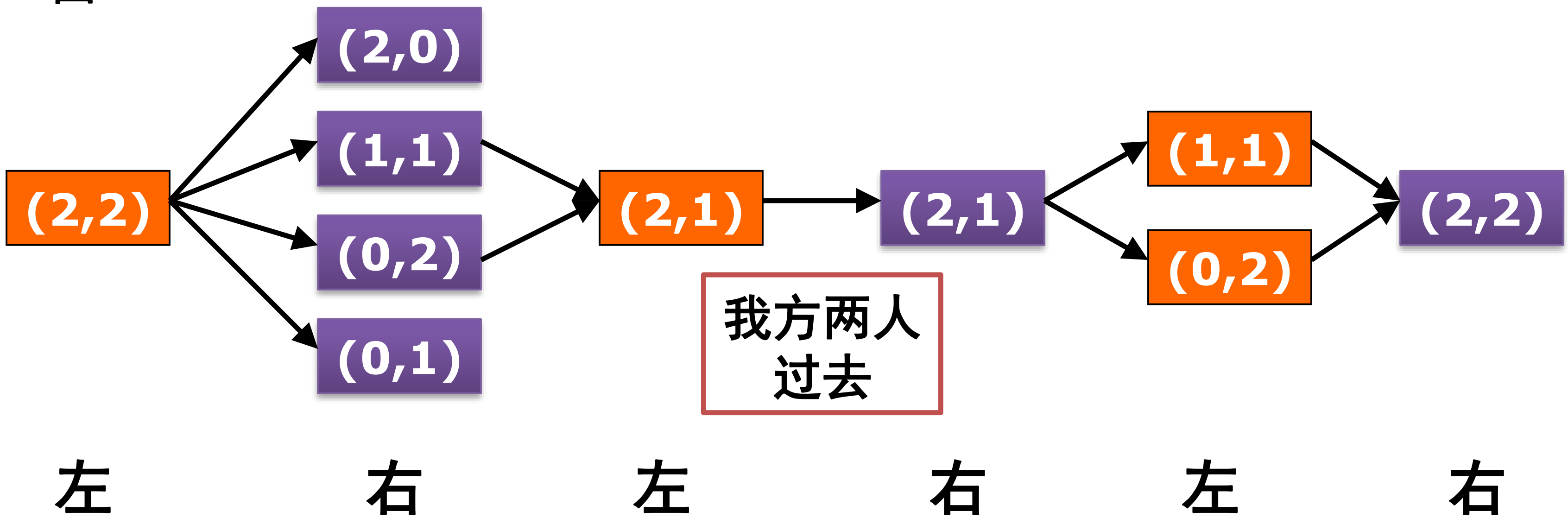
- **问题转化为证明图中存在黑色或蓝色三角形**
- **不失一般性，假设与A连接的五条边有三条蓝色边**
- **若B、C、E三点间的三条边中有一条蓝色边，便与A构成了蓝色三角形，问题证毕**
- **否则B、C、E之间构成黑色三角形，问题证毕**
- **如果与A连接的5条边有三条黑色边，则类似可证**

# 图的基本概念(6)

- **例5**
  - 敌我双方各2人组成一个小组到前方视察，途经一条河，河上只有一条小船，该船每次最多只能载2人。试问，在保证两岸我方人员不少于敌方人员的前提下，能否过河？如果可以，最少的渡河次数是多少？

**图论建模？**

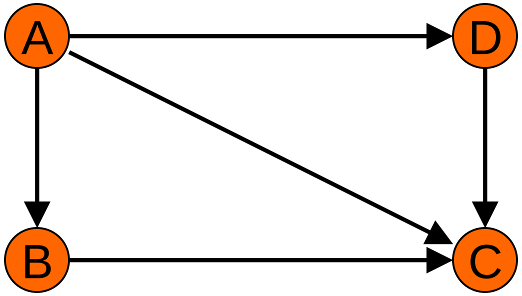
**假定是从左岸到右岸。将敌我双方在岸上的人数作为顶点，表示为：（我方人员数a，敌方人员数b）。这时，可以通过摆渡1人或2人将两岸间的某二个点加以联系，构成边。这样，根据约束条件逐步从左至右构成下图：**



# 图的定义

为解决问题和设计好算法，该怎么办？
**形式化描述！**

- **图：二元组 (V(G),E(G))称为图。其中V(G)是非空集，称为结点集，E(G)称为边集，E(G)为V(G)×V(G)的子集。以后简记G=(V,E)**
  - **例**



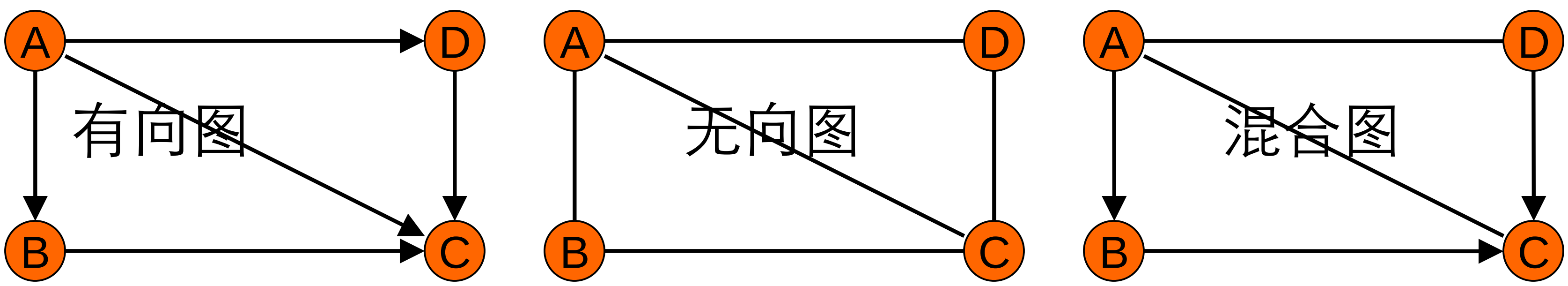

  - V={A,B,C,D}, E={(A,B), (A,C), (A,D), (B,C), (D,C)}

# 图的定义(2)

- 图分为有限图和无限图
  - 仅讨论有限图（V和E均有限）
- 图的阶
  - $V=\{v_1, v_2, v_3, \ldots, v_n\}$
  - $E=\{e_1, e_2, e_3, \ldots, e_m\}$
  - $|V|=n$ ——图的阶
  - $|E|=m$
  - **空图：若 $|E|=0$**

# 图的定义(3)

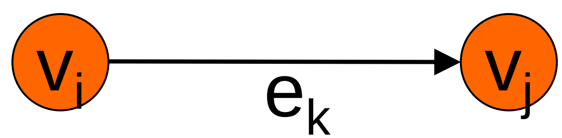
- 有向图与无向图（Why？）
  - 若图中的边都为有向的，则称为有向图
  - 若图中的边都为无向的，则称为无向图
  - 若图中既有有向边，又有无向边，则称为混合图。



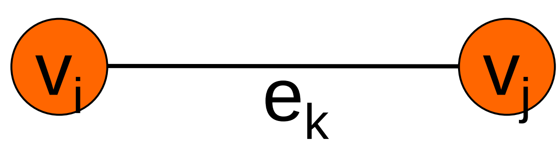

**请举个实际生活的例子?**

# 图的定义(4)

- 图的边可用$e_k=(v_i, v_j)$表示：图的核心是关联关系的刻画
  - 称$v_i$, $v_j$ 为相邻结点， $e_k$分别与$v_i$, $v_j$相关联
  - 若$e_k$是有向边

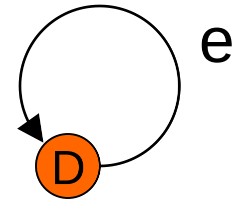


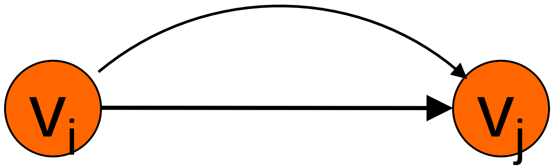

    - 称$v_i$是$e_k$的始点， $v_j$是$e_k$的终点
    - $v_i$是$v_j$的直接前驱， $v_j$是$v_i$的直接后继
  - 若$e_k$是无向边，称$v_i$, $v_j$是$e_k$的两个端点



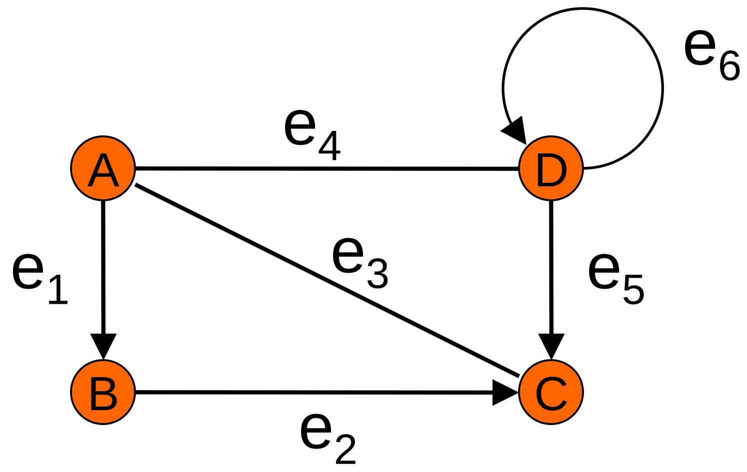
# 图的定义(5)

- 自环
  - 只与一个结点关联的边称为自环
- 重边
  - 若一对结点之间有多条边，则称为重边
- 多重图
  - 含有重边的图称为多重图
- 孤立点
  - 若v没有关联的边，则称v为孤立点



# 图的定义(6)




- A是B的直接前驱，C是B的直接后继
- A，C是相邻结点，是无向边$e_3$的端点
- $e_6$是自环

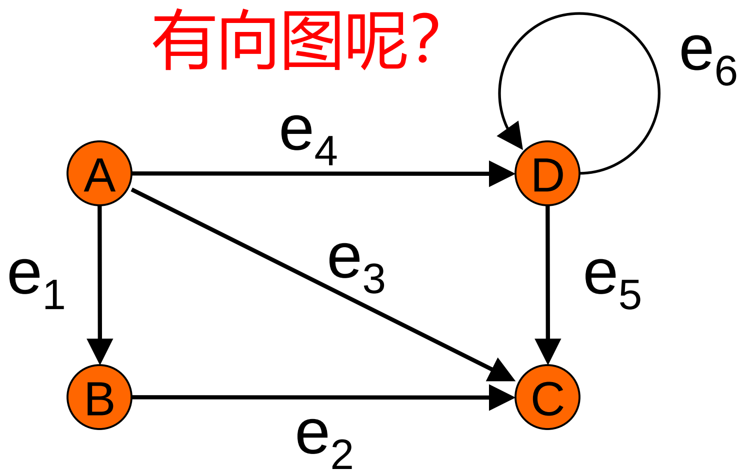
1点1边？　1点2边？　2点2边？

# 图的定义(7)

- **结点的度**
  - 无向图：与结点v关联的边数称为结点v的度d(v)
  - 有向图：正度$d^+(v)$与负度$d^-(v)$
    - 正度$d^+(v)$(v为始点的边的数目，出度）
    - 负度$d^-(v)$ (v为终点的边的数目，入度）
    - $d^+(v) + d^-(v) = d(v)$

发明定义时：无向图与有向图的统一

有向图呢?

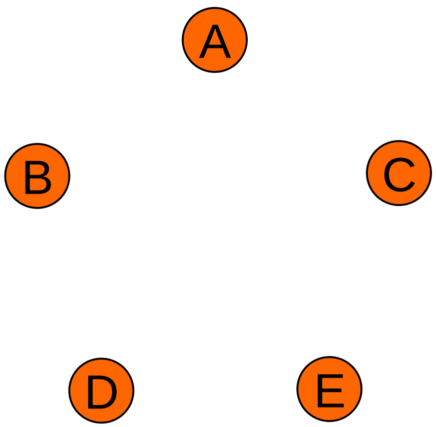



| | $d^+$ | $d^-$ | d |
|---|---|---|---|
| A | 3 | 0 | 3 |
| B | 1 | 1 | 2 |
| C | 0 | 3 | 3 |
| D | 2 | 2 | 4 |

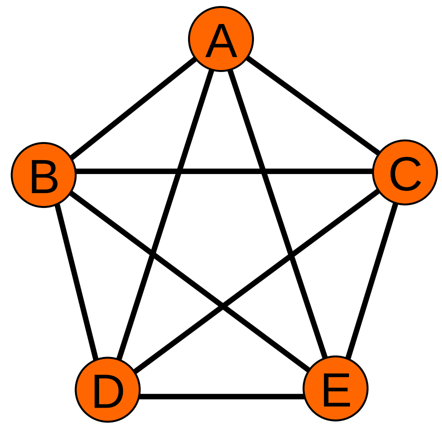
# 图的定义(8)

## ◆ 简单图

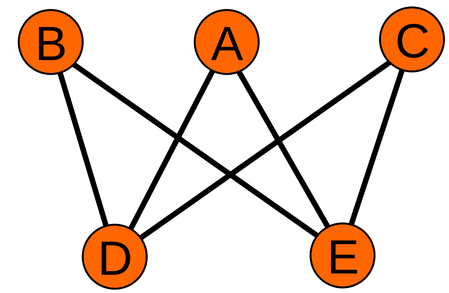
- 无重边、无自环的无向图称为简单图




$N_5$




$K_5$




$K_{3,2}$

- 空图$N_n$，完全图$K_n$，二分图，完全二分图$K_{m,n}$

# 图的定义(9)

- **图的性质：握手定理**
  - 1. 设图G有n个结点，m条边，则下式成立：

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

  - 证明:
    - 对每条边e=（i，j），都分别在d(i)，d(j)中各计1次，其对度的贡献为2。
    - 因为共有m条边，所以总度数为2m。
  - 适用范围
    - 有向图&无向图（“度”定义的重要性！）

# 图的定义(9)

- **图的性质**
  - 2. G中度为奇数的点的个数为偶数个。
    - 证明：
    - G中任意一个结点的度或为奇数，或为偶数；
    - 设$V_e$是度为偶的结点集，$V_o$是度为奇的结点集。
    - 因此有

$$\underbrace{\sum_{v \in V_e} d(v)}_{\text{偶度结点}} + \underbrace{\sum_{v \in V_o} d(v)}_{\text{奇度结点}} = 2m$$

    - 所以度为奇的结点度之和为偶数，因此$V_o$中含有偶数个结点。

# 图的定义(9)

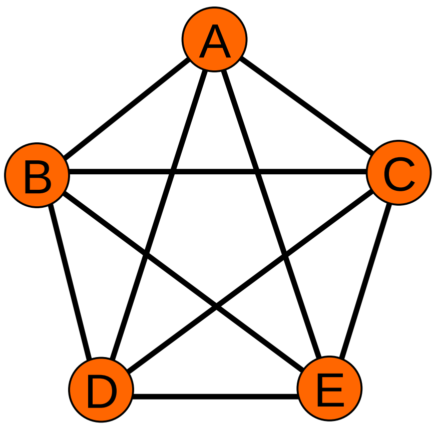
- **例**
  - 在9个工厂之间
    - 不可能每个工厂都只与其它3个工厂有业务联系。
    - 不可能只有4个工厂与偶数个工厂有业务联系。
  - 证：
    - 图论建模
      - 用结点表示工厂，用边表示工厂之间的业务联系
    - 若每个工厂都只与其它3个工厂有业务联系，即每个结点的度为3，共有9个结点，与性质2矛盾；
    - 同理，若只有4个工厂与偶数个工厂有业务联系，则有5个结点的度为奇数，也得出矛盾。

# 图的定义(10)

- **图的性质**
  - 3. 有向图中结点的正度之和等于负度之和。
    - 证：每条边对正、负度的贡献为1，因此正负度之和相等
  - 4. $K_n$的边数为 $\frac{1}{2}n(n-1)$
    - 证：因为完全图中每点的度都是n-1。

$\because \sum d(v) = n(n-1) = 2m$ .

$\therefore m = \frac{1}{2}n(n-1)$




$K_5$的边数？

# 图的定义(10)

- **图的性质**
  - 5. 非空简单图中一定存在度相同的结点。
    - 提示：鸽笼原理?
    - 证：假设图中没有孤立点，那么n个结点的度取值范围为1~(n-1)，根据鸽笼原理，至少有两个结点的度相同。

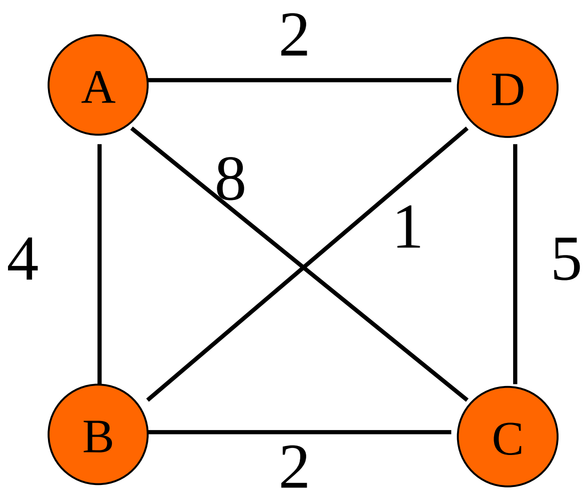
    若有孤立点，则任意结点的度数最大为(n-2)，类似可证。

**非空无向图中一定存在度相同的结点？**

# 图的定义(11)

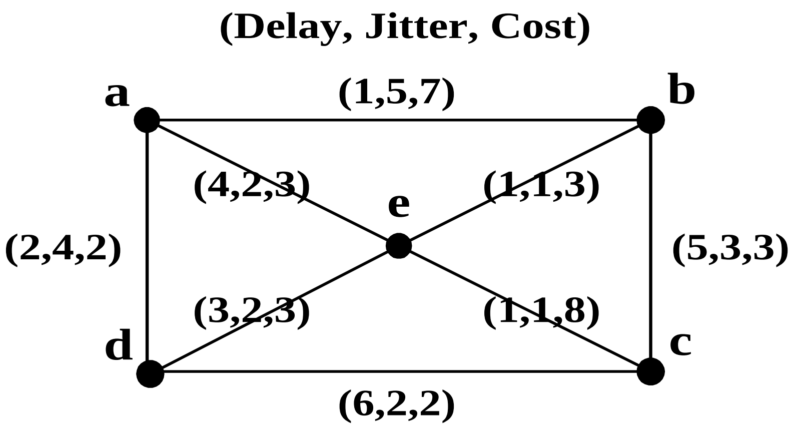
- 赋权图（Why）
  - 图中每一边$e_k$都赋予一个实数$w_k$作为该边的权
  - 正权图：每边的权均为正数

谁说最小就最好？




A->C的最优路径？

互联网服务质量路由




寻找路径$(a \Rightarrow c)$满足约束$c=(7,8,9)$?

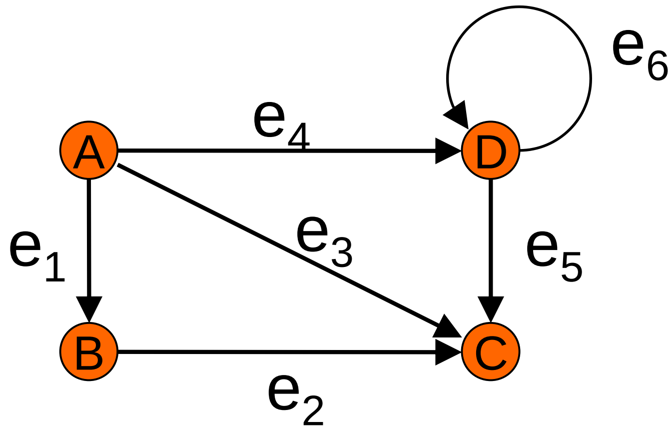
即$w(a \Rightarrow c) \leq (7,8,9)$

权的可加性？

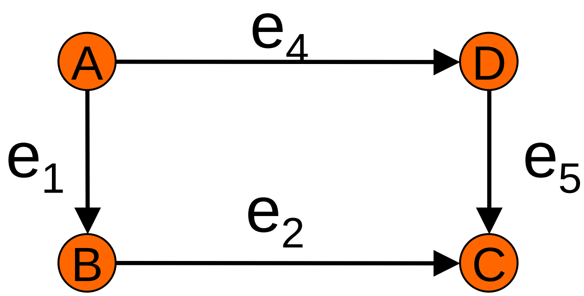
# 图的定义(11)

- 子图
  - 对图G=(V,E)与G'=(V',E')，若V'包含于V，E'包含于E，则称G'为G的一个子图（真子图）；
  - 特别若V=V', 则称G'为G的支撑子图或生成子图；
  - 若E'包含了G在结点子集V'的所有边，则称G'为G的导出子图。
  - G的空支撑子图与G本身称为G的平凡子图

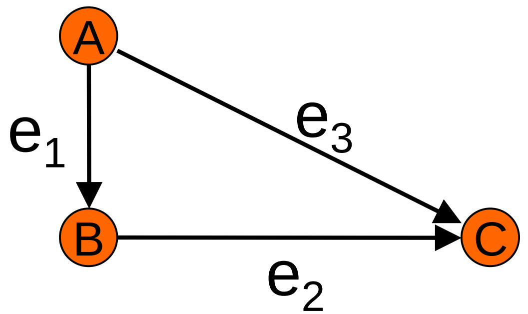
# 图的定义(11)




**G**




**G的支撑子图**




**G的导出子图**

（边全吗？）

# 图的定义(12)

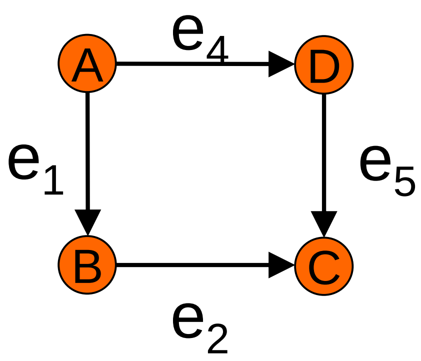
- **图的并、交、对称差**
  - 两个图G1=(V1, E1), G2=(V2, E2)。

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

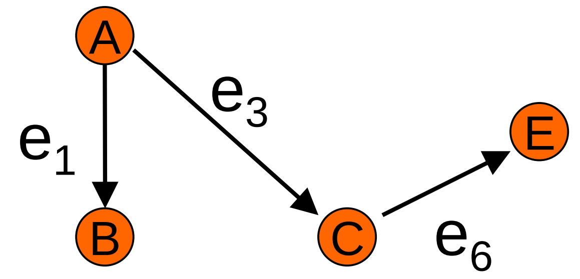
$$G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$$

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

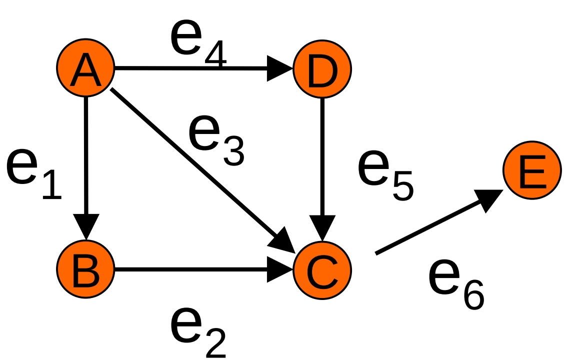
# 图的定义(12)



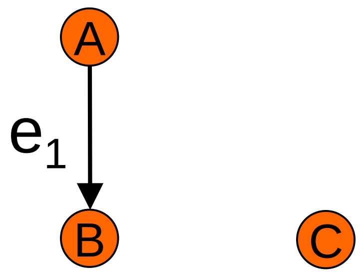

**$G_1$**



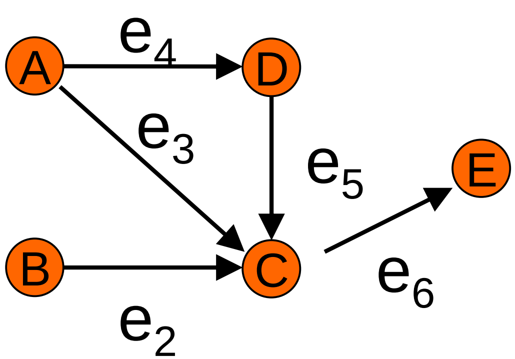

**$G_2$**




$G_1 \cup G_2$




$G_1 \cap G_2$




$G_1 \oplus G_2$

**是否可能增加其他形式的定义（如点交边并）？**

# 图的定义(12)

- **图的差**
  - G – H
    - 对于G的子图H，G - H表示在G中删除H中的各条边得到的图。
  - 补图
    - 对于简单图G，$K_n$ - G称为G的补图。

# 图的定义(13)

- **对于无向图G，结点v的邻点集**

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

- 有向图呢？
- 设v是**有向图G**的一个结点，则

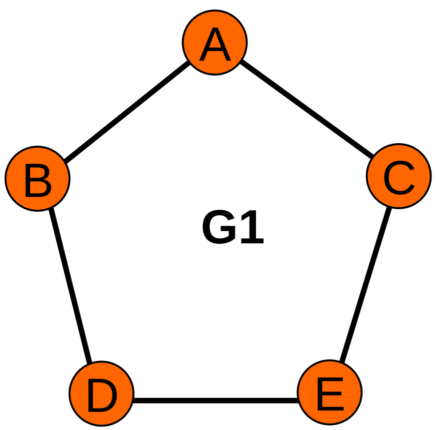
$$\Gamma^{+}(v) = \{u \mid (v, u) \in E\}$$

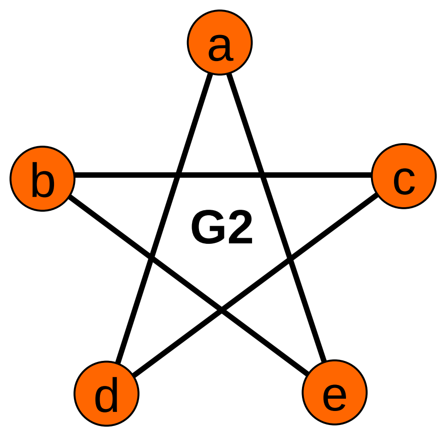
$$\Gamma^{-}(v) = \{u \mid (u, v) \in E\}$$

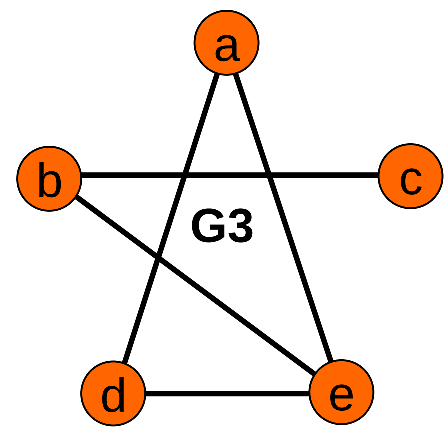
分别称为v的**直接后继集（外邻集）**与**直接前趋集（内邻集）**

# 图的定义(14)

- **同构**　　　　**相同的图？长得一样?**
  - 两个图$G_1=(V_1, E_1)$, $G_2=(V_2, E_2)$，若在$V_1$与$V_2$之间存在双射$f$，当且仅当$(u,v)$为$G_1$的边时，$(f(u),f(v))$为$G_2$的边。称图$G_1$与$G_2$同构。记作 $G_1 \cong G_2$



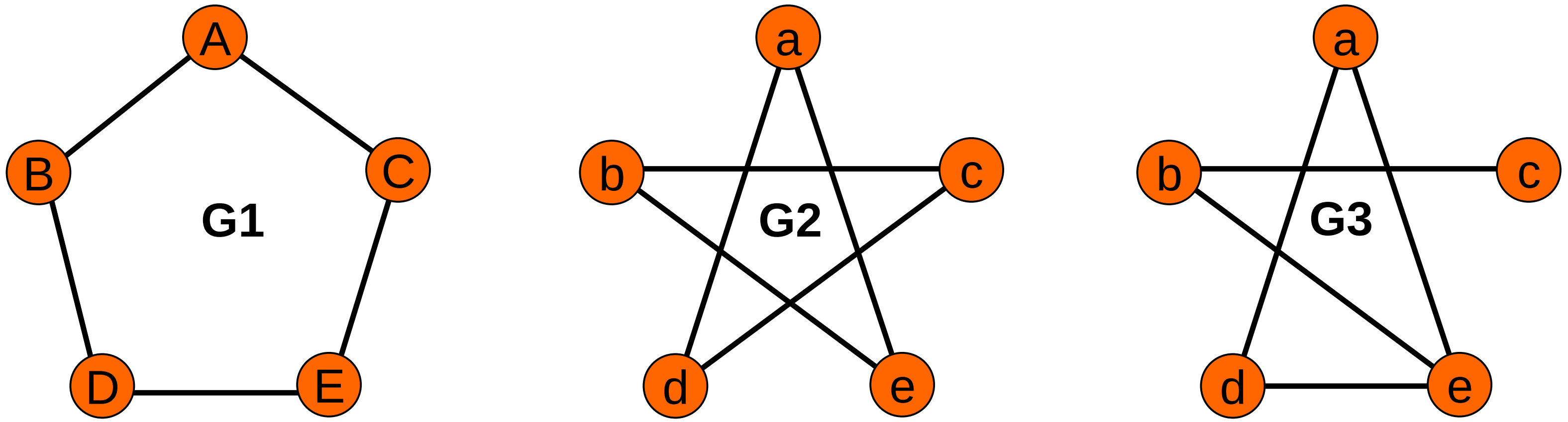




**f(A)=a, f(B)=d, f(C)=e, f(D)=c, f(E)=b**

**不同构**

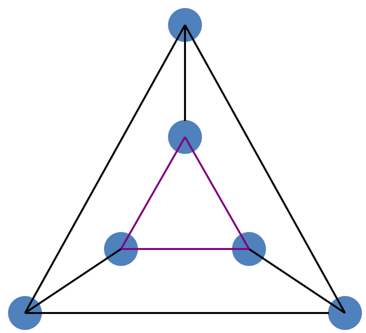
# 图的定义(15)

- **同构的性质**

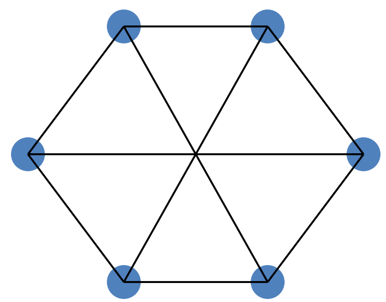



- **若 $G_1 \cong G_2$，则有**
  - 顶点数、边数相同：n=5，m=5
  - 结点度非减序列相同
    - G1(2,2,2,2,2), G3(1,2,2,2,3)
  - 存在同构的导出子图
    - G3存在三角形导出子图

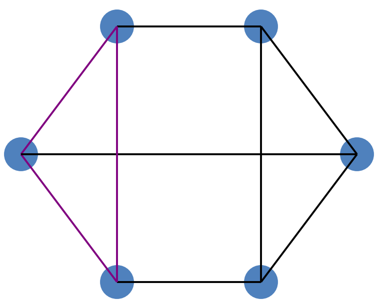
# 图同构(举例)



$G_1$

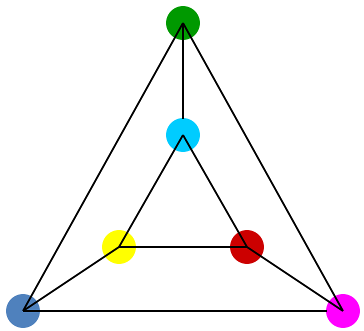


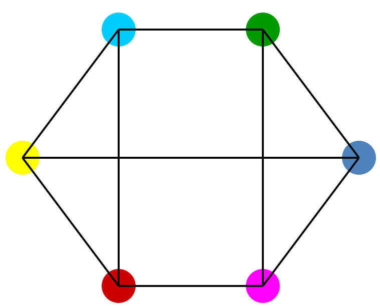
$G_2$



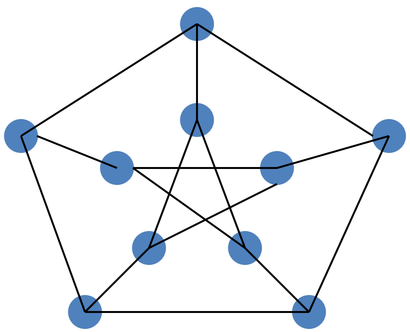
$G_3$

$G_1 \cong G_3$，  $G_1 \ncong G_2$

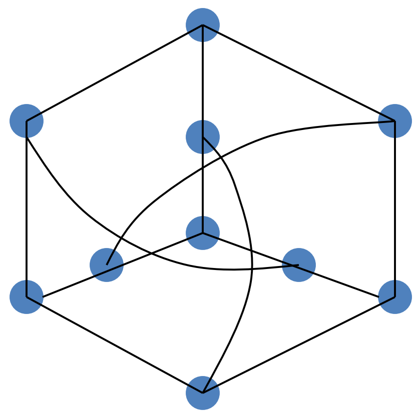




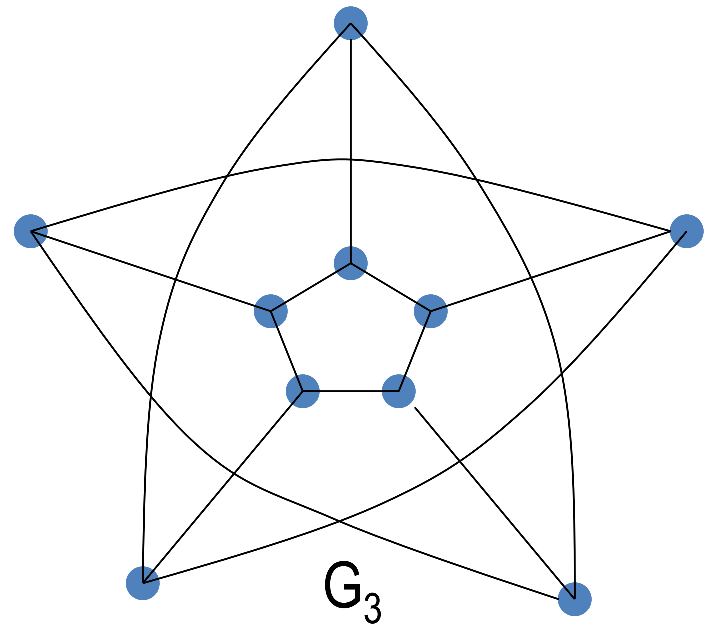
# 图同构



$G_1$



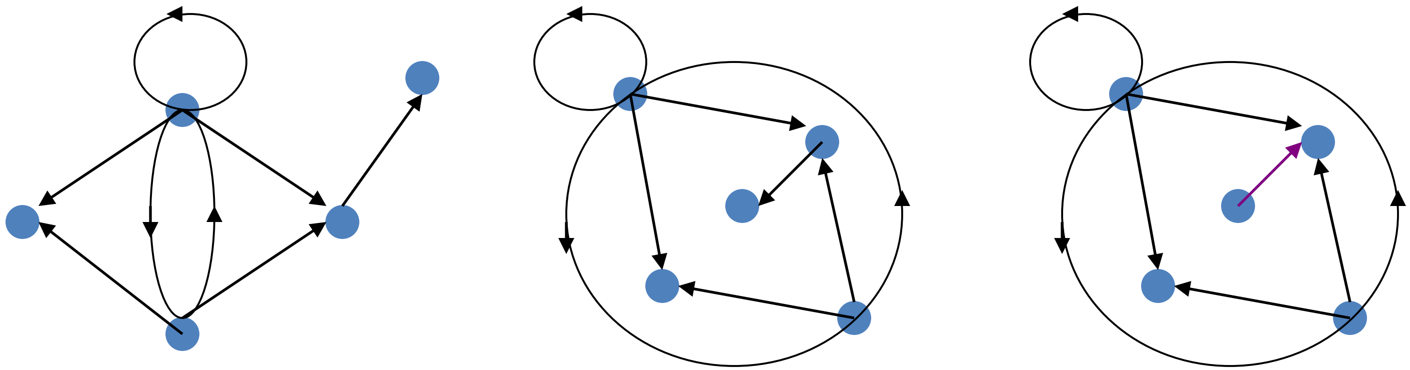
$G_2$



$G_3$

$$G_1 \cong G_2 \cong G_3$$

# 图同构



$D_1$ $D_2$ $D_3$

$$D_1 \cong D_2, \quad D_2 \ncong D_3$$

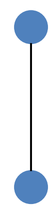
# 图族(graph class)

- 完全图, 有向完全图, 竞赛图
- 柏拉图图, 彼德森图, 库拉图斯基图
- r部图, 二部图(偶图), 完全r部图
- 超立方体

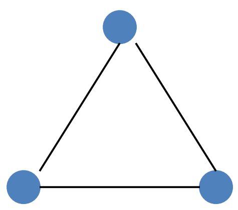
# 完全图

- 完全图是一个简单的无向图，其中每一对不同的顶点都只有一条边相连

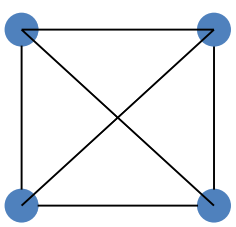
$K_1$



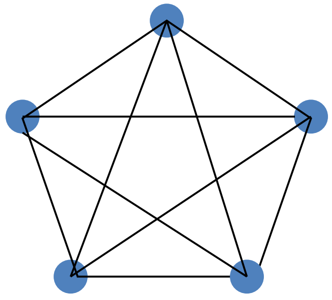
$K_2$



$K_3$

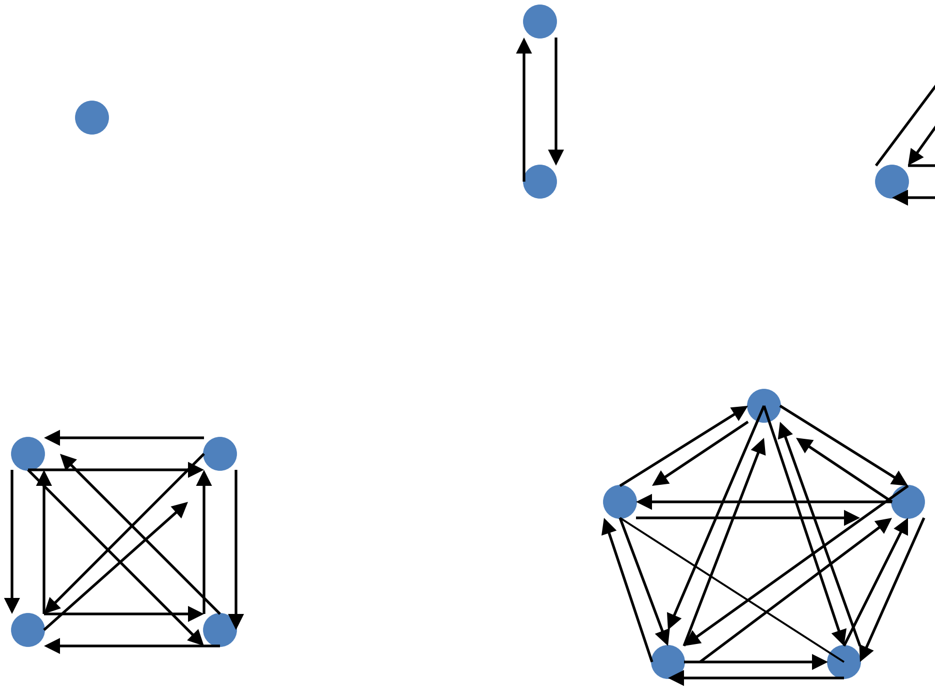


$K_4$



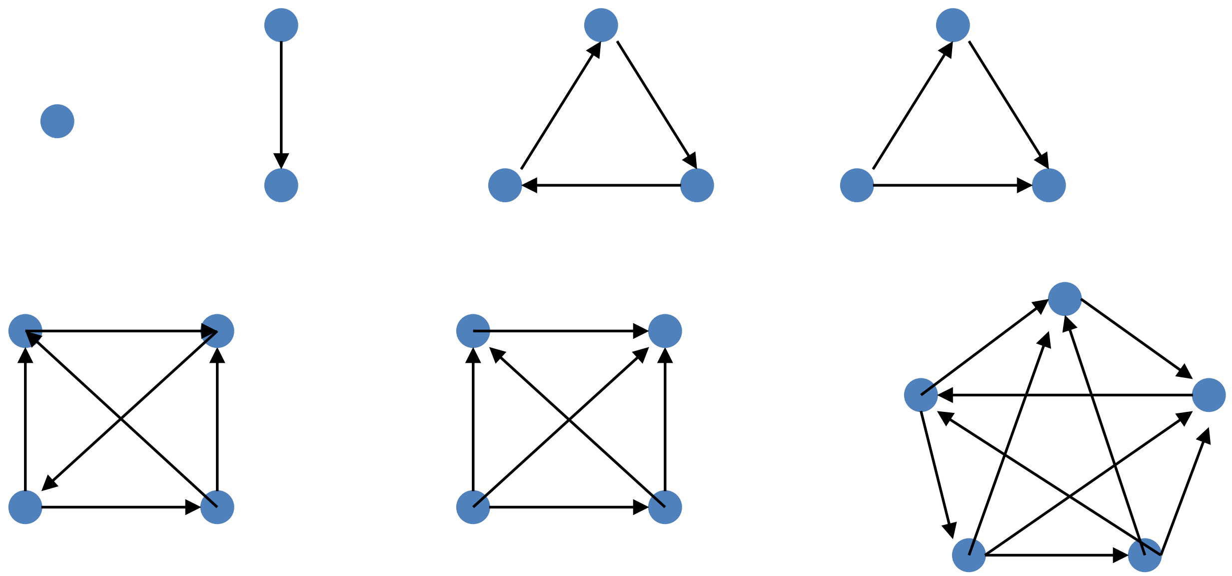
$K_5$

# 有向完全图



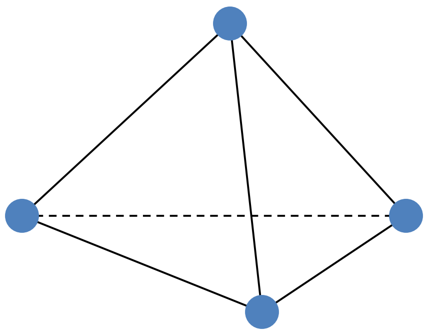
# 竞赛图

- 竞赛图是通过在无向完全图中为每个边分配方向而获得的有向图。
- 每对顶点之间都有一条边相连的有向图称为竞赛图。

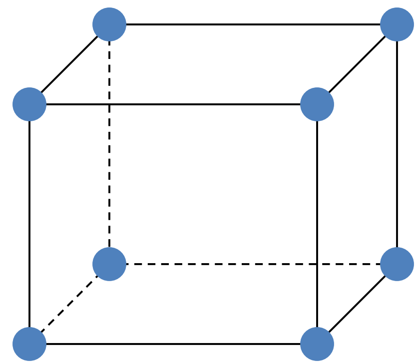


# 柏拉图图

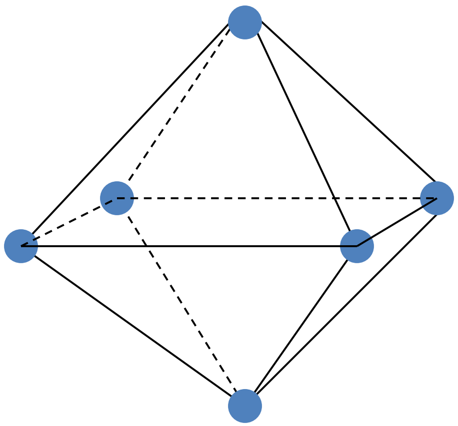
- 正四面体、正六面体、正八面体、正十二面体和正二十面体所相应的图的统称



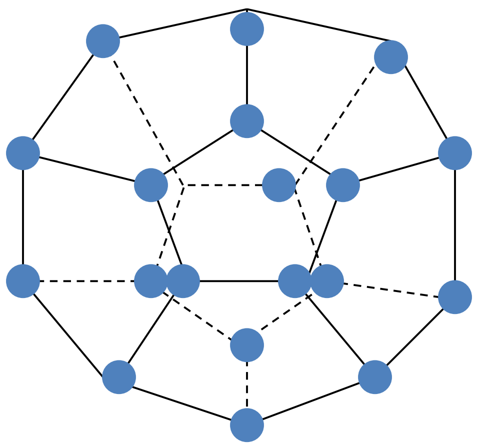
正四面体图



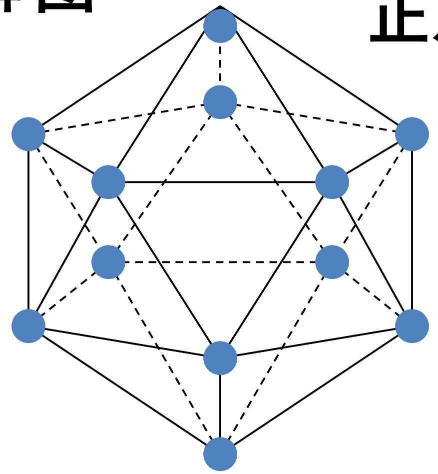
正六面体图



正八面体图



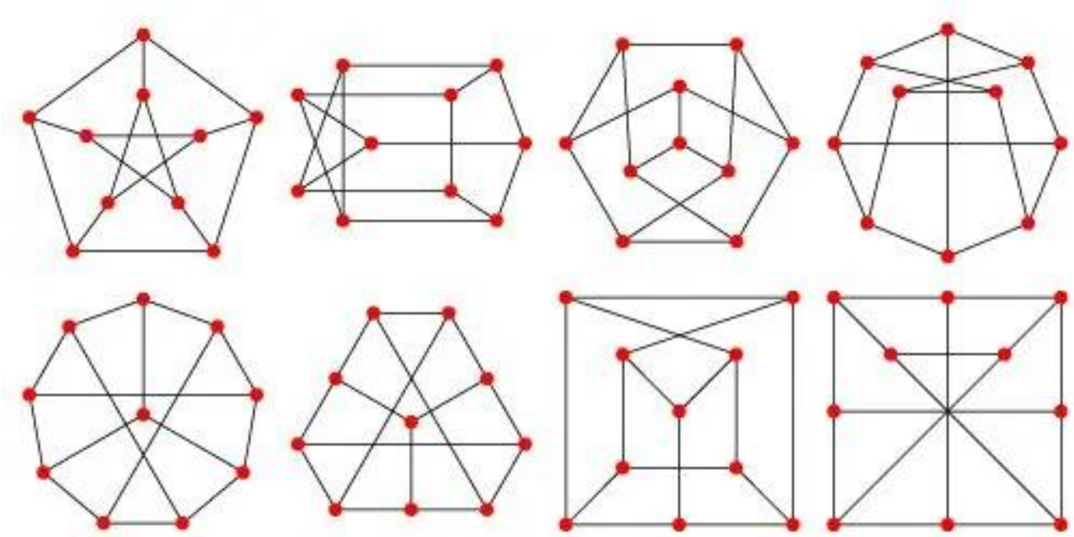
正十二面体图



正二十面体图

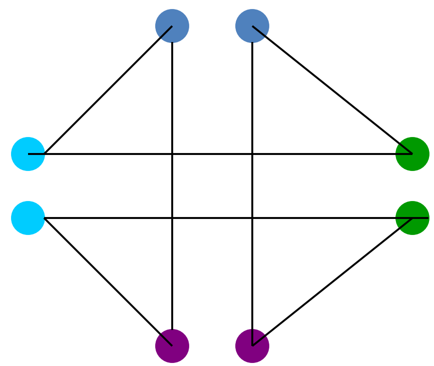
# 彼德森图(3正则图)

- 由10个顶点和15条边构成的连通简单图
- 有哈密顿路但不是哈密尔顿图，常常作为**反例**出现在图论之中
- 旋转对称+轴对称



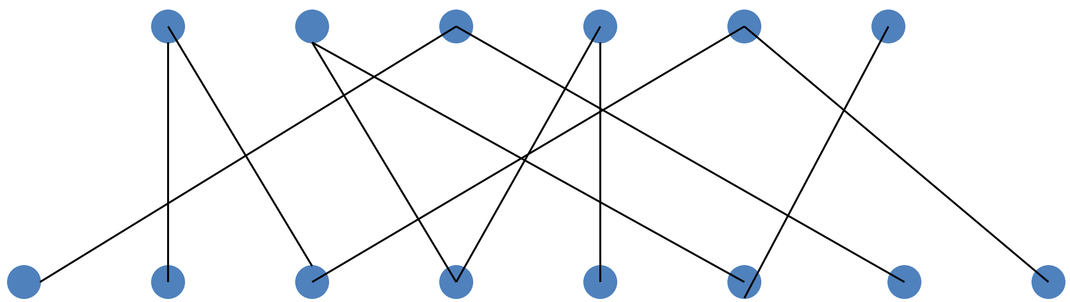
# r部图

- **r部图**: 任何一条边的两个顶点都不在同一个子集中
  - $G=<V,E>, V=V_1\cup V_2\cup\ldots\cup V_r,\ V_i\cap V_j=\varnothing\ (i\neq j),\ E\subseteq \cup(V_i \& V_j)$
- 记作 $G=<V_1,V_2,\ldots,V_r;E>$

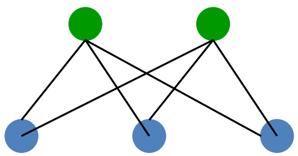


# 二部图(偶图)

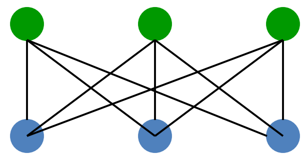
- **二部图**：$G=\langle V_1, V_2; E\rangle$，也称为**偶图**



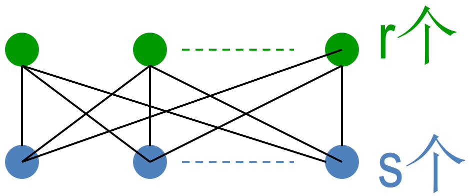
# 完全r部图



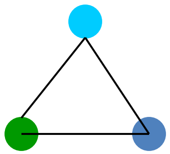
$K_{2,3}$



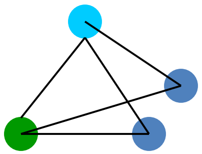
$K_{3,3}$



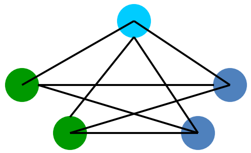

$K_{r,s}$



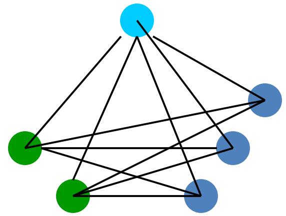
$K_{1,1,1}$



$K_{1,1,2}$



$K_{1,2,2}$



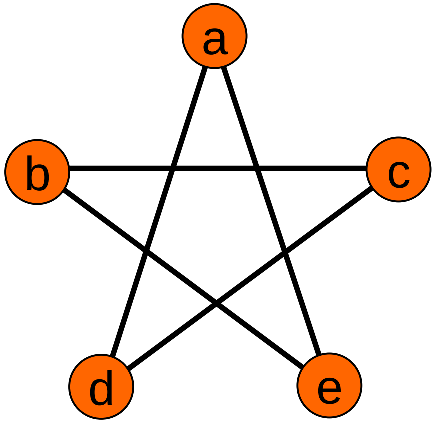
$K_{1,2,3}$

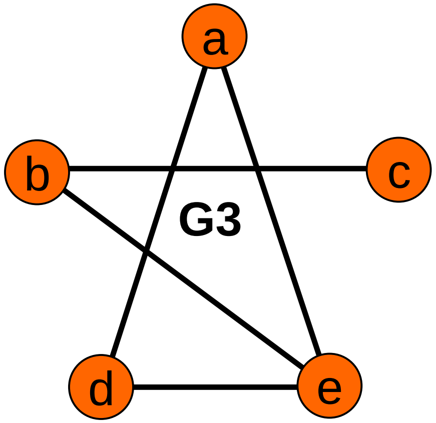
# 主要内容

- **图的代数表示方法**
  - 各种方法的特点（优缺点）

# 图的代数表示

- 火眼金睛 v. s. 计算机处理
- 计算机擅长/不擅长什么?
- 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
  - 邻接表

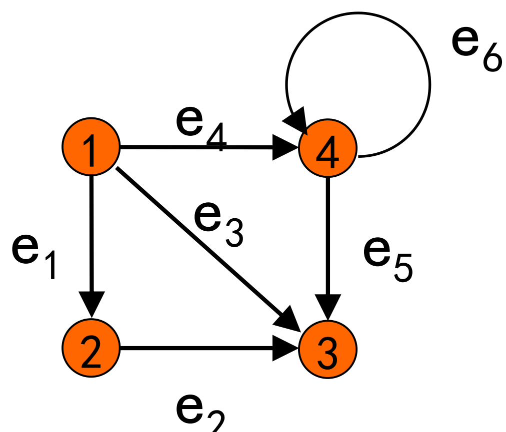




如何实现图的代数表示?

# 图的代数表示

- 发明：邻接矩阵（点&点）

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




$$\begin{array}{c} \\ v_1\\ v_2\\ v_3\\ v_4 \end{array}\begin{array}{cccc} v_1 & v_2 & v_3 & v_4\\ \end{array}$$

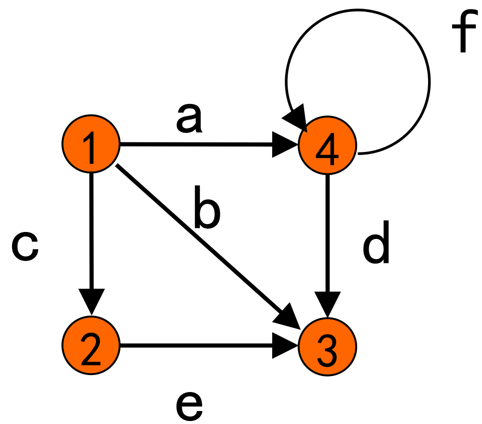
$$\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4\\ \hline v_1 & 0 & 1 & 1 & 1\\ v_2 & 0 & 0 & 1 & 0\\ v_3 & 0 & 0 & 0 & 0\\ v_4 & 0 & 0 & 1 & 1 \end{array}$$

- 自己的发明有什么特点？
  - 对于有向图的邻接矩阵中，$v_i$的正度和$v_i$的负度？
    - 第i行的1的个数表示$v_i$的正度，第i列的1的个数表示$v_i$的负度
  - 邻接矩阵可表示自环，但是不能表示重边

# 图的代数表示(2)

- **权矩阵**
  - 赋权图用权矩阵表示

除了点&点外，还有什么？
继续发明……



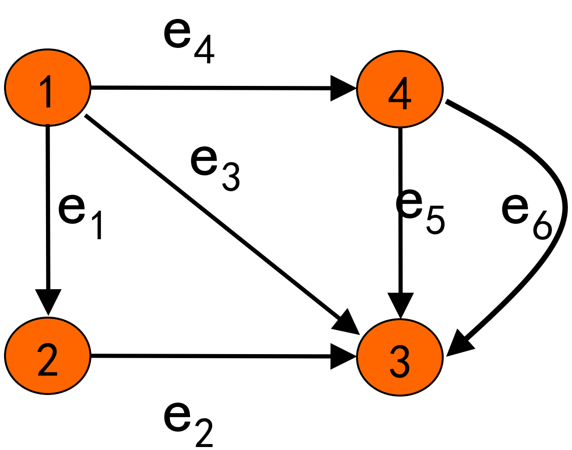

$$\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & c & b & a \\ v_2 & 0 & 0 & e & 0 \\ v_3 & 0 & 0 & 0 & 0 \\ v_4 & 0 & 0 & d & f \end{array}$$

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

# 图的代数表示(3)

- **发明：关联矩阵（n点&m边）**
  - 有向图

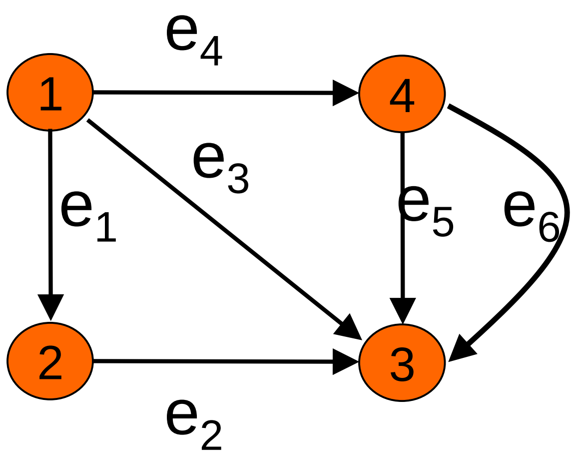
$$B = [b_{ij}]_{n\times m} \quad b_{ij} = \begin{cases} 1 & e_j = (v_i, v_k) \in E \\ -1 & e_j = (v_k, v_i) \in E \\ 0 & \text{其它} \end{cases}$$




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

# 图的代数表示(4)

- **关联矩阵**




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

- 关联图性质（有向图）
  - 每列只有一个1和一个−1
  - 每行中1的个数为相应结点的正度，−1个数为负度；
  - 能表示重边，不能表示自环

# 有向图关联矩阵性质

- 每列和为零：$\Sigma^{n}_{i=1}m_{ij}=0$
- 每行绝对值和为d(v)：$d(v_i)=\Sigma^{m}_{j=1}m_{ij}$,

  其中 1的个数为$d^{+}(v)$,

  -1的个数为$d^{-}(v)$
- 握手定理： $\Sigma^{n}_{i=1}\Sigma^{m}_{j=1}m_{ij}=0$
- 平行边：相同两列

# 有向图关联矩阵性质

- 每列和为零：$\Sigma^{n}_{i=1}m_{ij}=0$
- 每行绝对值和为d(v)：$d(v_i)=\Sigma^{m}_{j=1}m_{ij}$，
  其中 1的个数为$d^{+}(v)$,
  -1的个数为$d^{-}(v)$
- 握手定理： $\Sigma^{n}_{i=1}\Sigma^{m}_{j=1}m_{ij}=0$
- 平行边：相同两列

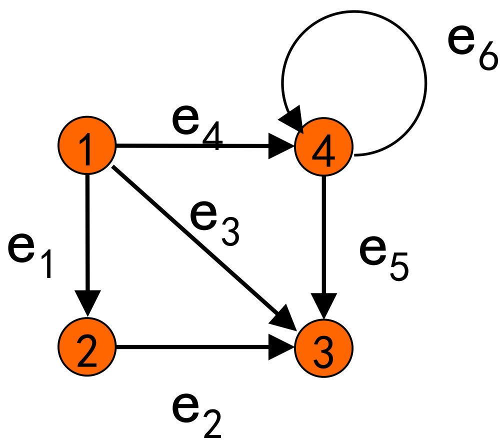
# 图的代数表示(6)

$$\begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline v_1 & 0 & 1 & 1 & 1 \\ v_2 & 0 & 0 & 1 & 0 \\ v_3 & 0 & 0 & 0 & 0 \\ v_4 & 0 & 0 & 1 & 1 \end{array}$$

**邻接矩阵**

$$\begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 0 & 1 & 1 & 0 & 0 \\ v_2 & -1 & 1 & 0 & 0 & 0 & 0 \\ v_3 & 0 & -1 & -1 & 0 & -1 & -1 \\ v_4 & 0 & 0 & 0 & -1 & 1 & 1 \end{array}$$

**关联矩阵**




- **基本表示的唯一性?**
  - 邻接矩阵与关联矩阵表示图是唯一的
- **能否有其他的表示方法?**
  - 点和点 or 点和边
  - 边和边?

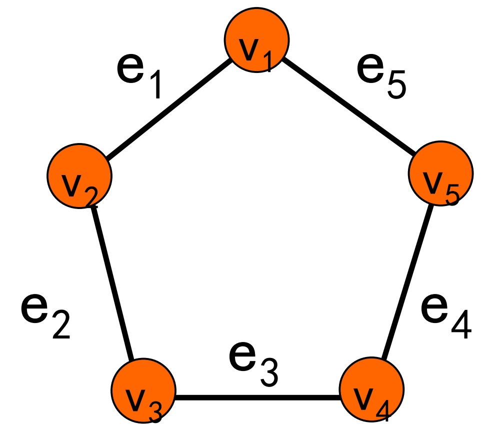
# 图的代数表示(7)

$$\begin{array}{cc} & \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

**邻接矩阵**

$$\begin{array}{cc} & \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} & \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 & -1 \\ 0 & 0 & 0 & -1 & 1 & 1 \end{bmatrix} \end{array}$$

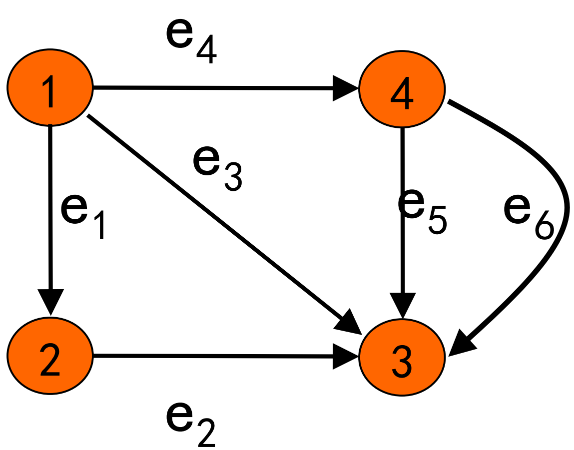
**关联矩阵**




- **基本矩阵表示存在什么问题？**
  - 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（稀疏矩阵）
- **因此引入边列表、正向表、逆向表、邻接表等**

# 图的代数表示(8)

- **边列表**
  - 首先看存在什么问题？再看从哪个角度优化？
  - 对列进行压缩？




信息量的核心是非零元的位置

$$
\begin{array}{c|cccccc}
 & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\
\hline
v_1 & 1 & 0 & 1 & 1 & 0 & 0 \\
v_2 & -1 & 1 & 0 & 0 & 0 & 0 \\
v_3 & 0 & -1 & -1 & 0 & -1 & -1 \\
v_4 & 0 & 0 & 0 & -1 & 1 & 1
\end{array}
$$

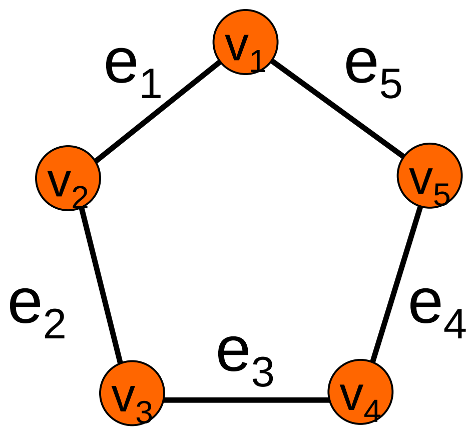
A：（1　2　1　1　4　4）

B：（2　3　3　4　3　3）

# 图的代数表示(9)

- **边列表**
  - 对关联矩阵的列进行压缩
  - 边列表由两个m维向量A和B组成
  - 当对G的结点与边进行编号后
  - 对第k条边$e_k=(v_i, v_j)$，则A(k)=i，B(k)=j，即A(k)存放第k条边始点的编号，B(k)存放其终点标号
    - 对赋权图怎么办？

**用m维向量Z存放权，$Z(k)=w_k$。**




A: (1 2 3 4 5)

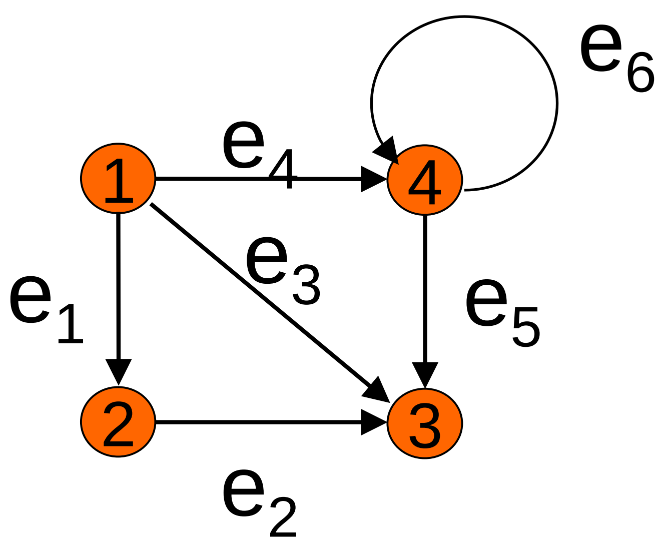
B: (2 3 4 5 1)

# 图的代数表示(10)

- **正向表**
  - 如何优化邻接矩阵？
  - 对行进行压缩？
  - 直接后继从哪里开始？

**直接后继节点？**
**排列一起？**

| | | | | | |
|---|---|---|---|---|---|
| 2 | 3 | 4 | 3 | 3 | 4 |



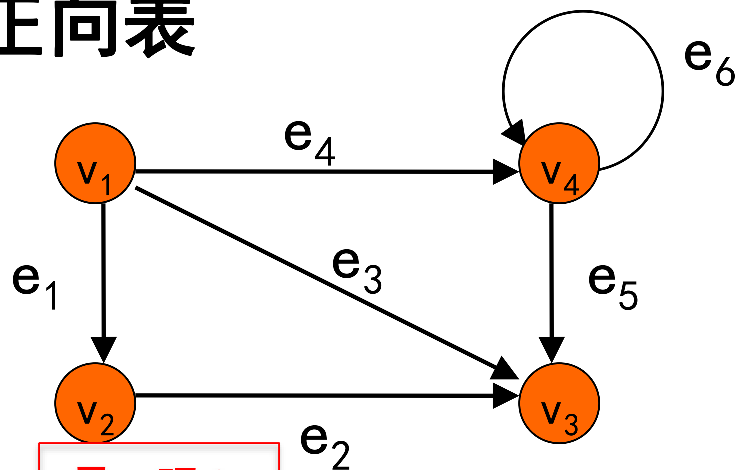

$$
\begin{array}{c|cccc}
 & v_1 & v_2 & v_3 & v_4 \\
\hline
v_1 & 0 & 1 & 1 & 1 \\
v_2 & 0 & 0 & 1 & 0 \\
v_3 & 0 & 0 & 0 & 0 \\
v_4 & 0 & 0 & 1 & 1
\end{array}
$$

# 图的代数表示(11)

- **正向表**
  - 对邻接矩阵的行压缩
  - 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
  - 当对G的结点与边进行编号后，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

# 图的代数表示(12)

- **正向表**




$$
\begin{array}{c}
 \\ v_1 \\ v_2 \\ v_3 \\ v_4
\end{array}
\begin{array}{cccc}
v_1 & v_2 & v_3 & v_4 \\
\end{array}
$$

$$
\begin{array}{c|cccc}
 & v_1 & v_2 & v_3 & v_4 \\
\hline
v_1 & 0 & 1 & 1 & 1 \\
v_2 & 0 & 0 & 1 & 0 \\
v_3 & 0 & 0 & 0 & 0 \\
v_4 & 0 & 0 & 1 & 1
\end{array}
$$

是$V_4$吗？

A: 位置指针

| $v_1$ | $v_2$ | $v_3$ | $v_4$ | $v_{end}$ |
|---|---|---|---|---|
| 1 | 4 | 5 | 5 | 7 |

B: 后继节点

| | | | | | |
|---|---|---|---|---|---|
| 2 | 3 | 4 | 3 | 3 | 4 |

**正向表，存在如下关系：**

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

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

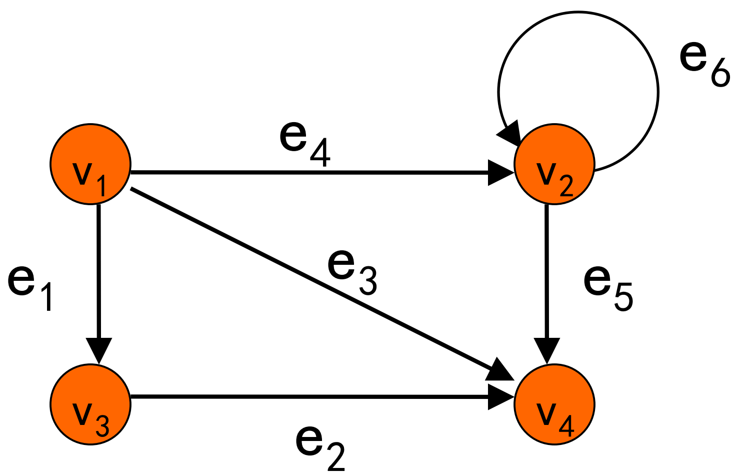
从B(A(i))到B(A(i+1)−1)的任一个值，都是$v_i$的直接后继。

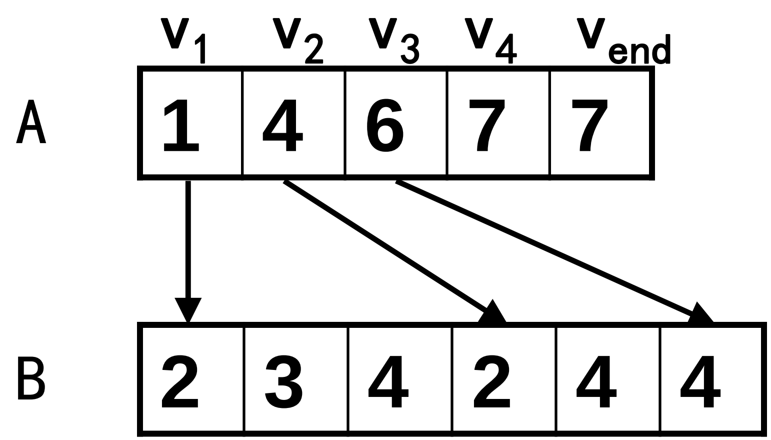
**对赋权图：**

用m维向量Z存放权$Z(k)=w_k$

# 图的代数表示(13)

- **无向图的正向表**
  - 对于无向图，由于边没有方向性，所以B中存放的是相应邻结点的编号，因此B与Z都要扩充为2m维的向量。

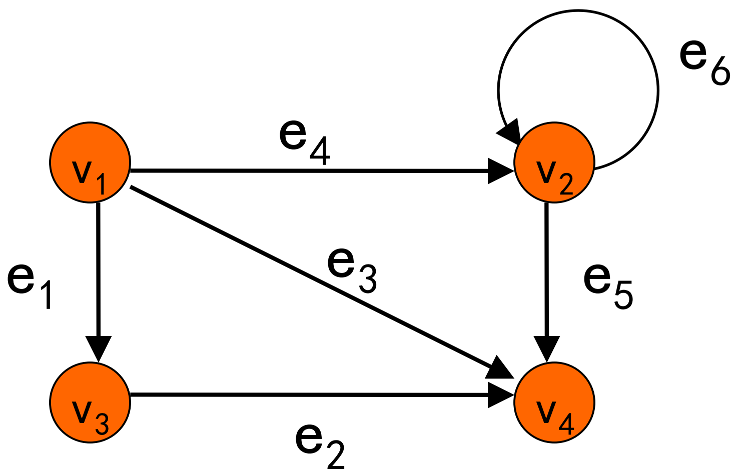


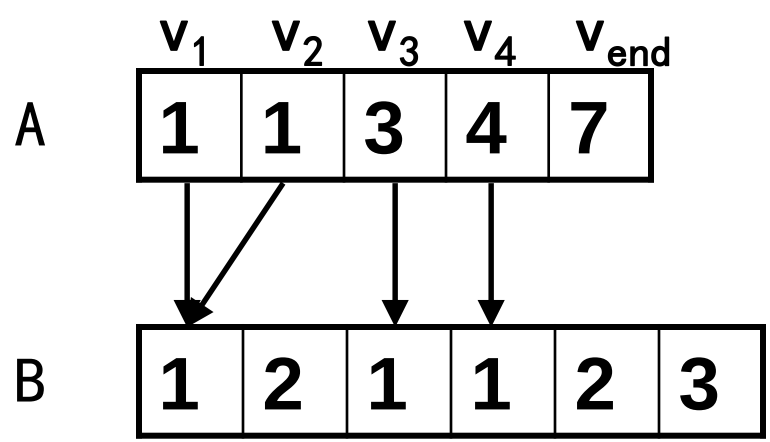




有了正向表（出度/后继），还应该有……

# 图的代数表示(14)

- **逆向表**
  - 将每个结点的直接前趋集中在一起存放。




**优缺点？ 对于图的动态变化缺少灵活性：去掉$e_4$？**
**如何提高灵活性以适应图的动态变化？加减边？**

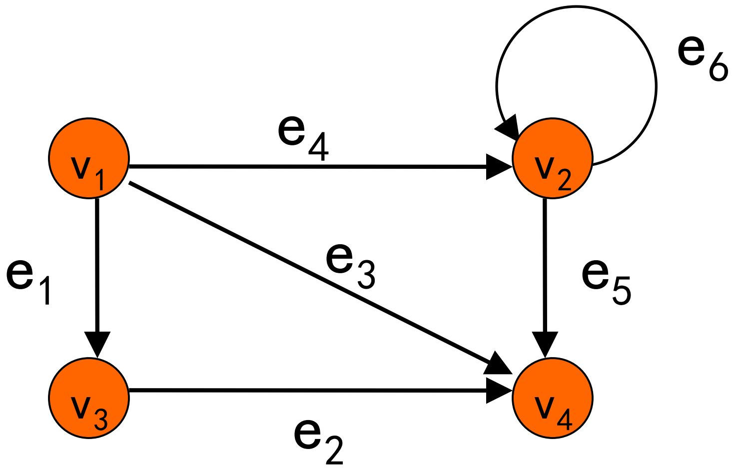
# 图的代数表示(15)

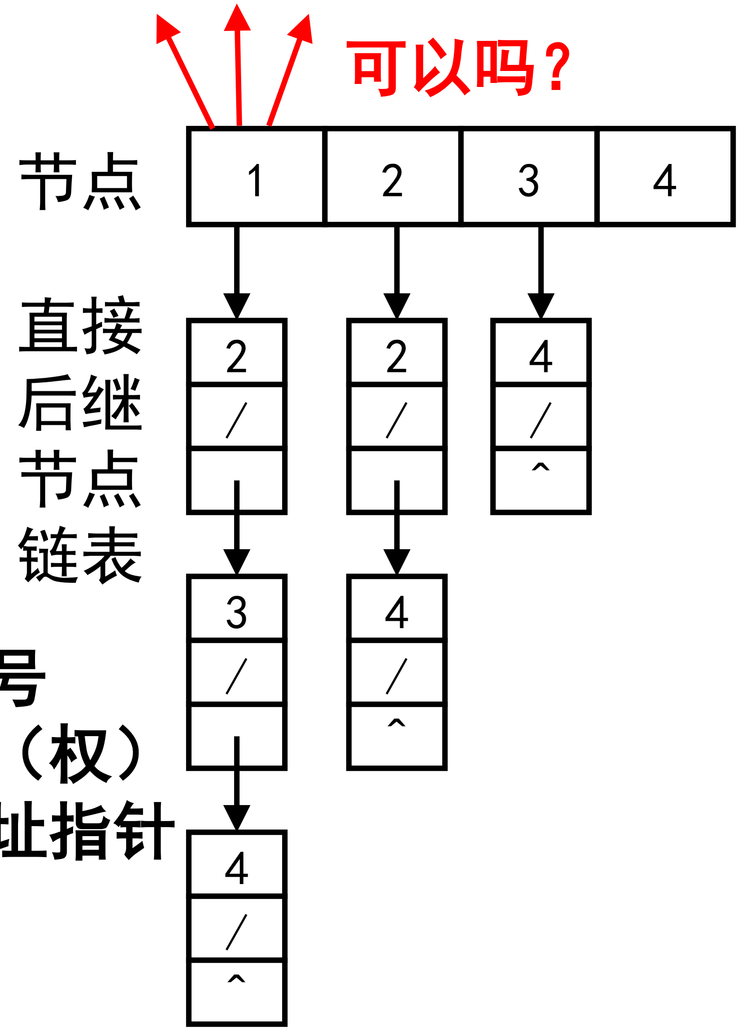
## 邻接表

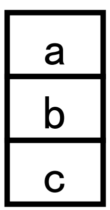
- 采用单链表结构表示一个图
- 对每个结点$v_i$用一个表结点表示
- 表结点由三个域a、b、c组成
  - 邻结点域a中存放邻结点的编号
  - 数据域b中存放相应边的数值（权）
  - 链域c存放下一个表结点的地址指针

# 图的代数表示(16)

◆ 邻接表举例




- 邻结点域a中存放邻结点的编号
- 数据域b中存放相应边的数值（权）
- 链域c存放下一个表结点的地址指针

如何进一步提高灵活性？

# 图的代数表示(17)

- **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
  - 邻接表
- **思考题**
  - 各自的特点
  - 重边、自环、空间、处理方法、相互转换

**Questions?**