面向计算机科学的离散数学

图论—树

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

图的分类

- 非连通图
- 连通图
  - 回路：欧拉回路、H回路、旅行商、邮路

没有回路是什么情况？
如何深入探讨？

- 树：从一串串定义开始……

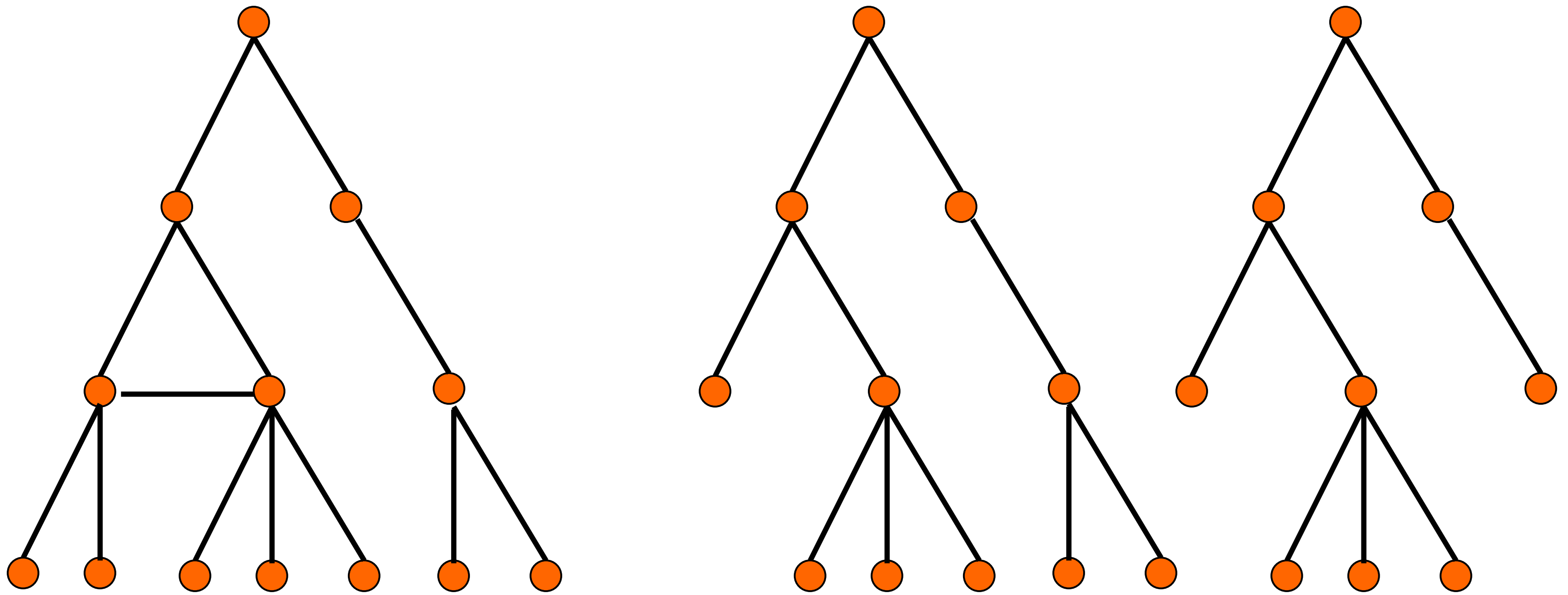
存在性，唯一性？数数看？

第三章 树

- 树的有关定义
- 基本关联矩阵及其性质
- 支撑树的计数

树的有关定义

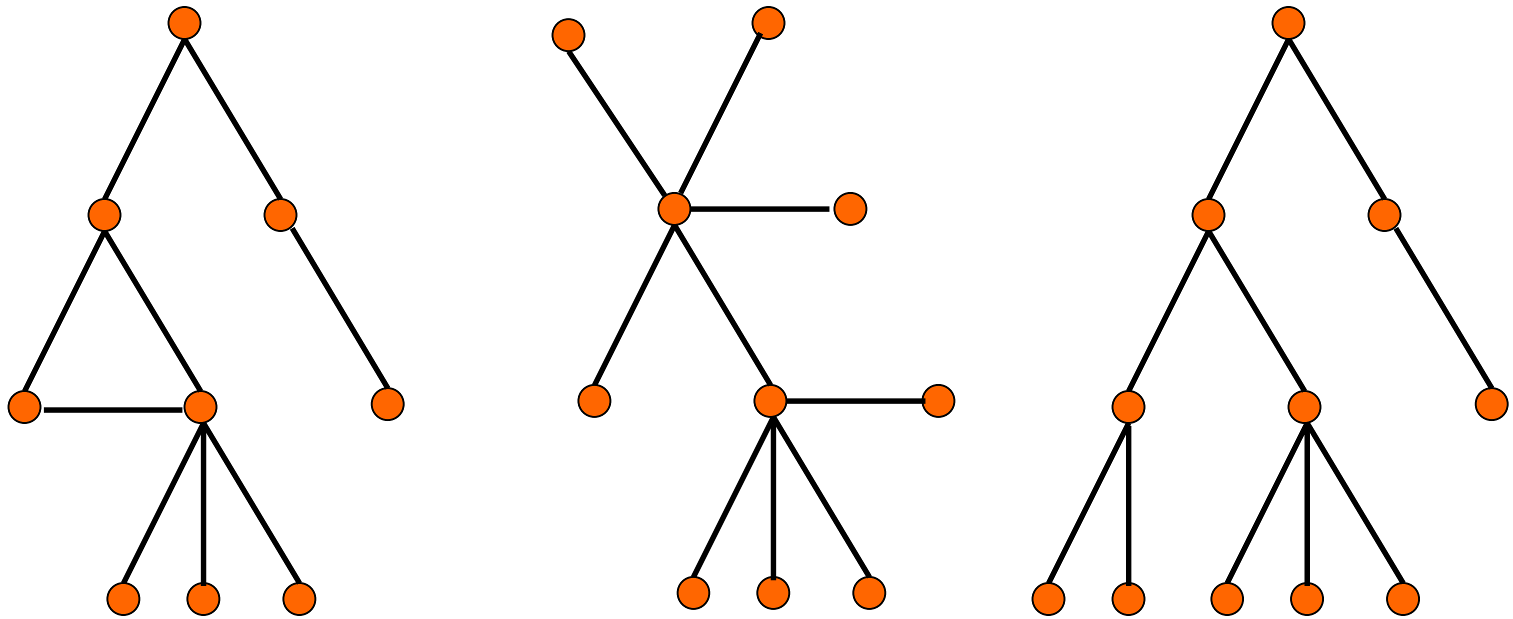
- **什么是树？**
  - 一个图G=(V,E)，若不含任何回路，则称为林;若此图是连通的，则称为树



树的有关定义(2)

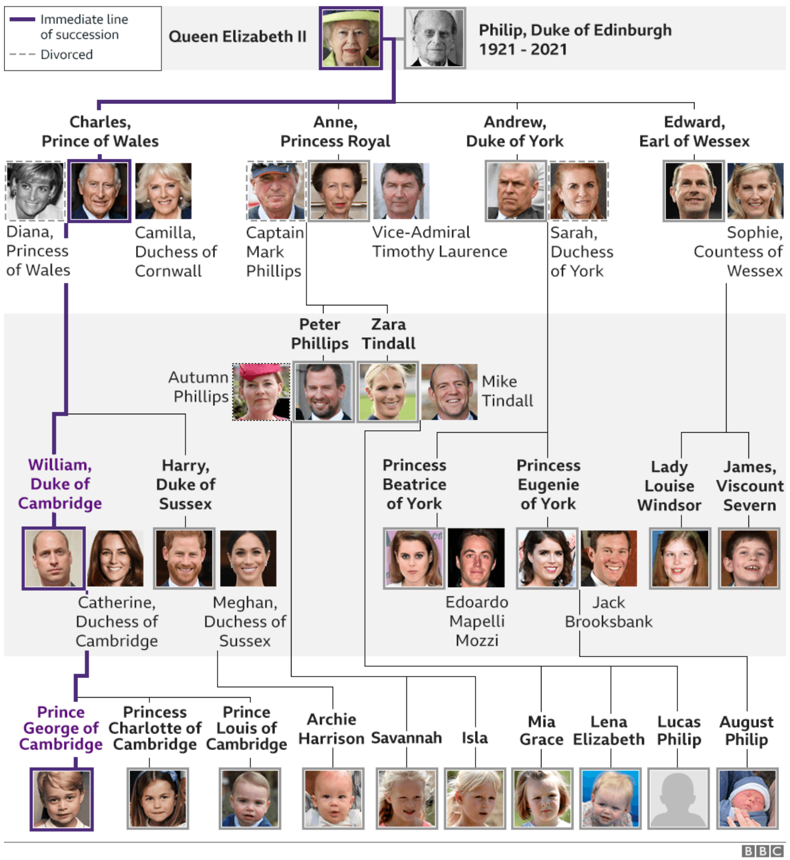
特殊情况？

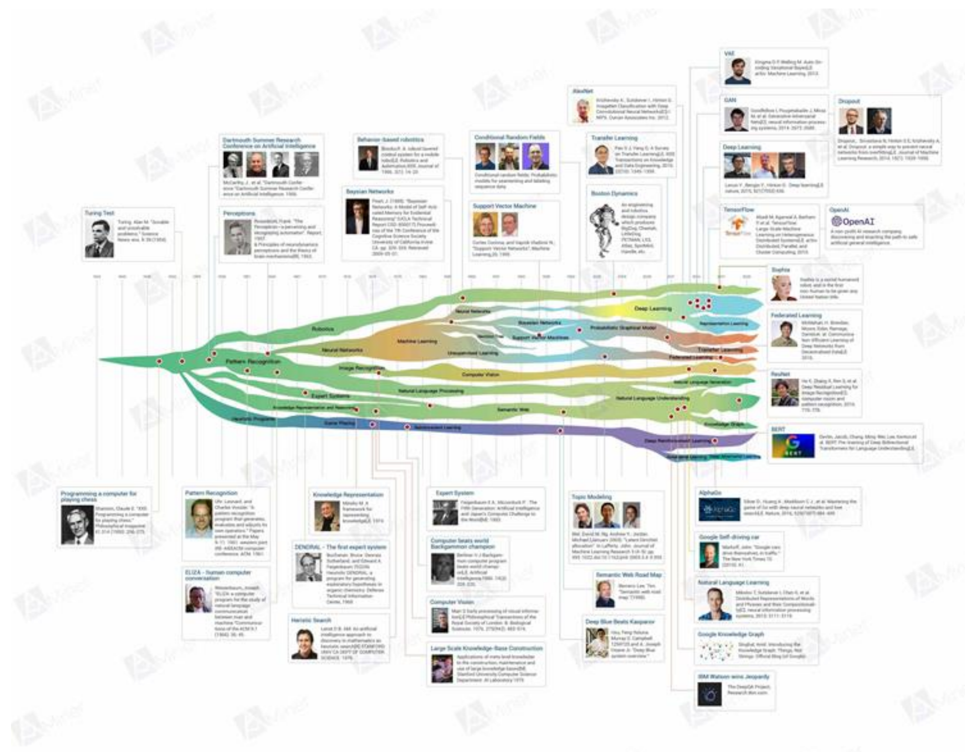
- **定义3.1.1　　树**
  - 一个不含任何回路的连通图称为树，用T表示
  - T中的边称为树枝，度为1的结点称为树叶



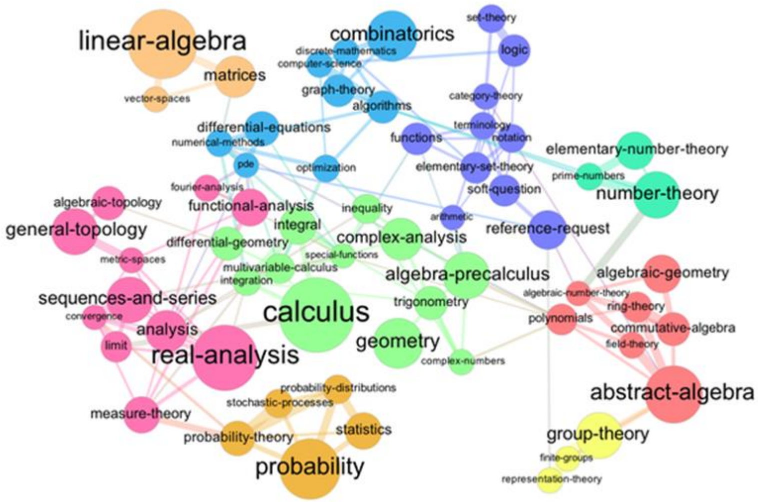
上图是否是树？

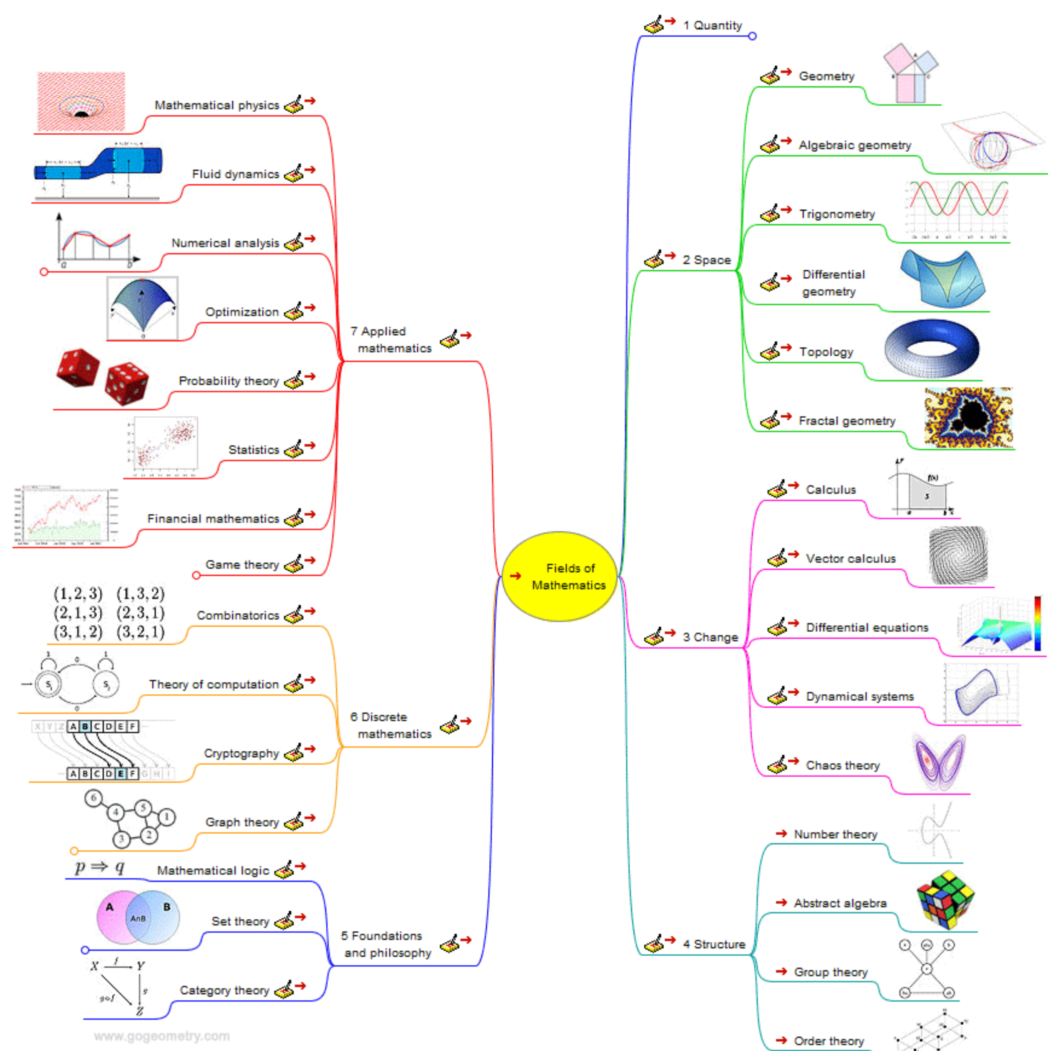
树是一种广泛使用的结构表达方式





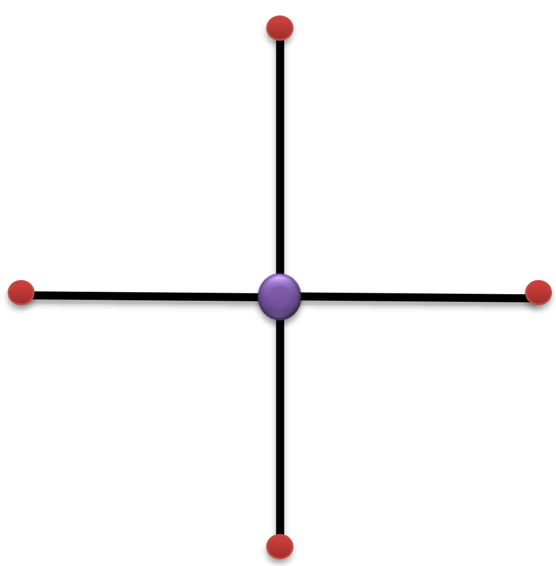
树是一种广泛使用的结构表达方式

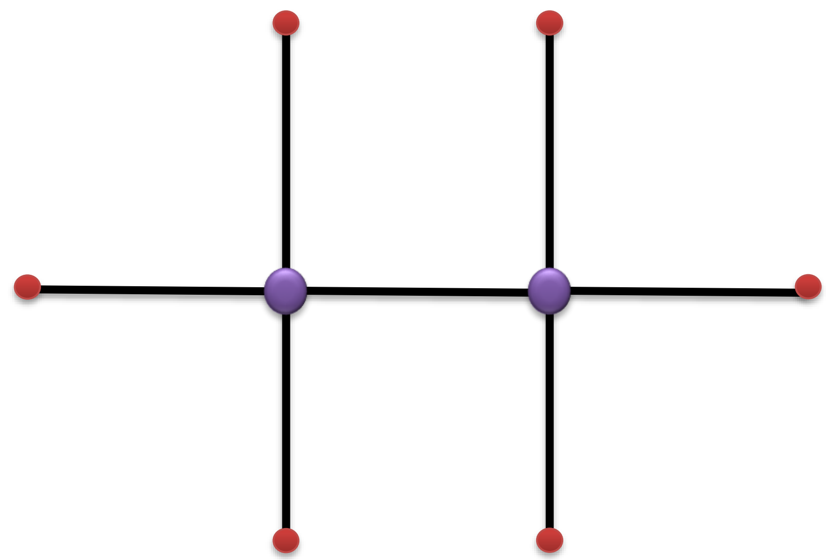




有机化学

- 1857年，凯莱研究碳氢化合物，并特别研究了饱和碳氢化合物
- 通过对这些图的数学分析，预言了新的饱和碳氢化合物





电路网络

- **古斯塔夫·罗伯特·基尔霍夫：德国物理学家，在电路、光谱学的基本原理有重要贡献**
  - 在多个领域都留下了以自己名字命名的定律（定理），其中包括著名的基尔霍夫电路定律（基尔霍夫电压定律、基尔霍夫电流定律）
- **1847年，基尔霍夫在关于电路网络的分析中首次用到了树**

石油管线铺设问题

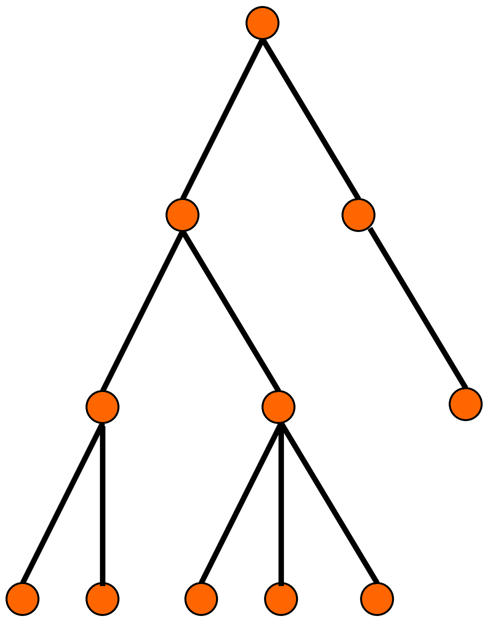
- 要为一个地区铺设石油管线
- 考虑不同的贮藏设施，以便能从一个贮藏设施向其它设施传输石油
- 考虑成本等因素希望建造尽量少的管道
- 寻找铺设的方案

树的有关定义(2)

计算机网络

从拓扑图到基于最短路径树的转发，潜在问题？

- 定义3.1.2　　割边（割边是关键路径）
  - 设e是图G的一条边，若G’=G−e比G的连通支数目增加，称e是G的一条割边
  - 树枝都是割边么？
  - 思考：什么样的边是割边呢？



存在性？判定？

树的有关定义(4)

- 定理 3.1.1
  - e=(u, v)是割边，当且仅当e不属于G的任何回路
  - 证明(反证法)：

必要性

- 若e=(u, v)属于G的某个回路，则G’=G-e中仍存在u到v的道路

故结点u和v属于G’中的同一连通支，即连通支数目没有增加，根据割边定义可知e不是割边，与已知矛盾

充分性

反之，若e不是割边， 则G’与G的连通支数一样

于是G’中u和v仍属同一连通支，即G’中存在道路p(u, v)

- p(u, v)+e就是G的一个回路

树的有关定义(5)

- **定理3.1.2 （树的等价定义）**
  - 对于n ≥ 2的图T(树)，下列性质等价：

1. T连通无回路
2. T连通且每条边都是割边
3. T连通且有n-1条边
4. T有n-1条边且无回路
5. T的任意两结点间有唯一道路
6. T无回路，但在任两结点间加上一条边后恰有一个回路

如何证明等价性？

树的有关定义(6)

- **证明**
  - 只需要证明
    - 1 →2， 2→3， 3→4， 4→5， 5→6， 6→1
    - 1 →2 （T连通无回路→T连通且每条边都是割边）
    - T无回路，即T的任意边e都不属于回路
    - 由定理3.1.1(e=(u，v)是割边，当且仅当e不属于G的任何回路)，e是割边

树的有关定义(7)

证明（续）

- 2 →3（T连通且每条边都是割边→T连通且有n-1条边）
- 对结点数n进行归纳
- 令n(T)，m(T)分别表示树T的结点数与边数
- 当n=2时命题成立
- 设n≤k时m(T)= n(T)-1成立
- 则n=k+1时，由于任一边e都是割边
- **故G' =G-e有两个连通支T_1和T_2，即n(T)=n(T_1)+n(T_2)**
- 由于n(T_i) ≤k，i=1,2,故m(T_i)= n(T_i)-1
- 所以m(T)= m(T_1)+m(T_2)+1= n(T_1)-1 + n(T_2)-1 + 1 =n(T)-1也成立

树的有关定义(8)

- **证明（续）**
  - 3→4(T连通且有n-1条边→T有n-1条边且无回路)
  - （反证法：假定图T有回路，进而考虑在C的基础上构造T所需要使用的最少边数）
  - 设C是其中一条含有k(<n)个结点的初级回路，即E(C)=k
  - 因为T连通，所以V(T)-V(C)中一定有结点u与C上某个结点v相邻即存在边(u,v)∈E(T)
  - 最终要连接 V(T)-V(C)中的n-k个结点，至少还需要n-k条边，才可能保持T连通
  - 因此边数至少为k+(n-k)=n，与有(n-1)条边矛盾

无回路的简单连通图，最大边数=n-1

树的有关定义(9)

- **证明（续）**
  - 4→5（T有n-1条边且无回路→T的任意两结点间有唯一道路）
  - 先证道路P(u, v)的存在性（反证法）
    - 设u, v是T的任意两结点
    - 如果不存在 P(u, v)，则u, v属于不同连通支T_1, T_2
    - 由于m(T)=n-1
    - 则至少有一支，比如T_1，使$n(T_1) \leqslant m(T_1)$成立
    - 这样T_1中有回路，矛盾，所以存在P(u, v)
  - 再证唯一性（对称差）：
    - 若存在两条不同的道路P(u, v)和 P'（u, v)，则其对称差P(u, v)$\oplus$P'（u, v)至少含有一个回路，故而得证

树的有关定义(10)

- **证（续）**
  - 5→6(T的任意两结点间有唯一道路→T无回路，但在任两结点间加上一条边后恰有一个回路)
  - 显然
  - 6→1(T无回路，但在任两结点间加上一条边后恰有一个回路→T连通无回路)
  - 显然

- 因此等价定理得证

树的有关定义(11)

- 定理3.1.2 （树的等价定义）
  - 对于n ≥ 2的图T(树)，下列性质等价：

1. T连通无回路
2. T连通且每条边都是割边
3. T连通且有n-1条边
4. T有n-1条边且无回路
5. T的任意两结点间有唯一道路
6. T无回路，但在任两结点间加上一条边后恰有一个回路

其它证明路径？

特殊情况：
割边？树叶？

树的有关定义(12)

- 定理3.1.3
  - 树T一定存在叶结点
  - 证明（基本方法？）
    - 反证法
    - 由于T是连通图，所以任意结点v_i，有$d(v_i) \geqslant 1$
    - 若无树叶，则有$d(v_i) \geqslant 2$
    - 这样，有
    - $$n-1=m=\frac{1}{2}\sum d(v_i)\ge n$$
    - 得出矛盾，因此T一定存在叶结点

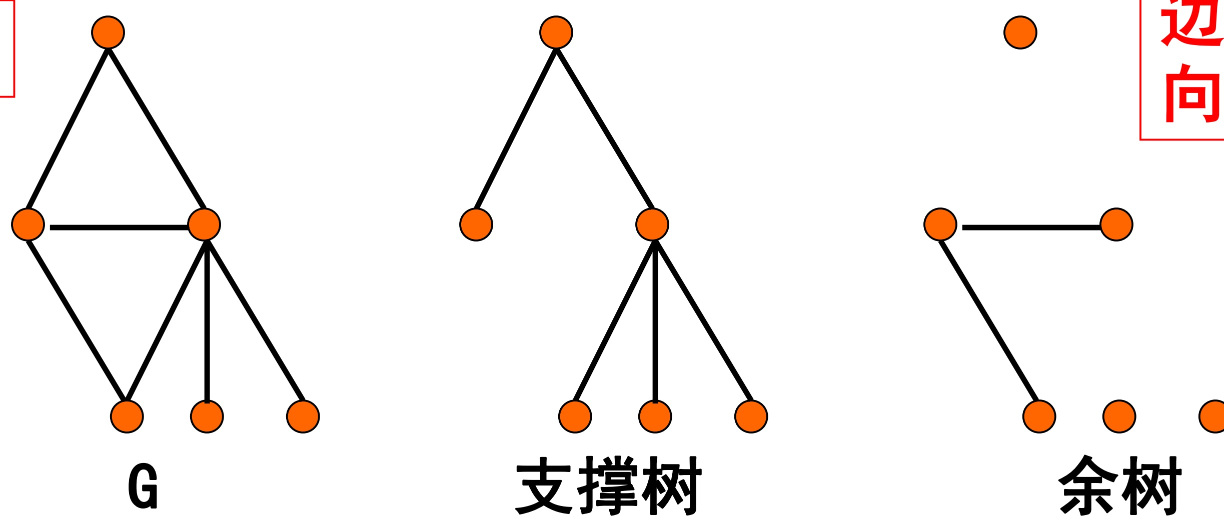
树的有关定义(13)

- **定义3.1.3　支撑树，生成树**
  - 如果T是图G的支撑子图，而且又是一棵树，则称T是G的一棵支撑树，或生成树，简称为G的树
- **余树**
  - 设T是G的支撑树，则称G-T为G的余树

有向图不考虑边的方向

牛定义后呢？

存在？
唯一？
多少个？
算算看？



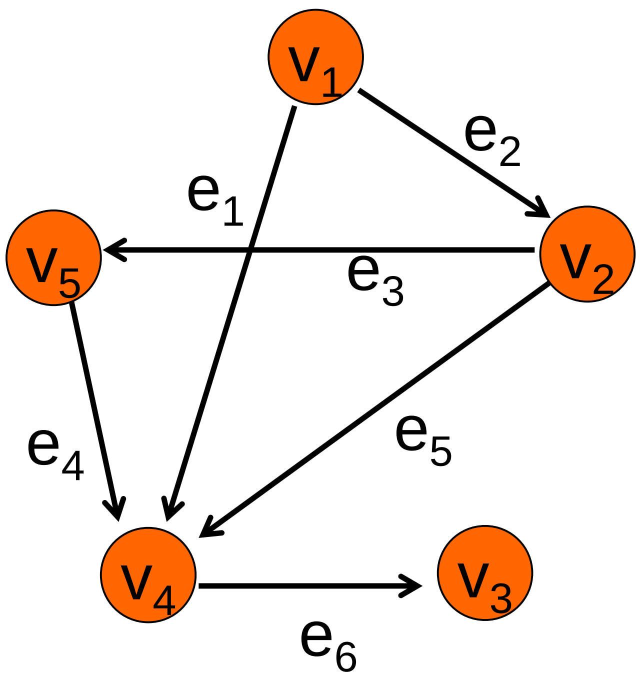
第三章 树

- 树的有关定义
- 基本关联矩阵及其性质
- 支撑树的计数

基本关联矩阵及其性质(1)

- **回顾关联矩阵（代数表示！！）**
  - 点和边的关联关系
  - 每行都是有价值的信息吗？

呼唤个定理？



$$
\begin{array}{c}
\\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5
\end{array}
\begin{array}{c}
\begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 \\
-1 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & -1 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

关联矩阵B

基本关联矩阵及其性质(2)

- **定理3.2.1**
  - 有向图G=(V,E)关联矩阵B的秩ran B < n
  - 证明：
    - B中每列都只有1和-1两个非0元素
    - 因此B的任意n-1行加到第n行后，第n行为全0
    - 即B的n个行向量线性相关，ranB<n

$$
\begin{array}{c}
\\
v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5
\end{array}
\begin{array}{c}
\begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 \\
-1 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & -1 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

少一行是不是就线性无关了呢？

基本关联矩阵及其性质(3)

- **定理3.2.3**
  - 有向连通图G=(V,E)关联矩阵B的秩ran B＝n-1
  - 证明（重点）
    - 由定理3.2.1知ranB<n,现只需证ranB≥n-1
    - 不失一般性，设B中最少的线性相关的行数为l
    - 显然l≤n
    - 设这l行分别与结点$v(i_1), v(i_2), \cdots, v(i_l)$相对应
    - 因此有

    $k_1 b(i_1)+k_2 b(i_2)+\cdots+k_l b(i_l)=0$, $k_j \neq 0$, $j=1,2,\cdots l$

    其中b(i)为节点i对应的行向量

基本关联矩阵及其性质(4

$$
\begin{array}{c|cccccc}
 & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\
\hline
v_1 & 1 & 1 & 0 & 0 & 0 & 0 \\
v_2 & 0 & -1 & 1 & 0 & 1 & 0 \\
v_3 & 0 & 0 & 0 & 0 & 0 & -1 \\
v_4 & -1 & 0 & 0 & -1 & -1 & 1 \\
v_5 & 0 & 0 & -1 & 1 & 0 & 0
\end{array}
$$

B中最少的
线性相关的行数为l

- **定理3. 2. 3（证明续）**
  - 由于矩阵B每列只有2个非零元
  - 所以在这l个行向量$b(i_j)$中，其第t(t=1, 2, …, m)个分量最多只有2个为非零元（当然也可能全为零）
  - 但是可以断言：不可能只有一个为非零
    - 否则因任意$k_j \neq 0$，式(1)不会成立
    - $k_1 b(i_1)+k_2 b(i_2)+\cdots+k_l b(i_l)=0$，$k_j \neq 0$，$j=1, 2, \cdots l$ (1)
  - 对矩阵B“行列变换”，使前l行是线性相关的诸行
  - 这在l行中每列都有2个非零元的换到前r列，其余m-r列它们全都是零元这样矩阵B变换为：

$$B' = \begin{array}{cc} \left[\begin{array}{cc} P & 0 \\ 0 & Q \end{array}\right] & \begin{array}{c} l \\ n\text{-}l \end{array} \\ \begin{array}{cc} r & m\text{-}r \end{array} & \end{array}$$

基本关联矩阵及其性质(5)

- **定理3.2.3（证明续）**
  - 但ranB’=ranB，且B’依然是G的一个关联矩阵，与B相比只是结点与边的编号不同而已

$$\begin{array}{ccc} & \begin{array}{cc} & \end{array} & \\ B' = & \begin{bmatrix} P & 0 \\ 0 & Q \end{bmatrix} & \begin{array}{c} l \\ n\text{-}l \end{array} \\ & \begin{array}{cc} r & m\text{-}r \end{array} & \end{array}$$

  - 若n-l>0，由B’可见，G至少分为2个连通支
    - 其中r条边只与l个结点相关，而其余m-r条边只与n-l个结点相关

B中最少的线性相关的行数为l

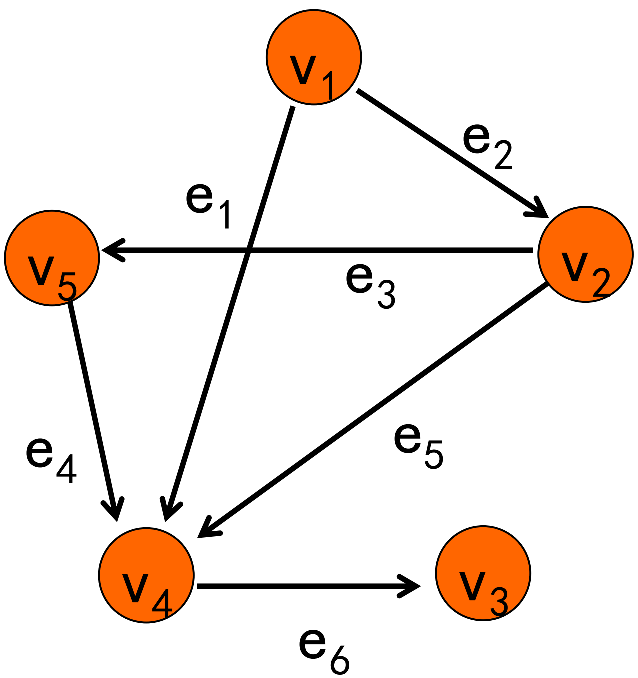
  - 这与G是连通图矛盾！
  - 因此一定有n-l=0，即l=n
  - 即B中最少需要n行才能线性相关
  - 而任何n-1行都将线性无关，即ranB≥n-1
  - 所以ranB=n-1，证毕

物理意义是什么？

然后呢？

基本关联矩阵及其性质(6)

- **定义3.2.1　基本关联矩阵**
  - 在有向连通图G=(V,E)的关联矩阵B中划去任意结点v_k所对应的行，得到一个(n-1)×m的矩阵B_k，称为G的一个基本关联矩阵



$$
\begin{array}{c}
 \\ v1 \\ v2 \\ v4 \\ v5
\end{array}
\begin{array}{c}
\begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 1 & 0 \\
-1 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & -1 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

基本关联矩阵B_3

基本关联矩阵及其性质(7)

- **定理3.2.3：有向连通图G=(V,E)关联矩阵B的秩ran B＝n-1**
- **定理3.2.4**
  - 有向连通图G=(V,E)的基本关联矩阵B_k的秩ran B_k＝n-1
  - 证：由以上定理3.2.3的证明过程可知，任意n-1个行向量是线性无关的，可得此结论

$$\begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v_1 & 1 & 1 & 0 & 0 & 0 & 0 \\ v_2 & 0 & -1 & 1 & 0 & 1 & 0 \\ v_3 & 0 & 0 & 0 & 0 & 0 & -1 \\ v_4 & -1 & 0 & 0 & -1 & -1 & 1 \\ v_5 & 0 & 0 & -1 & 1 & 0 & 0 \end{array}$$

关联矩阵B

行向量？

$$\begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v1 & 1 & 1 & 0 & 0 & 0 & 0 \\ v2 & 0 & -1 & 1 & 0 & 1 & 0 \\ v4 & -1 & 0 & 0 & -1 & -1 & 1 \\ v5 & 0 & 0 & -1 & 1 & 0 & 0 \end{array}$$

基本关联矩阵B_3

秩不变的边数最少什么情况？

基本关联矩阵及其性质(8)

- 思考-行列之间的关系？
  - 连通图基本关联矩阵B_k的秩是n-1， B_k中一定存在n-1个线性无关的列(对连通图有m>=n-1)
  - 哪些列线性无关的、哪些列线性相关？
- **推论3.2.1-树T（特殊的连通图）**
  - n个结点的树T的基本关联矩阵(方阵)的秩是n-1.

$$
\begin{array}{c|cccccc}
 & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\
\hline
v1 & 1 & 1 & 0 & 0 & 0 & 0 \\
v2 & 0 & -1 & 1 & 0 & 1 & 0 \\
v4 & -1 & 0 & 0 & -1 & -1 & 1 \\
v5 & 0 & 0 & -1 & 1 & 0 & 0
\end{array}
$$

思考
树枝边构成线性无关的列？
那线性相关的列是什么呢？

基本关联矩阵及其性质(9)

◈ **定理3.2.5**

- 设B_k是连通图G的基本关联矩阵，C是G中的一个回路，则C中各边所对应B_k的各列线性相关。
- 证明：（针对C是初级回路讨论）
  - 设C包含了G的l个结点l条边（不妨设l<n）
  - 这l条边对应关联矩阵B的l列，它们构成了B的子阵$B(G_C)$
  - C本身是含l个点l条边的连通子图，所以B(C)是l阶方阵
  - 而ranB(C)=l−1，故B(C)的l列线性相关，且是$B(G_C)$的子阵
  - 由于$B(G_C)$对应的各边只经过回路C的结点，而与其他结点无关，因此$B(G_C)$中其余结点所对应的行元素全为零
  - 因此$B(G_C)$中这l列仍是线性相关
  - 显然$B_k(G_C)$的这l列也线性相关

$$
\begin{array}{c}
\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\left[
\begin{array}{cccccc}
e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\
1 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 \\
-1 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & -1 & 1 & 0 & 0
\end{array}
\right]
\end{array}
$$

B(C)　$B(G_C)$

证明思路：ranB(C)，B(C)列线性相关，$B(G_C)$，$B_k(G_C)$

基本关联矩阵及其性质(10)

$$\begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v1 & 1 & 1 & 0 & 0 & 0 & 0 \\ v2 & 0 & -1 & 1 & 0 & 1 & 0 \\ v4 & -1 & 0 & 0 & -1 & -1 & 1 \\ v5 & 0 & 0 & -1 & 1 & 0 & 0 \end{array}$$

- **定理3.2.5的理解**
  - B_k中回路的各列线性相关
- **推论3.2.2**
  - 设H是连通图G的子图，如果H含有回路，则H的诸边对应的G的基本关联矩阵各列线性相关
- **定理3.2.6**
  - 令B_k是有向连通图G的基本关联矩阵，那么B_k的任意n-1阶子阵行列式非零的充要条件是其各列所对应的边构成G的一棵支撑树

思考：线性无关的列？最多是什么？

回顾：行列式

- **定义：** 由 n^2 个数 a_{ij} 组成如下数阵称为 n 阶行列式，其中 $D_1=|a_{11}|=a_{11}$

$$D_n=\begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots \\ a_{n1} & \cdots & a_{nn} \end{vmatrix}$$

行列向量
线性相关
矩阵的秩

- **行列式计算**
  - 在 n 阶行列式中, 把元素 a_{ij} 所在的第 i 行和第 j 列划去后, 留下来的 $n-1$ 阶行列式叫做元素 a_{ij} 的余子式，记作 M_{ij} ;元素 a_{ij} 的余子式为：

$$A_{ij}=(-1)^{i+j}M_{ij}$$

行列式

- 二阶行列式计算举例

$$D=\begin{vmatrix}1 & 2\\ 3 & 4\end{vmatrix}=1\times(-1)^{1+1}|4|+2\times(-1)^{1+2}|3|=1\times4-2\times3=-2$$

- 三阶行列式计算举例

$$\begin{vmatrix}1 & 3 & 2\\ 1 & 0 & 2\\ 0 & 1 & 3\end{vmatrix}=1\times\begin{vmatrix}0 & 2\\ 1 & 3\end{vmatrix}-1\times\begin{vmatrix}3 & 2\\ 1 & 3\end{vmatrix}=-2-7=-9$$

$$\begin{vmatrix}1 & 1 & 2\\ 0 & 3 & 2\\ 0 & 0 & 3\end{vmatrix}=1\times\begin{vmatrix}3 & 2\\ 0 & 3\end{vmatrix}=1\times3\times3=9$$

$$\begin{vmatrix}1 & 1 & 2\\ 3 & 3 & 2\\ 0 & 0 & 0\end{vmatrix}=?$$

基本关联矩阵及其性质(11)

$$\begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v1 & 1 & 1 & 0 & 0 & 0 & 0 \\ v2 & 0 & -1 & 1 & 0 & 1 & 0 \\ v4 & -1 & 0 & 0 & -1 & -1 & 1 \\ v5 & 0 & 0 & -1 & 1 & 0 & 0 \end{array}$$

证明（必要性）：

- 要证：B_k的n-1阶子阵行列式非零，则各列/边构成G的支撑树
- 如果某个n-1阶子阵$B_k(G_T)$的行列式非零
- 则由推论3.2.2(如果H含有回路，则H的诸边对应的G的基本关联矩阵各列线性相关)，T中不含回路
- 因为$B_k(G_T)$是基本关联矩阵的(n-1)阶子阵，所以其对应的图包含n个结点、n-1条边
- 根据定理3.1.2的等价定义4(T有n-1条边且无回路)，T是G的一棵树

基本关联矩阵及其性质(12)

- 证明（充分性）
  - 要证：支撑树=>B_k的对应n-1阶子阵行列式非零
  - 设T是G的一棵树，包含n个结点，n-1条边

$$
\begin{array}{c|cccccc}
 & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\
\hline
v1 & 1 & 1 & 0 & 0 & 0 & 0 \\
v2 & 0 & -1 & 1 & 0 & 1 & 0 \\
v4 & -1 & 0 & 0 & -1 & -1 & 1 \\
v5 & 0 & 0 & -1 & 1 & 0 & 0
\end{array}
$$

  - 子图T的基本关联矩阵B_k(T)是n-1阶方阵，其秩 ran B_k(T) ＝ n-1，即满秩，所以行列式非零

    (定理3.2.4：有向连通图G基本关联矩的秩ran B_k＝n-1)

  - 因为T是G的子图，B_k(T)恰好对应B_k的某个n-1阶子阵（即使用了T的n-1条边）
  - 即B_k对应的该n-1阶行列式非零

基本关联矩阵及其性质(11)

- **回顾：定理3.2.6**
  - 令B_k是有向连通图G的基本关联矩阵，那么B_k的任意n-1阶子阵行列式非零的充要条件是其各列所对应的边构成G的一棵支撑树

$$\begin{array}{c} \\ v1 \\ v2 \\ v4 \\ v5 \end{array}\begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 \\ -1 & 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

- **思考？**
  - 行列式非零的n-1阶子阵与支撑树一一对应吗？
  - 如何计算支撑树的数目？
  - 图G的基本关联矩阵B_k中，行列式非零的n-1阶子阵数目与G不同的支撑树数目，对应吗？

基本关联矩阵

你能发明出来吗？

- **行向量**
  - 关联矩阵B为(n*m)，行向量相关，ran(B)=n-1
  - 基本关联矩阵B_k为(n-1)*m， ran(B_k)=n-1
  - B和B_k中(n-1)行向量线性无关
- **列向量**

横看成岭侧成峰，排列组合求创新

  - 设H是连通图G的子图，如果H含有回路，则H的诸边对应的G的基本关联矩阵各列线性相关
  - B_k的任意n-1阶子阵行列式非零的充要条件是其各列所对应的边构成G的一棵支撑树（det(B_0)=0/1/-1）
  - B_k中行列式非零的n-1阶子阵的数目与G不同的支撑树数目之间一一对应

第三章 树

- 树的有关定义
- 基本关联矩阵及其性质
- 支撑树的计数

基本关联矩阵及其性质(12)

- **再看看B**
  - 欲求数目（最大非零行列式）
  - 必先理解（每个行列式）

子阵的

行列式呢？

$$
\begin{array}{c}
\begin{array}{cccccc} \quad e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\
\begin{array}{c} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 \\
-1 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & -1 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

基本关联矩阵及其性质(13)

$$
\begin{array}{c|cccccc|}
 & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\
v_1 & 1 & 1 & 0 & 0 & 0 & 0 \\
v_2 & 0 & -1 & 1 & 0 & 1 & 0 \\
v_3 & 0 & 0 & 0 & 0 & 0 & -1 \\
v_4 & -1 & 0 & 0 & -1 & -1 & 1 \\
v_5 & 0 & 0 & -1 & 1 & 0 & 0
\end{array}
$$

- **定理3.2.2**
- **设B_0是有向图G=(V,E)关联矩阵B的k阶方阵，则det(B_0)=0,1,−1**
- **证明：**
  - 因为B_0是B的某一k阶子阵，B_0每列最多只有2个非零元
  - 若其中某列全为零元，则det(B_0)=0
  - 若B_0每列都有2个非零元，则线性相关，有det(B_0)=0
  - 否则B_0中存在某列只有一个非零元
    - 按该列展开得到det(B_0)={±det(B_1)
    - 但B_1 的阶为k−1
    - 依次类推，可知最终det(B_0)为0，1，或−1

支撑树的计数

$$B_k = \begin{array}{c} \\ v1 \\ v2 \\ v4 \\ v5 \end{array}\begin{array}{c}\begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & 0 \\ -1 & 0 & 0 & -1 & -1 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix}\end{array}$$

支撑树数目：
有多少个n-1
阶非零行列式？

- **定理3.3.1（Binet-Cauchy定理）**
  - 已知两个矩阵$A=(a_{ij})_{m\times n}$和$B=(b_{ij})_{n\times m}$满足$m \leqslant n$，则 $\det(AB)=\sum A_iB_i$
  - A_i:A中不同的m列构成的行列式
  - B_i:B中相应的m行构成的行列式

组合累加：子阵行列式的乘积

支撑树的计数(2)

- **例**
  - 已知矩阵A和B如下，求det(AB)

$$A=\begin{bmatrix} 4 & 3 & 2 \\ -2 & 4 & 3 \end{bmatrix}, B=\begin{bmatrix} 5 & 1 \\ 0 & 3 \\ 4 & 2 \end{bmatrix}$$

  - 解
  - 由矩阵乘法，

$$AB=\begin{bmatrix} 28 & 17 \\ 2 & 16 \end{bmatrix}$$

  - 所以det(AB)=414

支撑树的计数(3)

- **例**

$$A=\begin{bmatrix}4 & 3 & 2\\ -2 & 4 & 3\end{bmatrix}, B=\begin{bmatrix}5 & 1\\ 0 & 3\\ 4 & 2\end{bmatrix}$$

  - 由比内-柯西定理计算

$$\det(AB)=\sum_i A_i B_i$$

$$=\begin{vmatrix}4 & 3\\ -2 & 4\end{vmatrix}\begin{vmatrix}5 & 1\\ 0 & 3\end{vmatrix}+\begin{vmatrix}4 & 2\\ -2 & 3\end{vmatrix}\begin{vmatrix}5 & 1\\ 4 & 2\end{vmatrix}+\begin{vmatrix}3 & 2\\ 4 & 3\end{vmatrix}\begin{vmatrix}0 & 3\\ 4 & 2\end{vmatrix}$$

$$=414$$

支撑树的计数(4)

$$\det(AB)=\sum A_iB_i$$

- 例（续）
  - 显然可见，用比内一柯西定理计算乘积矩阵的行列式比通常方法复杂

更复杂？？？

  - 但该定理揭示了乘积矩阵的行列式与各矩阵子行列式之间的关系
  - 连通图G不同支撑树的计数，恰好利用了这种关系

支撑树的计数(5)

$$
\begin{array}{c}
 \\ v1 \\ v2 \\ v4 \\ v5
\end{array}
\begin{bmatrix}
e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\
1 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 1 & 0 \\
-1 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & -1 & 1 & 0 & 0
\end{bmatrix}
$$

- **有向连通图的树计数**
- **定理3.3.2**
  - 设B_k是有向连通图G=(V,E)的某一基本关联矩阵，则G的不同树的树目是$\det(B_k B_k^T)$
  - 证明：设$B_k=(b_{ij})_{(n-1)\times m}$，由于G是连通图，故$n-1 \leqslant m$
    - 由比内-柯西定理，得

$$\det(B_k B_k^T) = \sum_i |B_i||B_i^T|$$

    - 其中$|B_i|$是B_k的某个 n-1阶子阵的行列式
    - $|B_i^T|$是B_i转置的行列式

支撑树的计数(6)

- **证（续）**
  - 由于B_i^T是n-1阶子阵B_i的转置，因此$|B_i^T|=|B_i|$

$$\det(B_k B_k^T)=\sum_i |B_i||B_i^T|=\sum_i |B_i|^2$$

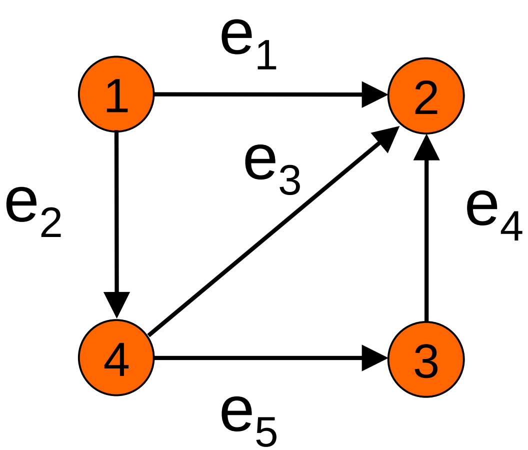
  - 由定理3.2.6，若$|B_i|\neq 0$，则对应边构成G的一棵树
  - 由定理3.2.2，此时$|B_i|$=1或-1，即$|B_i|^2$=1
  - 这说明若B_i的各列对应的边构成G的一棵树，则对$\det(B_k B_k^T)$中的贡献为1
  - 上式是对$|B_i|^2$的全部组合求和
  - 因此$\det(B_k B_k^T)$是G中不同树的数目

$$\begin{array}{c|cccccc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\ \hline v1 & 1 & 1 & 0 & 0 & 0 & 0 \\ v2 & 0 & -1 & 1 & 0 & 1 & 0 \\ v4 & -1 & 0 & 0 & -1 & -1 & 1 \\ v5 & 0 & 0 & -1 & 1 & 0 & 0 \end{array}$$

支撑树的计数(7)

- 例 3. 3. 2
  - 求下图中树的数目（不考虑边的方向）

- **解：任取一个基本关联矩阵，如B_4**



$$B_4 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$\therefore \det(B_4 B_4^T) = \det \begin{bmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 2 \end{bmatrix} = 8$$

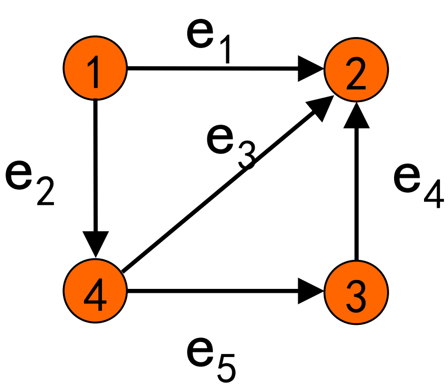
- 数数看?
- **5边去2边：C(5,2)-2=5*4/2-2=8**

支撑树的计数(8)

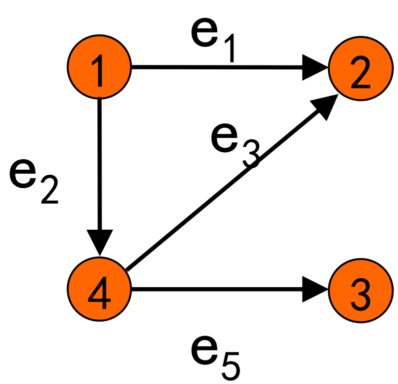
- **不含或必含特定边的树计数**
  - 有向连通图
  - 若不含特定边e
    - 则G’=G-e的树就与之一一对应
  - 若必含特定边e
    - 计算G的树的数目，减去G’=G-e的树的数目
    - 可将e的两个端点收缩成一个点，则得到n-1个结点的新图G’，G’的树与G的必含e的树一一对应

支撑树的计数(9)

- **例3.3.3**
  - 求图中不含e_4的树数目



- **解：作G-e_4，得到下图。**

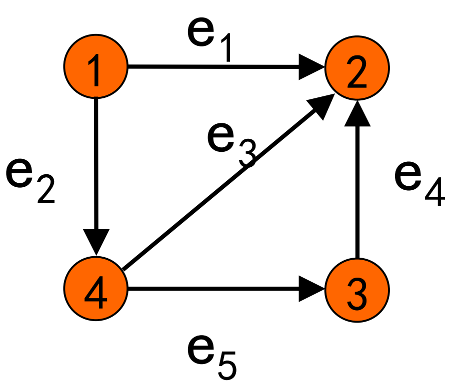


$$B_4 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\therefore \det(B_4 B_4^T) = \det \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = 3$$

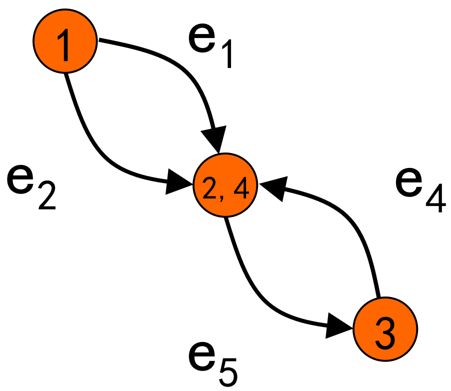
支撑树的计数(10)

- **例3. 3. 4**
  - 求图中必含e_3的树数目



解：将图中v_2，v_4收缩为$v_{2,4}$

$$B_3 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1 & -1 & -1 & 1 \end{bmatrix}$$



$$\therefore \det(B_3 B_3^T) = \det\begin{bmatrix} 2 & -2 \\ -2 & 4 \end{bmatrix} = 4$$

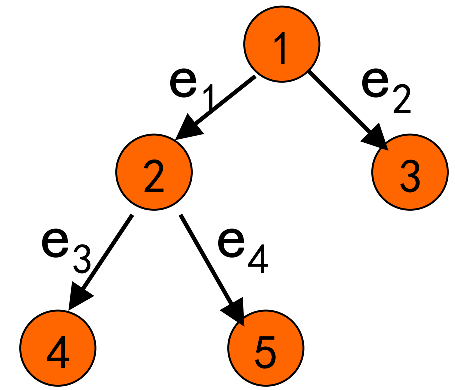
支撑树的计数(11)

- **无向连通图的树计数**
  - 将无向图G的各边加一方向，得有向图G’，G’的树与G的树一一对应
- **例：求完全图K_n中不同树的数目**
  - 对各边任给一方向，得到有向完全图G，G中v_k对应的基本关联矩阵是B_k。可得

$$\therefore \det(B_k B_k^T) = \det\begin{bmatrix} n-1 & -1 & & -1 \\ -1 & n-1 & \ddots & -1 \\ -1 & -1 & & n-1 \end{bmatrix} = n^{n-2}$$

支撑树的计数(12)

- **有向连通图G根树的计数**



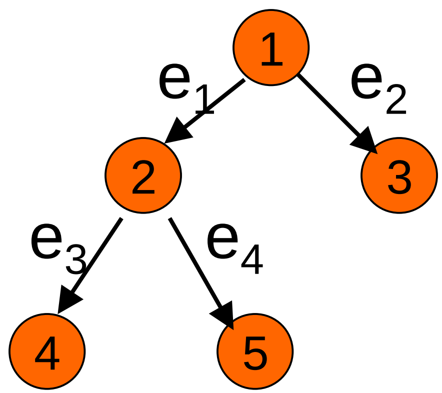
- **定义3.3.1 根树**
  - T是有向树

    存在某结点v_0的负度为0，其余结点负度为1，则称T是以v_0为根的外向树，或称根树T
  - 根树能否从树根沿着正向边走到任意叶子？
    - 考虑关键路径
    - 不断去掉负度为0的点

透过现象看本质
连问三个为什么！
$\det(B_k B_k^T)$？？？

支撑树的计数(13)

- **例**
  - 下图是一棵根树，求根结点基本关联矩阵B_1



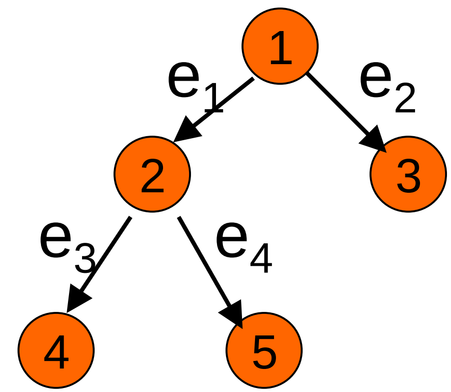
$$\begin{array}{c c} & \begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\ \begin{array}{c} v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} & \begin{bmatrix} -1 & 0 & 1 & 1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \end{array}$$

  - 根结点基本关联矩阵B_1的特点
    - 由于v_1的负度为0，其余结点负度均为1
    - 因此根结点的基本关联矩阵一定是：每行每列只有1个−1元素

支撑树的计数(14)

根树基本关联矩阵的特征

- 若对根树的结点和边序号重新编号
- 使得每条边$e_j=(v_i, v_{j+1})$，且满足$i<j+1$
- 则得到根结点基本关联矩阵B_0'为上三角矩阵，对角元均为-1
- 若把根树基本关联矩阵的所有1均变为零，行列式值不变
- 其他的树呢？
  - 一定存在节点（如根节点）负度为0，修改后出现全0行，即行列式值变为0



$$
\begin{array}{c}
\\ v_2 \\ v_3 \\ v_4 \\ v_5
\end{array}
\begin{array}{c}
\begin{array}{cccc} e_1 & e_2 & e_3 & e_4 \end{array} \\
\begin{bmatrix}
-1 & 0 & 1 & 1 \\
0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1
\end{bmatrix}
\end{array}
$$

支撑树的计数(15)

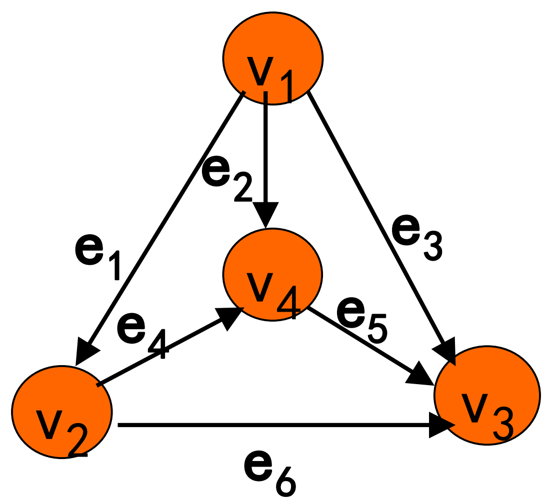
- **定理3.3.3**
  - 设 $\vec{\mathrm{B}}_k$ 表示有向连通图G的基本关联矩阵B_k中将全部1改为0之后的矩阵，则
  - G中以v_k为根的根树数目是 $\det(\vec{\mathrm{B}}_k \mathrm{B}_k^T)$
- **证明**
  - 由比内-柯西定例 $\det(\vec{\mathrm{B}}_k \mathrm{B}_k^T) = \sum_i \left|\vec{\mathrm{B}}_i\right| \left|\mathrm{B}_i^T\right|$
  - 若$|B_i| \neq 0$，说明这n-1条边构成一棵树
  - 此时如果 $\left|\vec{\mathrm{B}}_i\right| \neq 0$ ，说明此树是以v_k为根的根树
  - 此时 $\left|\vec{\mathrm{B}}_i\right| = \left|B_i^T\right|$ ，因此它们在 $\det(\vec{\mathrm{B}}_k \mathrm{B}_k^T)$ 中的贡献为1
  - 由于遍历了所有n-1条边的组合，所以v_k为根的根树数目是 $\det(\vec{\mathrm{B}}_k \mathrm{B}_k^T)$

支撑树的计数(16)

- 例3. 3. 6 计算下图中以v_1为根的根树数目
  - 解：v_1所对应的基本关联矩阵是

$$B_1 = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & -1 \\ 0 & -1 & 0 & -1 & 1 & 0 \end{bmatrix}$$

$$\vec{B}_1 B_1^T = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 3 & -1 \\ -1 & 0 & 2 \end{bmatrix}$$

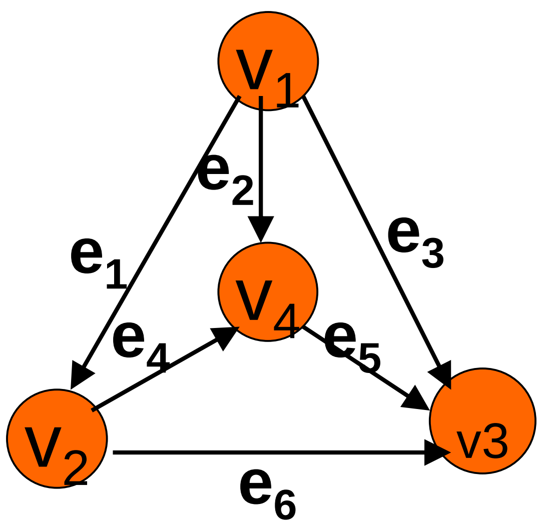


- ∴det($\vec{B}_1 B_1^T$)=6

数数v_1为根的根树数目？
考虑是否要e_2 及连接v_3方式，结果3*2条

支撑树的计数(17)

- **例3.3.7**
  - 求图中以v_1为根不含e_5的根树数目
  - 解：作G’=G−e_5,则G’的以v_1为根的根树数目正是所求，于是



$$B_1=\begin{bmatrix} -1 & 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 0 \end{bmatrix}$$

$$\det(\vec{B}_1 B_1^T)=\det\begin{bmatrix} 1 & 0 & 0 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{bmatrix}=4$$

基本关联矩阵及支撑树计数

你能发明出来吗？

- 行向量
  - 关联矩阵B为(n*m)，行向量相关，ran(B)=n-1
  - 基本关联矩阵B_k为(n-1)*m， ran(B_k)=n-1
  - B和B_k中(n-1)行向量线性无关
- 列向量

横看成岭侧成峰，排列组合求创新

  - 设H是连通图G的子图，如果H含有回路，则H的诸边对应的G的基本关联矩阵各列线性相关
  - B_k的任意n-1阶子阵行列式非零的充要条件是其各列所对应的边构成G的一棵支撑树（det(B_0)=0/1/-1）
  - B_k中行列式非零的n-1阶子阵的数目与G不同的支撑树数目之间一一对应
- 基于比内柯西定理，实现支撑树的计数

第三章 树

支撑树不唯一？

多少个（计数）？？　　最优树？？

关联矩阵不仅用于计算支撑树，还有呢？

- 回路矩阵与割集矩阵计算
- Huffman树算法
- 最短树算法

可逆矩阵

- **定义：设A为n级方阵，若存在n级方阵B，使得AB＝BA＝I，则称A为可逆矩阵，称B为A的逆矩阵，即$B=A^{-1}$.**
- 初等变换法求解逆矩阵
  - 矩阵的初等变换
    - （1）互换任意两行的位置 $r_i \leftrightarrow r_j$
    - （2）用非零数乘某行 kr_i
    - （3）用一个常数乘矩阵的某一行，再加到另一行上去 $r_i + kr_j$
  - 初等变换法

$$(A|I) \xrightarrow{\text{行初等变换}} (I|A^{-1})$$

可逆矩阵

- **例：**

求矩阵 $A=\begin{pmatrix} 1 & 1 & 2 \\ -1 & 2 & 0 \\ 1 & 1 & 3 \end{pmatrix}$ 的逆 A^{-1}

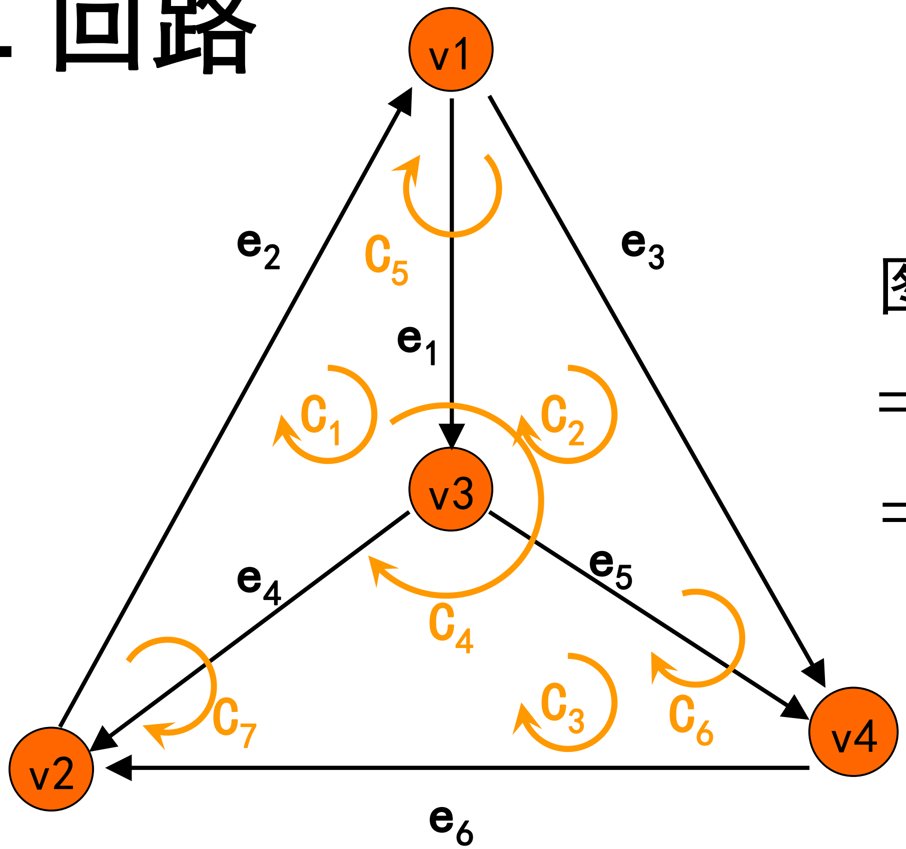
$$A^{-1}=\begin{pmatrix} 2 & -1/3 & -4/3 \\ 1 & 1/3 & -2/3 \\ -1 & 0 & 1 \end{pmatrix}$$

$$(A|I)=\left(\begin{array}{ccc|ccc} 1 & 1 & 2 & 1 & 0 & 0 \\ -1 & 2 & 0 & 0 & 1 & 0 \\ 1 & 1 & 3 & 0 & 0 & 1 \end{array}\right) \xrightarrow{r_2+r_1} \left(\begin{array}{ccc|ccc} 1 & 1 & 2 & 1 & 0 & 0 \\ 0 & 3 & 2 & 1 & 1 & 0 \\ 1 & 1 & 3 & 0 & 0 & 1 \end{array}\right)$$

$$\xrightarrow{r_3-r_1} \left(\begin{array}{ccc|ccc} 1 & 1 & 2 & 1 & 0 & 0 \\ 0 & 3 & 2 & 1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 & 1 \end{array}\right) \xrightarrow{\cdots\cdots} \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 2 & -1/3 & -4/3 \\ 0 & 1 & 0 & 1 & 1/3 & -2/3 \\ 0 & 0 & 1 & -1 & 0 & 1 \end{array}\right) = \left(I \middle| A^{-1}\right)$$

3.4回路矩阵

- 例3.4.1 回路



图有几个回路？

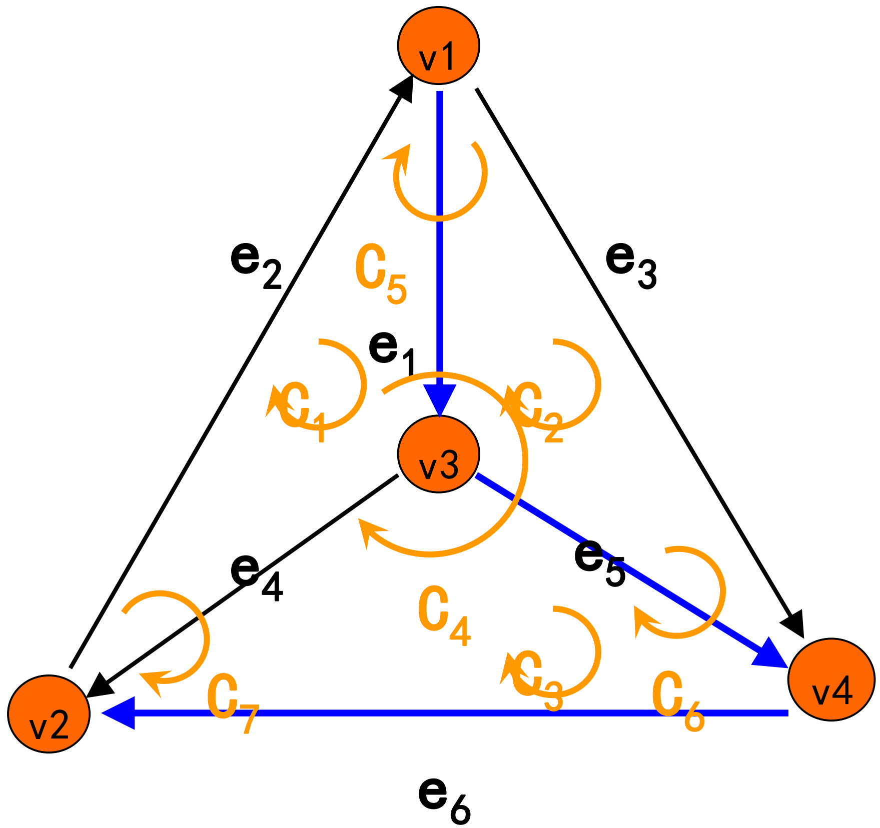
=2n−1?

=m+n−3?

考虑每条余树边是否使用……

3.4回路矩阵

- 例3.4.1 回路



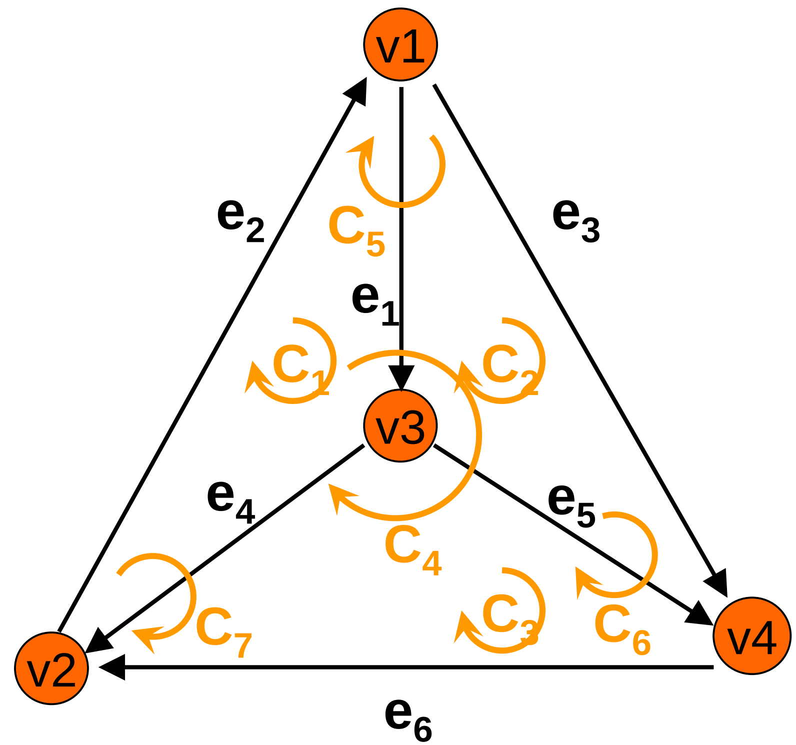
如何进行代数表示?

$$
\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\begin{bmatrix}
e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \\
1 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 \\
-1 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & -1 & 1 & 0 & 0
\end{bmatrix}
$$

最多可能包含$2^{m-n+1}-1$个不同的初级回路

回路矩阵(2)

- **回路的代数表示**
  - 给定C的一参考方向
    - 该回路的边若与参考方向一致就称为正向边
    - 否则就称为反向边
  - 有向连通图G的全部初级回路构成的矩阵，称为G的完全回路矩阵，记为C_e



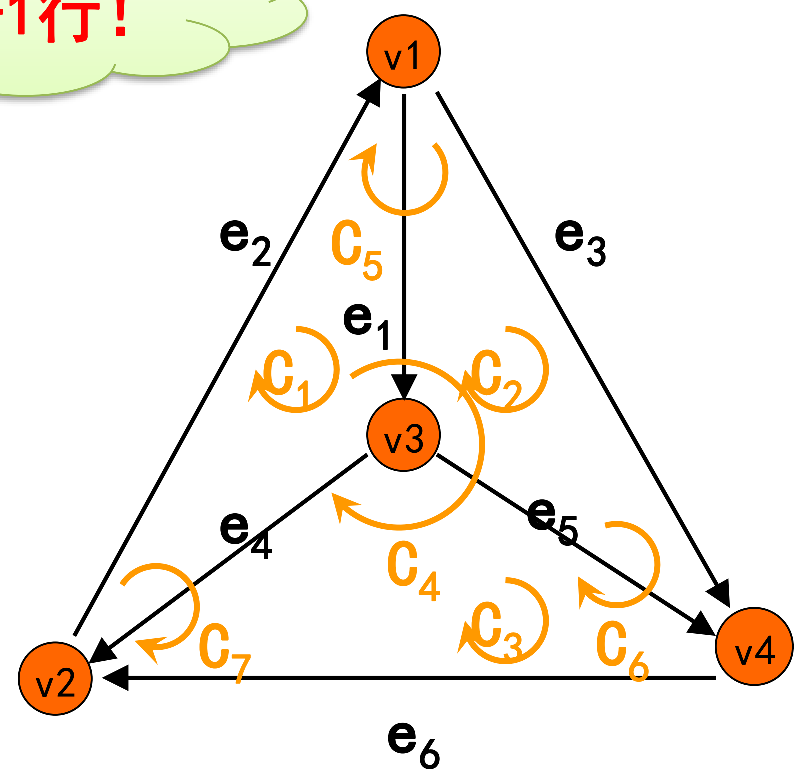
$$C_{ij}=\begin{cases}1, & e_j\in C_i \text{且与回路} C_i \text{方向一致。}\\ -1, & e_j\in C_i \text{且与回路} C_i \text{方向相反。}\\ 0, & \text{其他}\end{cases}$$

回路矩阵(4)

- **例3. 4. 1**
  - 完全回路矩阵

2^m−n+1^−1行!

$$C_e=\begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & -1 & 0 \\ -1 & 0 & 1 & -1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$



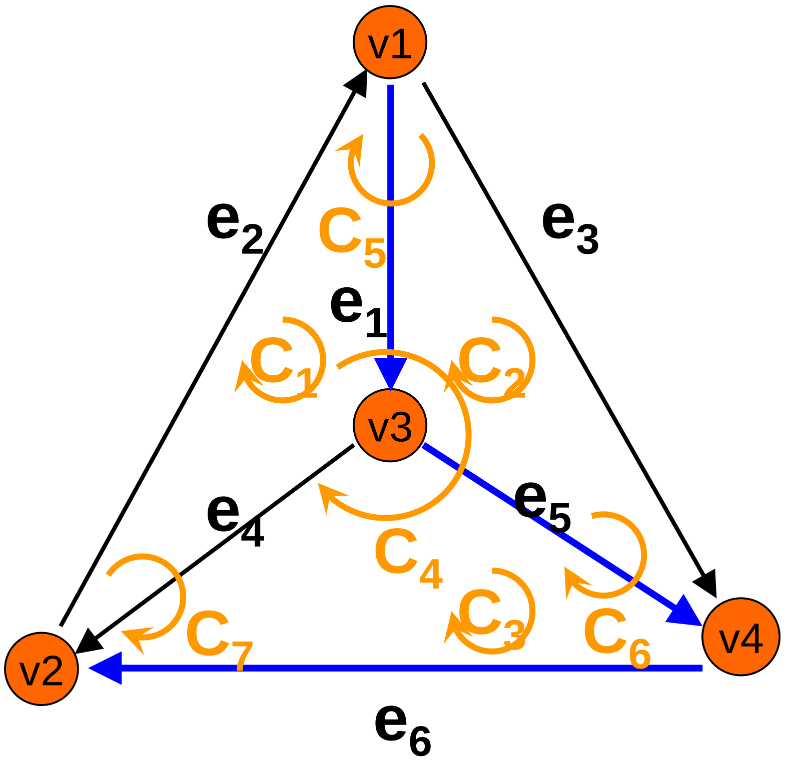
  - 这些回路是否是独立的?
  - 秩是多少? 哪些回路独立?

$C_1\oplus C_2=C_5\qquad C_1\oplus C_3=C_7$

进一步考虑余树边?

回路矩阵(5)

- 设T是图G=(V,E)的一棵支撑树
- 对任意余树边e∈E(G)- E(T), T+e都构成一个唯一回路C
- 当有向图G=(V,E)的树T确定后，每条余树边e与T的子集所对应的回路称为基本回路
- 基本回路的方向与余树边e的方向一致



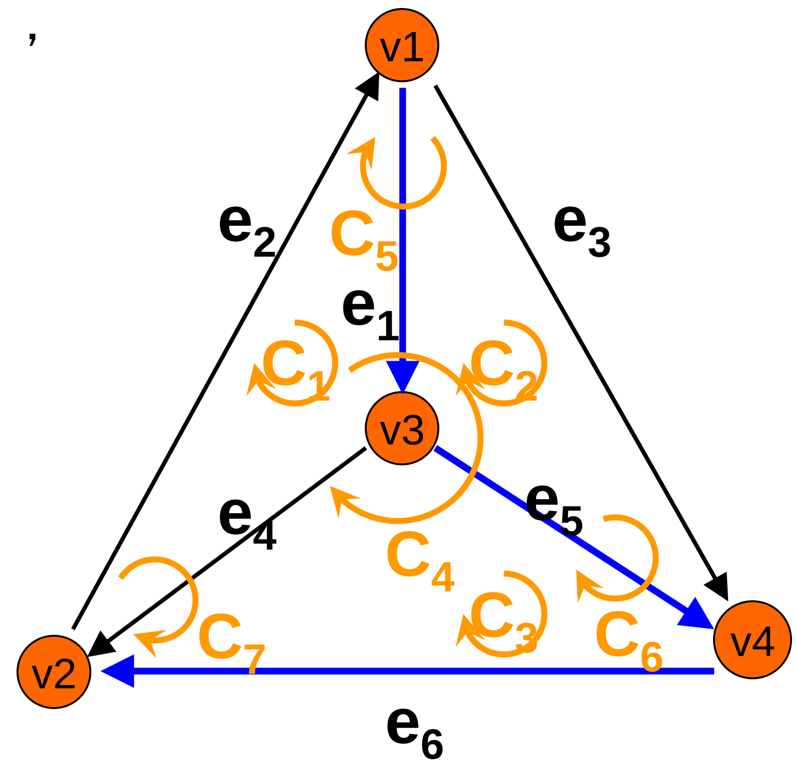
- 由全部基本回路构成的矩阵称为G的基本回路矩阵，记为$C_{f\,(m-n+1)*m}$

回路矩阵(6)

- **例3. 4. 2**
  - 给定图的一棵树T={e_1, e_5, e_6} ，则其基本回路矩阵是

$$C_f=\begin{array}{c}\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix} \\ \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{matrix}\end{array}$$

余 树 边

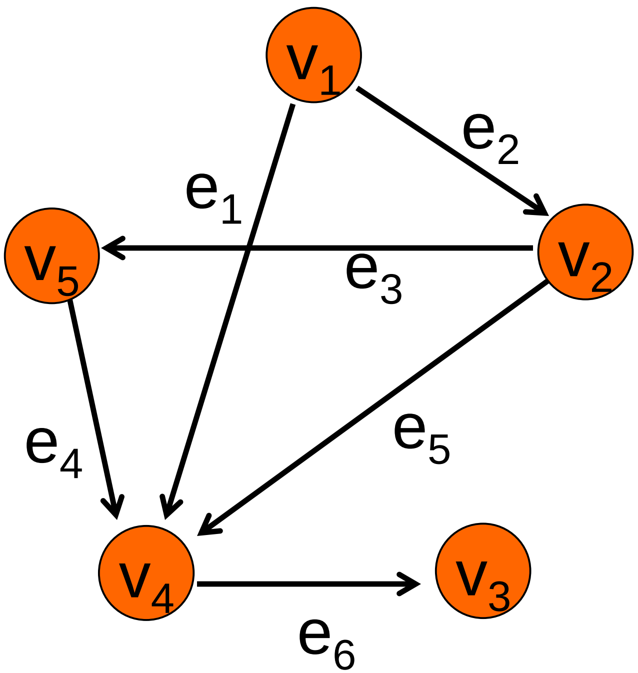


  - 显然基本回路矩阵中每个回路都是独立的
  - 因此ran C_f=m−n+1

其他回路呢？C_e的秩呢？

关联矩阵及其性质

- **定理3.2.3**
  - 有向连通图G=(V,E)关联矩阵B的秩ran B＝n-1



$$
\begin{array}{c}
\\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5
\end{array}
\begin{array}{c}
\begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\
\begin{bmatrix}
1 & 1 & 0 & 0 & 0 & 0 \\
0 & -1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 \\
-1 & 0 & 0 & -1 & -1 & 1 \\
0 & 0 & -1 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

回路矩阵(7)

- **定理3.4.1（引理）**
  - 有向连通图G=(V,E)的关联矩阵$B_{n\times m}$与完全回路矩阵$C_{e(\sim)\times m}$的边次序一致时，恒有：

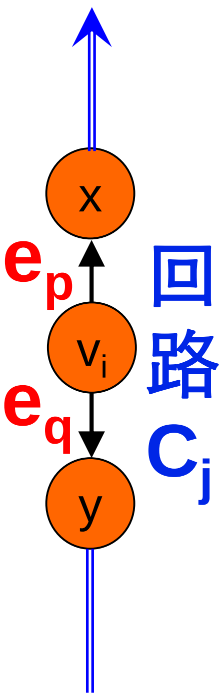
$$BC_e^T = 0$$

B: 结点和边的关系
C: 回路和边的关系

- **证明：** $D=BC_e^T, d_{ij} = \sum_{k=1}^{m} b_{ik} \bullet c_{jk}$
  - 设
  - 其中b_{ik}是结点v_i与边e_k的关联状况
  - c_{jk}是回路 c_j与边e_k的相关情况
  - d_{ij}不为0：给定v_i和c_j，存在边e_k与v_i和c_j都关联

回路矩阵(8)

- 证(续)
- 考虑回路C_j与结点 v_i的相处，只有两种可能：
- （1）C_j不经过v_i，则与v_i关联的任一边都不是C_j中的边，所以$b_{ik}=0$，即$d_{ij}=0$
- （2）C_j经过v_i，则必定经过与 $D=BC_e^T, d_{ij}=\sum_{k=1}^{m} b_{ik} \bullet c_{jk}$ v_i关联的2条边 e_p和e_q
  - 如果e_p和e_q在C_j中方向相反，对 v_i它们却是同进同出的，因此对 e_p和e_q有b不变而c相反，即$d_{ij}=0$
  - 若e_p和e_q在C_j中方向一致，则对v_i来说它们是一进一出的，因此对 e_p和e_q有c不变而b相反，即$d_{ij}=0$
- 由于 d_{ij}的任意性，故定理得证 $BC_e^T=0$



回路矩阵(11)

- 呼唤定理
  - 有向连通图G=(V,E)的完全回路矩阵C_e的秩是m-n+1

如何证明呢?

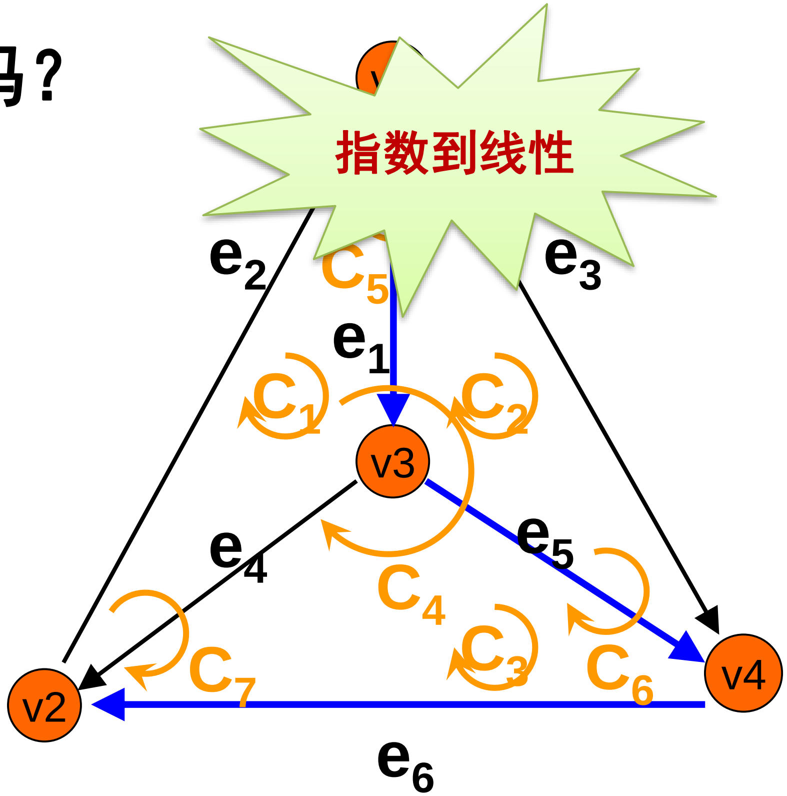
- 苦思冥想不得其解☹
- 找找数学系的朋友：
  - Sylvester定理指出，两个矩阵$A_{n\times s}$，$B_{s\times m}$，如果 AB=0，则ran A+ran B ≤s

回路矩阵(12)

- **定理3.4.2**
  - 有向连通图G=(V,E)的完全回路矩阵C_e的秩是m-n+1
- **证明:**
  - 由于基本回路矩阵C_f 是完全回路矩阵C_e 的子阵且ran C_f =m-n+1，故ran C_e ≥m-n+1
  - 现证ran C_e ≤m-n+1
  - Sylvester定理指出，两个矩阵$A_{n\times s}$，$B_{s\times m}$，如果 AB=0，则ran A+ran B ≤s
  - 由定理3.4.1 $BC_e^T = 0$，得到 ran B+ran C_e ≤m，关联矩阵B有ran B=n-1
  - 因此 ran C_e ≤m-n+1

回路矩阵(13)

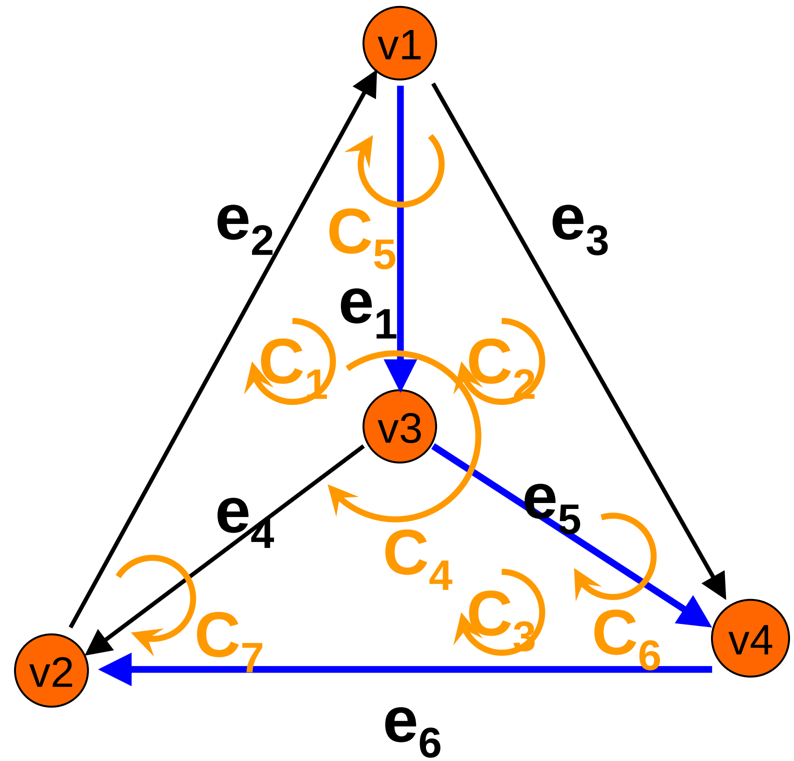
- 基本回路矩阵C_f有多少个？等价吗？
- 定义3.4.3：回路矩阵
  - 由连通图G中m-n+1个互相独立的回路组成的矩阵，称为G的回路矩阵，记为C
- 回路矩阵的简单性质：
  1. 基本回路矩阵C_f是回路矩阵
  2. $BC^T=0$，（其中B和C的边次序一致）
  3. $C=P\ C_f$ ， 其中P是非奇异(即行列式不为0)的方阵，C与基本回路矩阵C_f的边次序一致



回路矩阵(14)

- 例3.4.2：给定图的一棵树T={e_1, e_5, e_6}，则其基本回路矩阵是

$$C_f = \begin{array}{c} \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix} \\ \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{matrix} \end{array}$$



- 每条余树边用且仅用1次
- ran C_e= ran C_f=m−n+1

如何给出基本回路？设计算法还是矩阵运算？

回路矩阵(15)

- 每条余树边用且仅用1次
  - 将行、列分别进行交换，使树枝边放在后，余树边放在前且次序与它所构成的回路一致，就可以写成分块矩阵形式

$$C_f = \begin{array}{c} \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ -1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix} \\ \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{matrix} \end{array} \qquad C_f = \begin{array}{c} \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 \end{bmatrix} \\ \begin{matrix} e_2 & e_3 & e_4 & e_1 & e_5 & e_6 \end{matrix} \end{array}$$

余树边　树枝边

- 亦即 $C_f = \begin{pmatrix} I & C_{f_{12}} \end{pmatrix}$
- 其中 $C_{f_{12}}$ 是树枝边所对应的子阵

$BC^T=0$：呼唤定理

回路矩阵(9)

- 定理3.4.1（引理）
  - 有向连通图G=(V,E)的关联矩阵$B_{n\times m}$与完全回路矩阵$C_{e\ (\sim)\times m}$的边次序一致时，恒有：

$$BC_e^T = 0$$

推论： 有向连通图的基本关联矩阵B_k，基本回路矩阵C_f，在边次序一致的情况下，有

$$B_k C_f^T = 0$$

回路矩阵(10)

定理3.4.4 若有向连通图G=(V,E)的基本关联矩阵B_k是和基本回路矩阵C_f的边次序一致，其中

$$C_f = (I; C_{f_{12}}) \quad B_k = (B_{11}; B_{12}),$$

，则

$$C_{f_{12}} = -B_{11}^T (B_{12}^{-1})^T$$

证明： 由推论知 ，写成块矩阵形式

$$(B_{11} \quad B_{12})\begin{bmatrix} I \\ C_{f_{12}}^T \end{bmatrix} = 0 \Rightarrow B_{12} \cdot C_{f_{12}}^T = -B_{11}$$

根据基本关联矩阵，可以通过计算得到基本回路矩阵

回路矩阵(16)

- **定理3.4.4**

$\boxed{BC^T=0}$

  - 若有向连通图G=(V,E)的基本关联矩阵B_k和基本回路矩阵C_f的边次序一致，并设$C_f=(I\ \ C_{f12})$，$B_k=(B_{11}\ \ B_{12})$，分别对应余树边和树枝边，则

$$C_{f_{12}} = -B_{11}^{T}B_{12}^{-T}$$

余树边

树枝边

- **已知回路矩阵C的秩为(m-n+1)，那么哪(m-n+1)列线性无关？（回路矩阵的大小C(m-n+1, m)）**

$$\begin{array}{cc} & \begin{array}{cccccc} e_2 & e_3 & e_4 & e_1 & e_5 & e_6 \end{array} \\ C_f = & \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 \end{array}\right] \end{array}$$

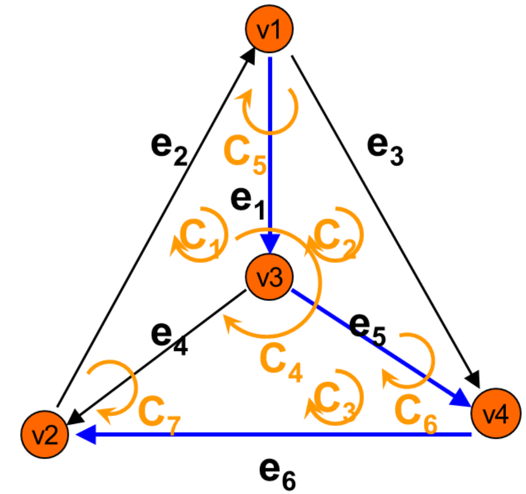
余树边　树枝边

- **定理3.4.3**
  - 连通图G的回路矩阵C的任一(m-n+1)阶子阵行列式非零，当且仅当这些列对应于G的某一棵余树
  - 即去掉的(n-1)条边不含基本回路

回路矩阵(17)

- 例3.4.3：已知图3.11基本关联矩阵

$$B_4 = \begin{array}{c} \begin{bmatrix} 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & -1 \\ -1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \\ \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{matrix} \end{array}$$



- 其中e_1, e_5, e_6所对应的矩阵行列式非零
- 求基本回路矩阵C_f

分析：B子阵行列式非零即边无回路，4点3边？

回路矩阵(18)

- 解：由e_1, e_5, e_6可构成G的一棵树，对B_4进行列变换，得到

$$B_4 = \begin{array}{c} \begin{bmatrix} -1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & -1 & 1 & 0 \end{bmatrix} \\ \begin{matrix} e_2 & e_3 & e_4 & e_1 & e_5 & e_6 \end{matrix} \end{array} = \begin{pmatrix} B_{11} & B_{12} \end{pmatrix}$$

余树边　树枝边

其中

$$B_{11} = \begin{bmatrix} -1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 1 \end{bmatrix} \quad B_{12} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix} \quad B_{12}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$C_{f_{12}} = -B_{11}^{T} B_{12}^{-T}$$

回路矩阵(19)

- **基本回路矩阵的右子阵**

$$C_{f_{12}} = -B_{11}^{T} B_{12}^{-T} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & 0 \\ 0 & -1 & -1 \end{bmatrix}$$

- **基本回路矩阵**

$$C_f = \begin{array}{c} \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 \end{bmatrix} \\ \begin{matrix} e_2 & e_3 & e_4 & e_1 & e_5 & e_6 \end{matrix} \end{array}$$

余树边　　　　树枝边

回路矩阵(20)

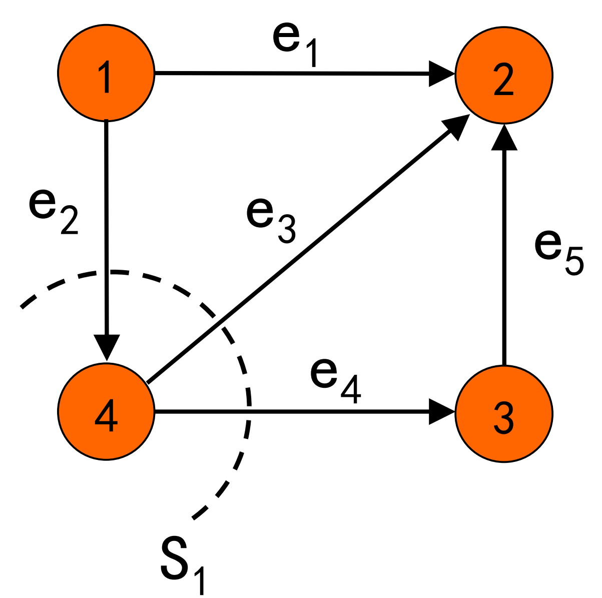
- **回顾：回路矩阵的创新思路**
  - 回路的代数表示（方向）
  - 有多少个回路呢？
  - 完全回路矩阵，相关性和秩？
  - 如何构造独立的回路（树）
  - 基于$BC^T=0$计算C
- **于是接着如何写书呢？**

照回路画割集

边集合：道路、回路、树，还有什么？

3.4 割集矩阵(1)

- **定义3.4.4 割集**
  - 设S是有向图G=(V,E)的边子集，若同时满足：
    1. G’ =(V,E-S) 比G的连通支数多1
    2. 对任意 $S' \subset S$，G与G” =(V,E-S’)的连通支数相同
  - 则称S是G的一个割集

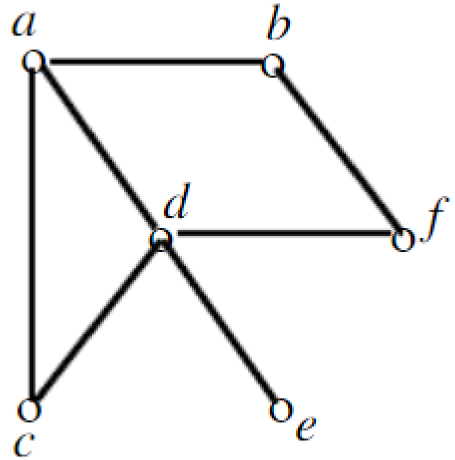


$S_1=\{e_2, e_3, e_4\}$ 是割集

单选题 1分

设置

下图中关于割边割集的说法，正确的是：



A {(a, d)}是割边

B {(a, d)}是割集

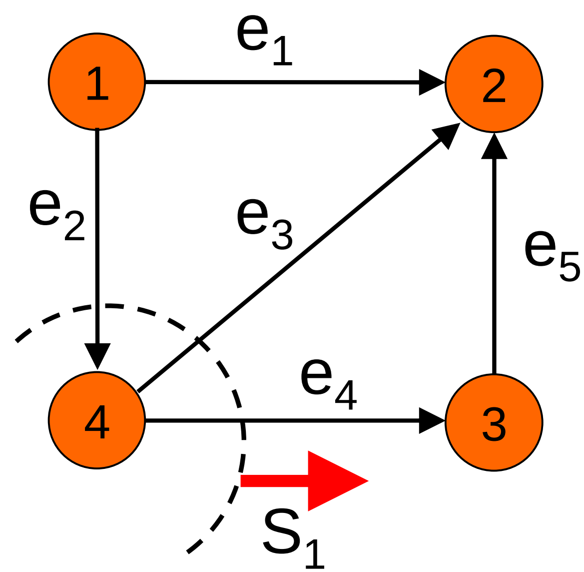
C {(d, e)}是割集

D {(a, d) ,(a, c)}是割集

提交

割集矩阵(2)

- 割集能有方向吗？
  - 给S确定一个方向，则S中每条边e与S 同向或反向
- 例3.4.4
  - e_2与S_1方向相反，e_3、e_4与S_1方向相同



如何代数表示？

$$
B_4 = \begin{array}{c} \begin{bmatrix} 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & -1 \\ -1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \\ \begin{matrix} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{matrix} \end{array}
\qquad
C_f = \begin{array}{c} \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 \end{bmatrix} \\ \begin{matrix} e_2 & e_3 & e_4 & e_1 & e_5 & e_6 \end{matrix} \end{array}
$$

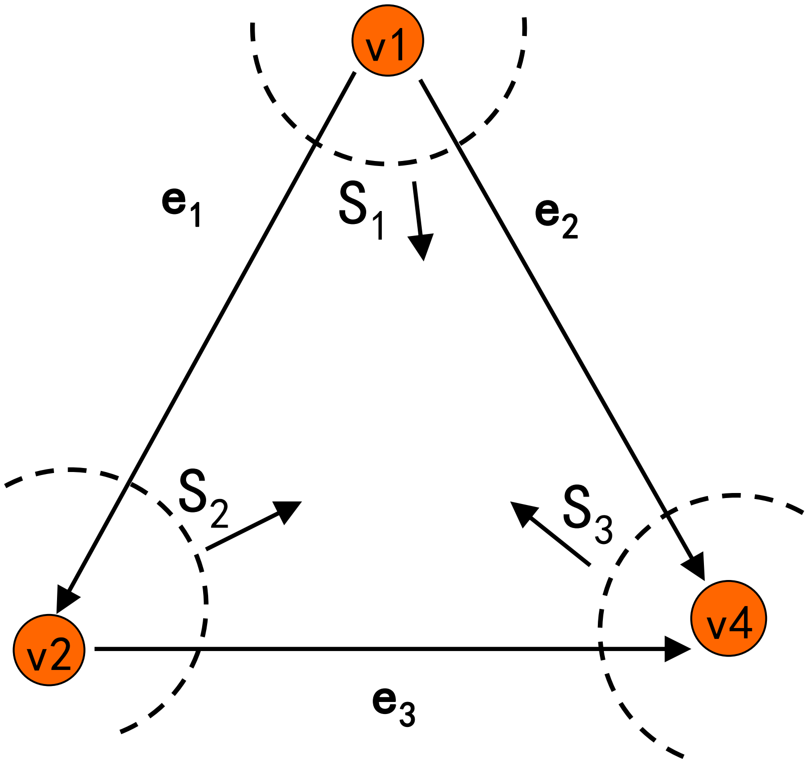
割集矩阵(3)

- **定义3.4.5**
  - 有向连通图G的全部割集组成的矩阵，称为完全割集矩阵。记作S_e，其元素

$$S_{ij}=\begin{cases}1, & e_j\text{在}S_i\text{中且方向一致}\\ -1, & e_j\text{在}S_i\text{中且方向相反}\\ 0, & \text{其他}\end{cases}$$

割集矩阵(4)

- 例3. 4. 5 求左图的完全割集矩阵



$$
S_e = \begin{array}{c} \\ \\ \\ \end{array}
\begin{array}{cccc}
 & e_1 & e_2 & e_3 \\
\end{array}
$$

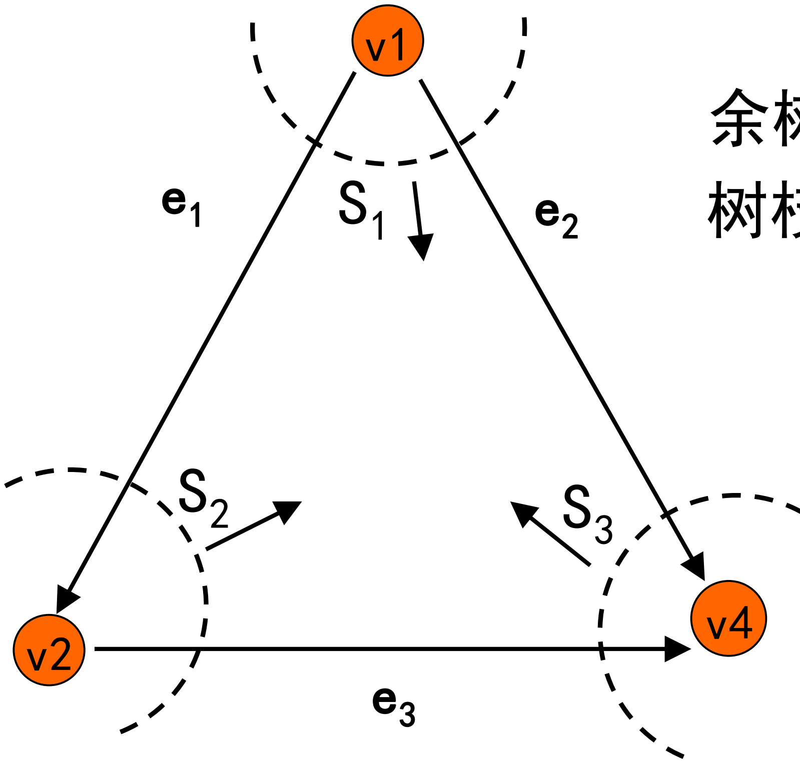
$$
S_e = \begin{array}{c c} & \begin{array}{ccc} e_1 & e_2 & e_3 \end{array} \\ & \begin{pmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & -1 \end{pmatrix} \begin{array}{c} S_1 \\ S_2 \\ S_3 \end{array} \end{array}
$$

割集的独立性？

如何构造独立的割集？

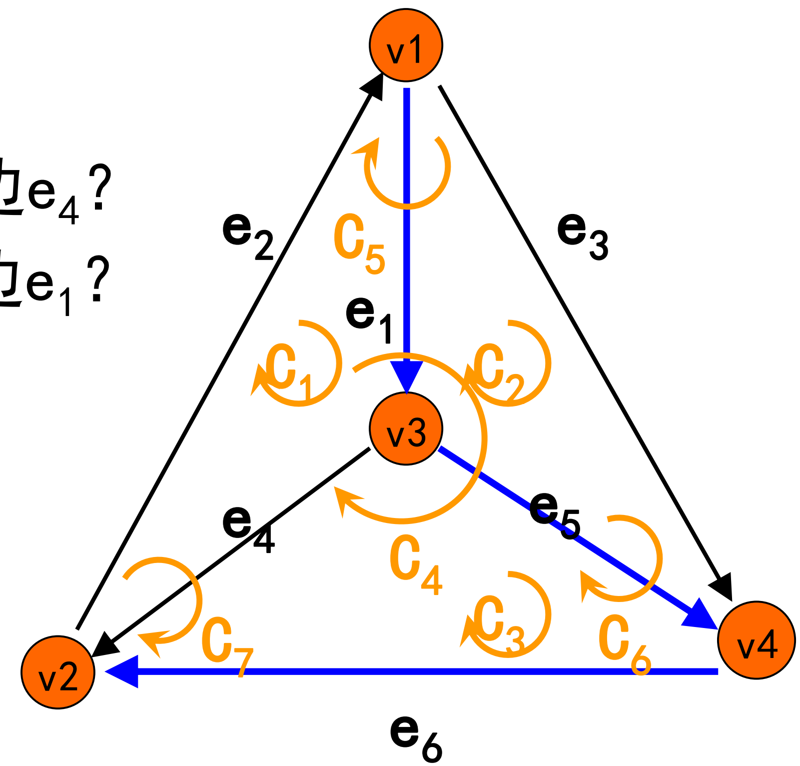
割集矩阵(4)

- 例3. 4. 5 求左图的完全割集矩阵



余树边e_4?

树枝边e_1?



割集的独立性?

如何构造独立的割集?

割集矩阵(5)

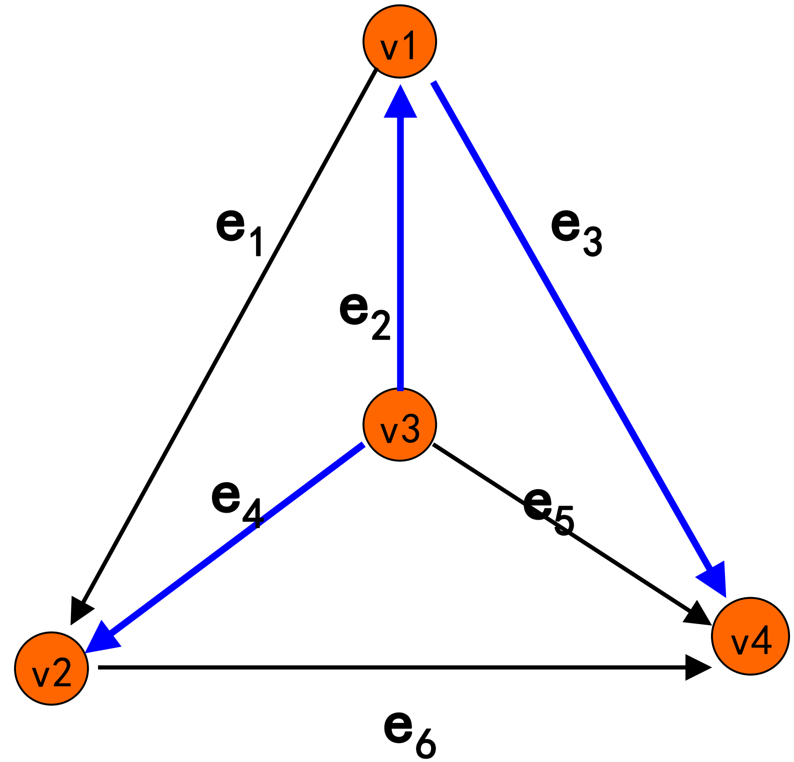
- **定义3.4.6**
  - 设T是连通图G的一棵树， e_i是一个树枝。
  - 对树枝边e_i存在G的割集S_i，S_i只包括一条树枝边e_i及某些余树边，且与e_i方向一致。
  - 这时S_i为G的对应树T的一个基本割集。
- **定义3.4.7**
  - 给定有向连通图G的一棵树T，则由全部基本割集组成的矩阵成为基本割集矩阵。记为S_f 。
  - $\mathrm{ran}\ S_f = n - 1$

割集矩阵(6)

$$S_f=\begin{array}{c}\begin{bmatrix}-1 & 1 & 0 & 0 & 1 & 1\\ 0 & 0 & 1 & 0 & 1 & 1\\ 1 & 0 & 0 & 1 & 0 & -1\end{bmatrix}\\ \begin{matrix}e_1 & e_2 & e_3 & e_4 & e_5 & e_6\end{matrix}\end{array}$$

$$S_f=\begin{array}{c}\begin{bmatrix}-1 & 1 & 1 & 1 & 0 & 0\\ 0 & 1 & 1 & 0 & 1 & 0\\ 1 & 0 & -1 & 0 & 0 & 1\end{bmatrix}\\ \begin{matrix}e_1 & e_5 & e_6 & e_2 & e_3 & e_4\end{matrix}\end{array}$$

余树边 & 树枝边



$$T=\{e_2,e_3,e_4\}$$

完全割集矩阵S_e的秩？

割集矩阵(7)

- **定理3.4.5**
  - 当有向连通图G的完全回路矩阵C_e和完全割集矩阵S_e的边次序一致时，有 $S_e C_e^T = 0$

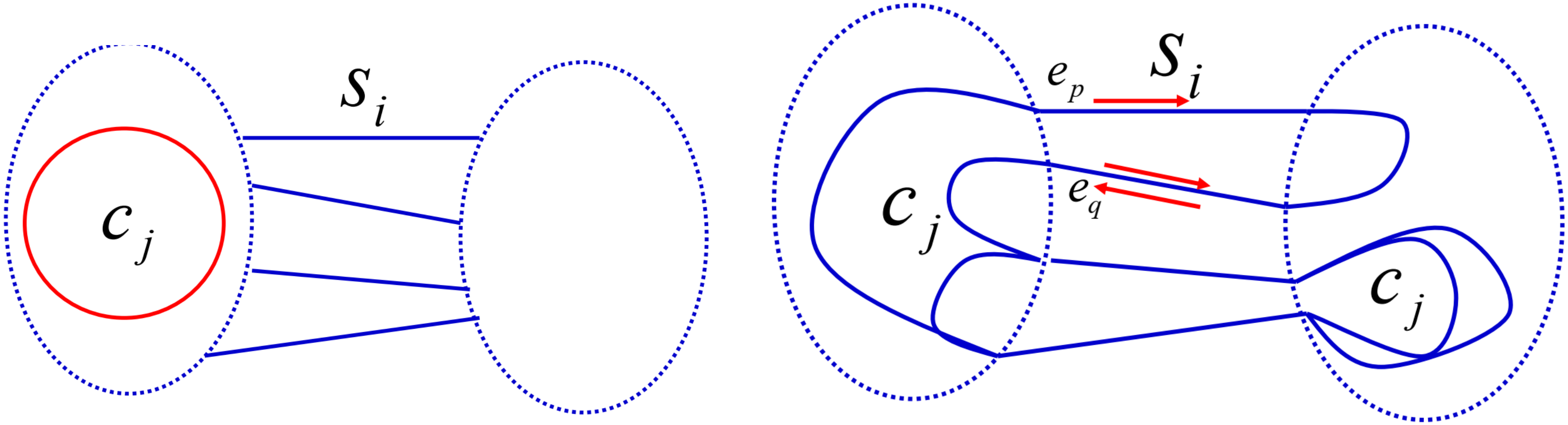
证明：设 $D = S_e C_e^T$ $\qquad d_{ij} = \sum_{k=1}^{m} s_{ik} \cdot c_{jk}$

其中 s_{ik} **是第i个割集** S_i **中** e_k **的情况**

c_{jk} **是第j个回路** c_j **中** e_k **的情况**

考虑：$BC^T=0$

回路和割集必重复偶数条边



割集矩阵(7)

- **定理3.4.5**
  - 当有向连通图G的完全回路矩阵C_e和完全割集矩阵S_e的边次序一致时，有 $S_e C_e^T = 0$

- **定理3.4.6**
  - 连通图G的完全割集矩阵S_e的秩是n-1
    - 基本割集矩阵S_f是S_e的子矩阵，而ran S_f = n − 1，因此ran S_e ≥n-1
    - 由定理3.4.5及Sylvester定理
    - ran S_e +ran C_e ≤m，而ran C_e = m-n+1，故ran S_e ≤n-1

这(n-1)个割集很重要?

割集矩阵(8)

- **定义3.4.8**
  - 连通图G的n-1个互相独立的割集构成的矩阵成为G的割集矩阵，记为S。
- **割集矩阵S有以下性质**
  - 基本割集矩阵S_f是割集矩阵
  - $SC^T=0$，S和C的边次序一致
  - $S_f = PS$，其中P是非奇异方阵，S与S_f的边次序一致

如何给出基本割集？设计算法还是矩阵运算？

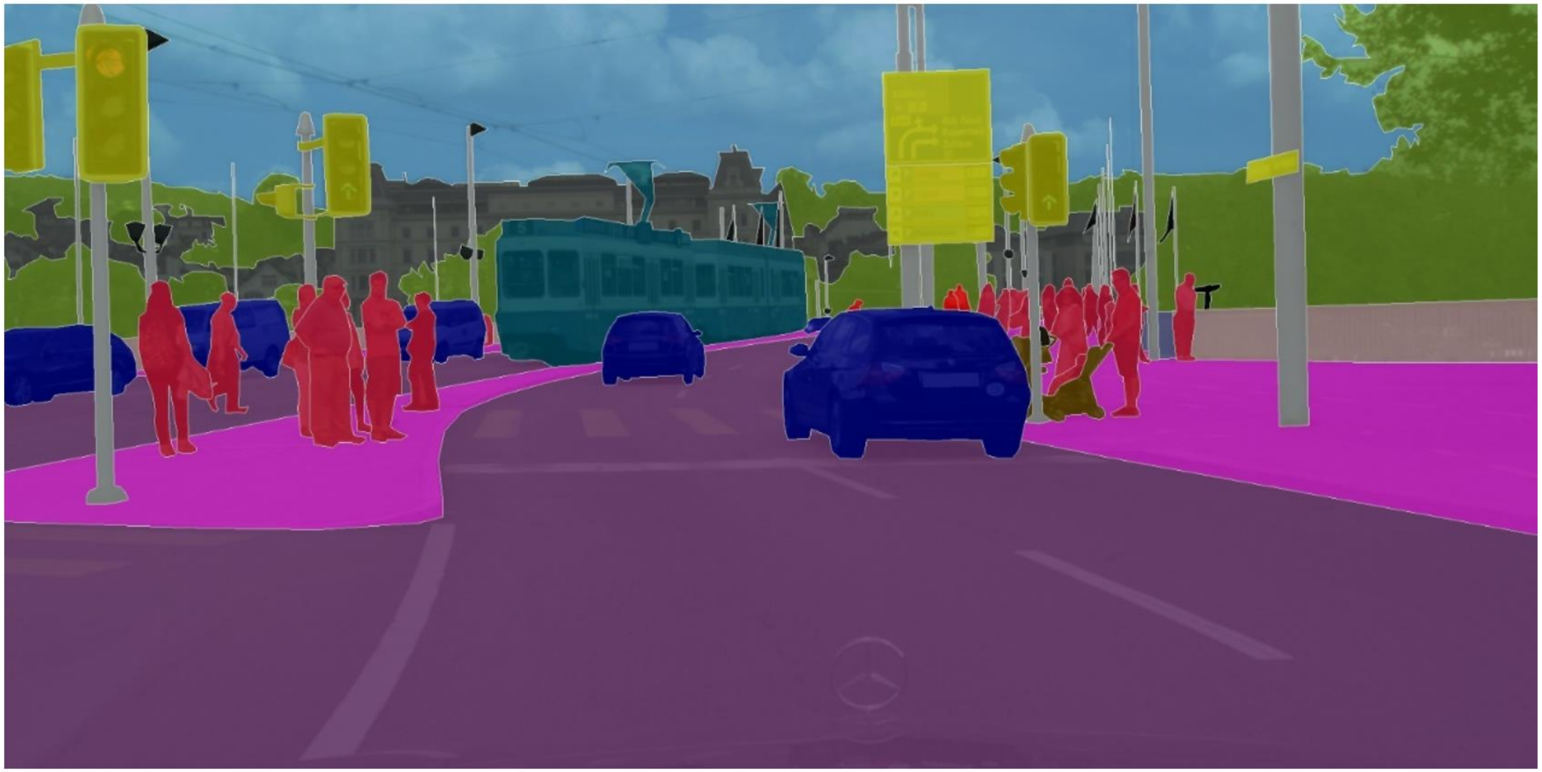
割集矩阵(9)

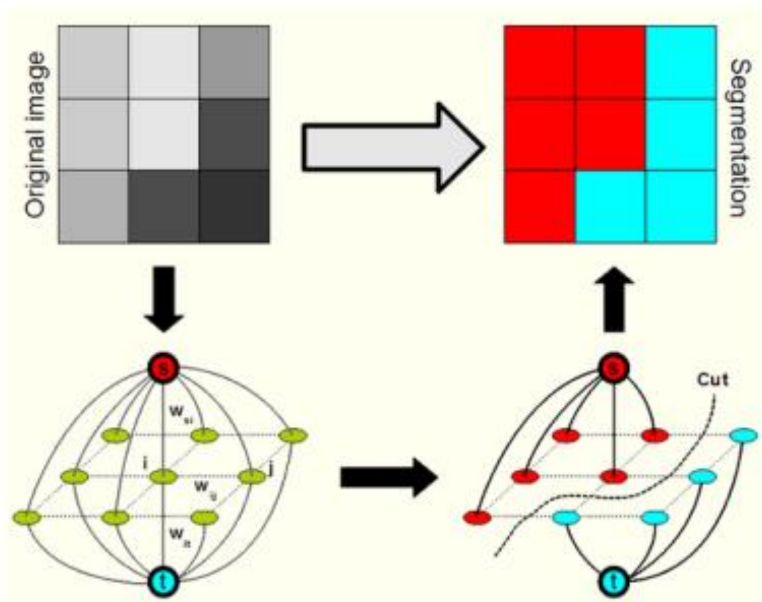
- **定理3.4.8**
  - 设S_f和C_f分别是连通图G中关于某棵树T的基本割集矩阵和基本回路矩阵，且边次序一致，并设

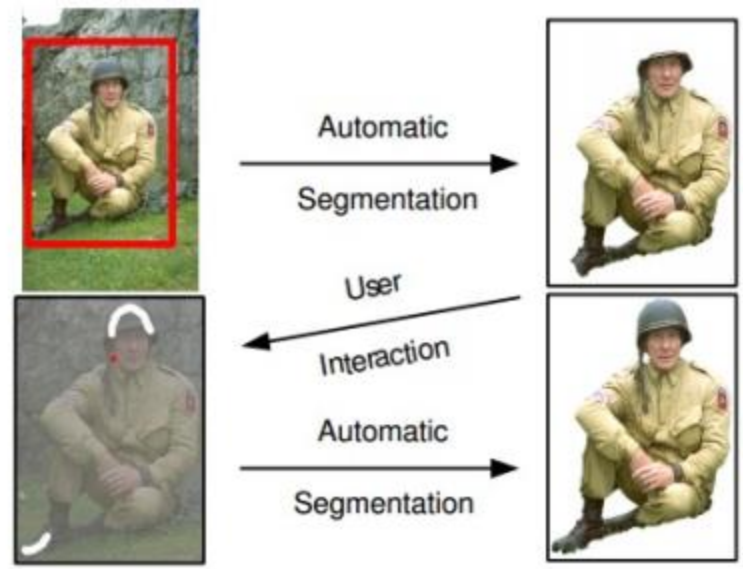
$$S_f = (S_{f_{11}} \quad I), C_f = (I \quad C_{f_{12}})$$

则 $S_{f_{11}} = -C_{f_{12}}^T$

  - 由定理3.4.5 $S_e C_e^T = 0$ 易证
- **推论** 已知 $C_{f_{12}} = -B_{11}^T B_{12}^{-T}$
  - 当连通图G的基本割集矩阵与基本关联矩阵的边次序一致时，有 $S_{f_{11}} = B_{12}^{-1} B_{11}$




Original image
Segmentation
Cut


Automatic
Segmentation
User
Interaction
Automatic
Segmentation

第三章 树

- 树的有关定义
- 基本关联矩阵及其性质
- 支撑树的计数
- 回路矩阵与割集矩阵
- Huffman树
- 最短树

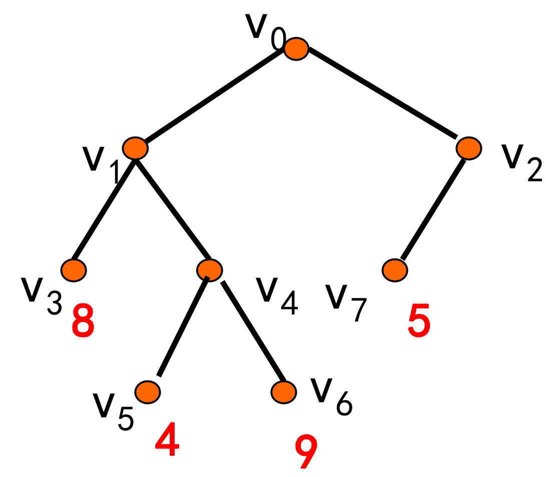
Huffman树

- 例3.6.1
  - 已知13个英文字母构成字符串adacatedecade。试用二进制字符串代替字母，并保证该英文字符串与二进制串能够一一对应。
  - 例如ASCII码
    - a=0x61，c=0x63， …… ，t=0x74
    - 总长度：8bit * 13=104 bit
  - 其他方法
    - 用0表示a，用1表示d，用00表示c ……?
    - 接收到：010000…
  - 如何编码(保证能唯一解码)，能够使总长最小?

接收端
不能恢复

Huffman树(1)

- **定义3.6.1（二叉树与完全二叉树）**
  - 除树叶外，其余结点的正度最多为2的外向树称为二叉树
  - 正度都是2的称为完全二叉树
  - 为什么学二叉树而不是三叉树？

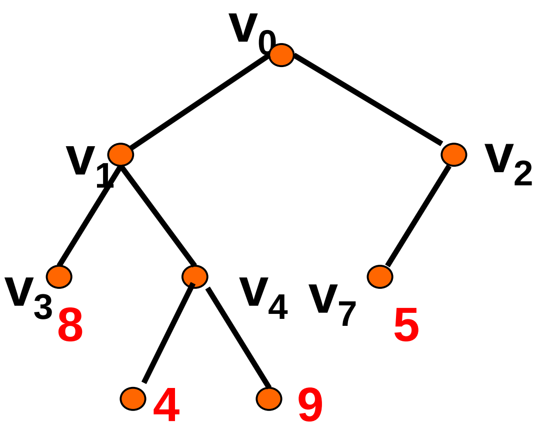


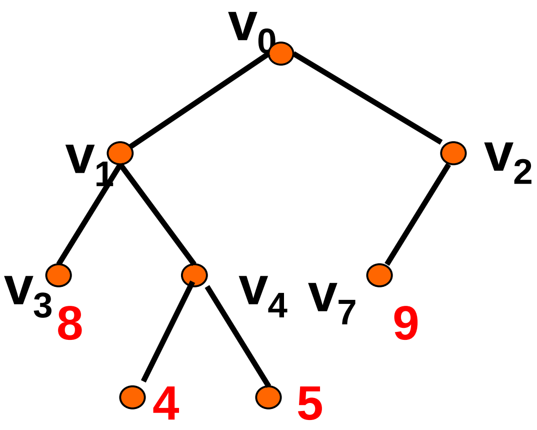
- **赋权二叉树**
  - 二叉树T的每一个叶结点v_i都分别赋以一个正实数w_i
- **带权路径总长度（WPL）**
  - 树根v_0到叶结点v_i的路径$P(v_0, v_i)$所包含的边数记为路径的长度l_i，则二叉树T带权的路径长度总长是

$$WPL=\sum_i l_i w_i, v_i\text{是树叶}$$

Huffman树(2)

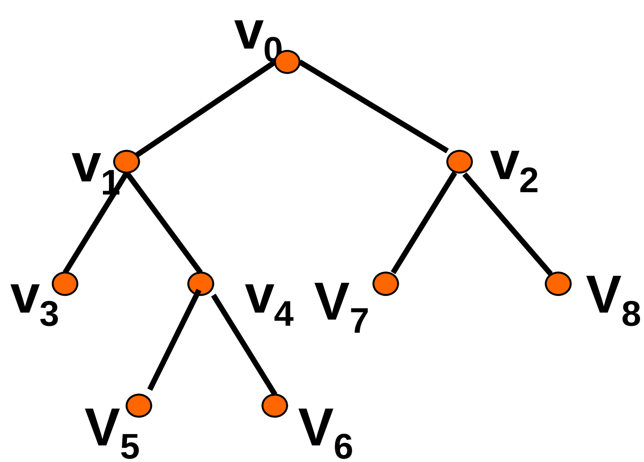
- **最优二叉树**
  - 若给定了树叶树目以及它们的权，可以构造出不同的赋权二叉树，在这些二叉树中，带权路径总长WPL最小的二叉树称为最优二叉树





Huffman树(2)

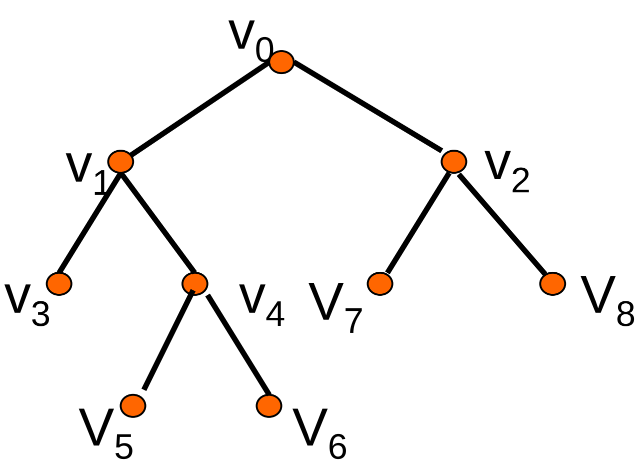
- **例3.6.1**
  - 已知英文字符串adacatedecade
  - 试用二进制字符串代替某个字母，并保证该英文字符串与二进制串构成一一对应
  - 解：该字符串中有字母a, d, e, c, t
  - 令每个字母对应二叉树的一个树叶

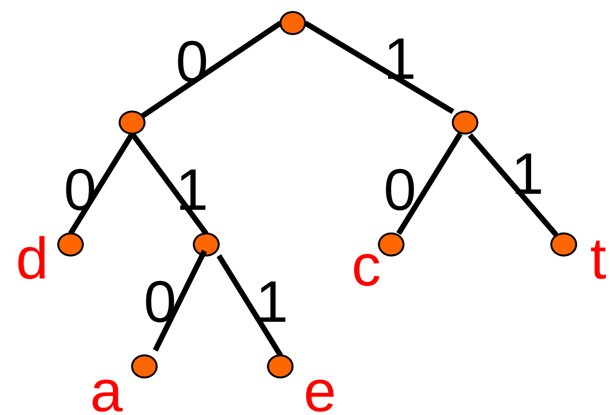


Huffman树(3)

例（续:adacatedecade）

- 根到树叶的路径是唯一的，而且这条路径绝不会是树根到另一个树叶路径的一部分
- 构造一一映射：根到树叶的路径与字母
- 如果在树T中令向左的边为0，向右的边为1，那么这些路径又与二进制串构成了一一映射



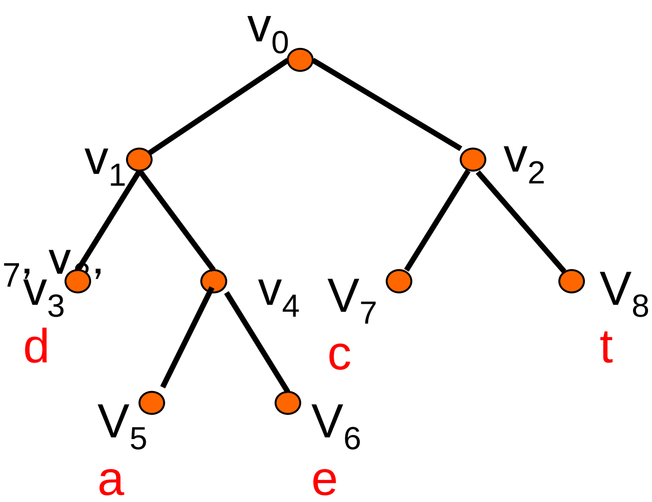


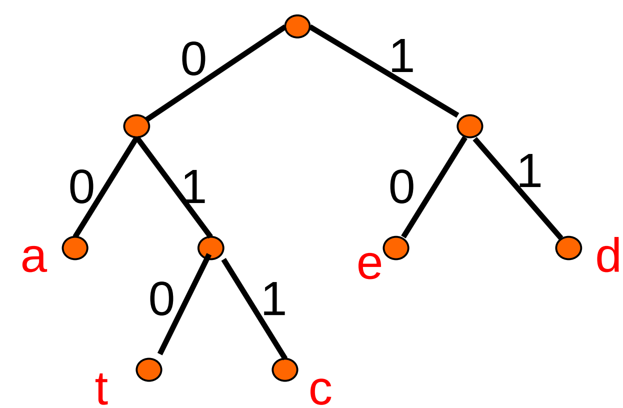
010000101001010110001110010000011

Huffman树(4)

例（续）

- 英文字符串adacatedecade
- 例如令d， a， e， c， t分别对应左图的v_3, v_5, v_6, v_7, v_8, 则d←00, a ←010, e ←011, c ←10, t ←11
- 该英文字符串对应 01000010100101011000111001000011
- 如果字母与树叶的对应情况如下图，即a←00, t←010, c←011, e ←10, d←11
- 则对应字符串是00110001100010101110011001110
- 这两种情况下字符串的总长分别是33和29
- 如何构造总长最短的二叉树？
- 字母a, d, e, c, t分别出现4, 3, 3, 2, 1次

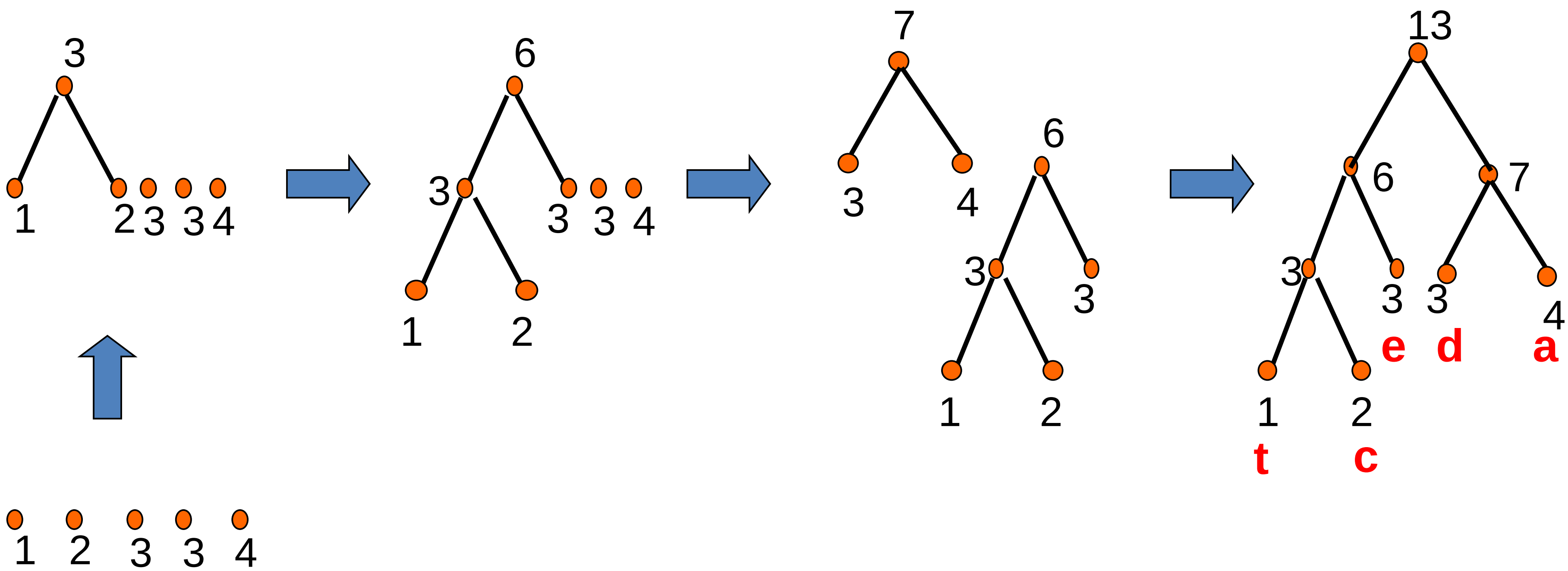




树的结构？字母的位置？
重要性相同的节点位置相同

Huffman树(6)

- **例3.6.2**
  - 字母a, d, e, c, t分别出现4, 3, 3, 2, 1次
  - 构造权序列为(1, 2, 3, 3, 4)的Huffman树



Huffman树(5)

- **Huffman树**
  - Huffman给出了构造n个树叶的最优二叉树算法，由此算法得到的树称为Huffman树
- **算法描述如下**
  1. 对n个权值(叶子节点)由小到大进行排序，满足 $w_{i1} \leqslant w_{i2} \leqslant w_{i3} \leqslant \cdots \leqslant w_{in}$
  2. 计算虚拟权值$w_i=w_{i1}+w_{i2}$，作为新增中间结点v_i的权，v_i的左儿子为v_{i1}，右儿子为v_{i2}
  3. 在权序列中删除 w_{i1}, w_{i2}，加入新的虚拟节点w_i，n=n-1。当n＝1时结束，否则，转（1）

Huffman树(7)

- **复杂度分析**
  - 算法的计算复杂度主要取决于步骤1，而且是n个权值的第一次排序，它一般需进行nlogn次比较
  - 之后每当产生w_i时，只需在新序列中进行插入运算，其复杂度是logn
  - 总共进行n-2次循环
  - 因此整个算法的计算复杂度是O(nlogn)
- **定理3.6.1**
  - 由Huffman算法得到的二叉树为最优二叉树

Huffman树(8)

- **Huffman树和Huffman编码**
  - 使用熵编码的无损压缩
- **广泛应用于各种压缩算法中**
  - ZIP（无损压缩）
  - MP3，JPEG（有损压缩）

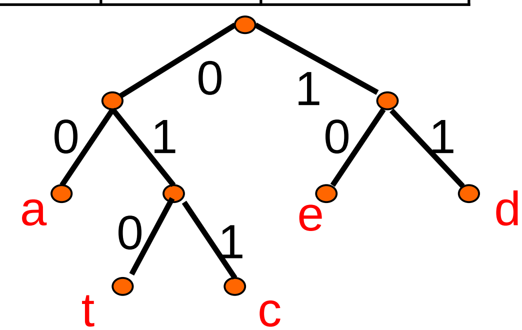
卷积码等
前向纠错

- **例：JPEG文件格式**

SOI 文件头	……	DHT 定义Huffman表	APP0 JFIF应用数据块	……	EOI 文件尾

- **IP分组压缩，流量压缩仪**
- **讨论：能唯一解码，是否能抗差错？**

01000010100101011000111001000011



第 三 章 树

- 树的有关定义
- 基本关联矩阵及其性质
- 支撑树的计数
- 回路矩阵与割集矩阵
- Huffman树
- 最短树

支撑树不唯一

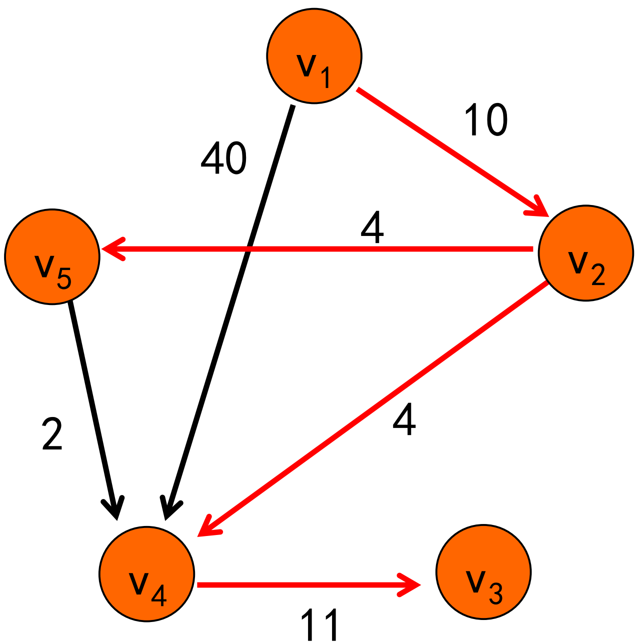
计数？ 最优树？？

最短树

- **铺设输油管道**
  - 已知任两个加油站之间输油管道的铺设费用，要让每个站都能供应上油，怎么铺？
- **最短树和最长树问题**
  - 赋权连通图中，总长最小的支撑树叫最短树
  - 在赋权连通图中，求总长最小或最大的支撑树

需求驱动！！！

曾学过最短*？**

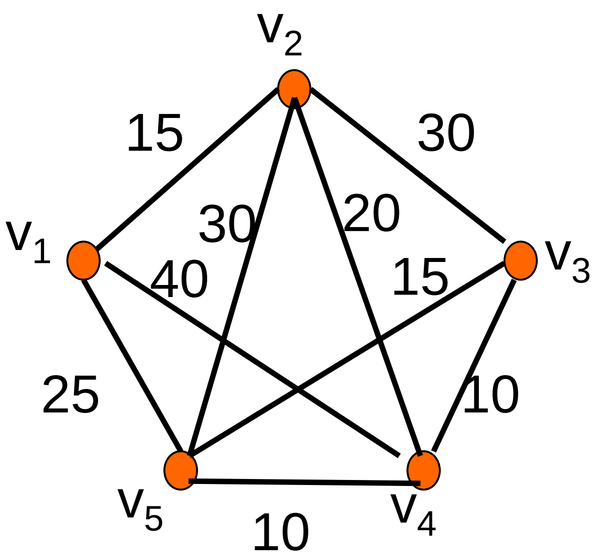


最短路径（树）

最短树(2)

- 最短树求解基本思路（Kruskal算法）
  - 不断往边集T中加入全图最短边
  - 如果此时会构成回路，那么它一定是这个回路中的最长边，删之
  - 直至最后达到n-1条边为止
  - 这时T中不包含任何回路，因此构造出最短树。

“AI”



- 最短路径
  - Dijkstra算法：每次加入到根结点最近的新结点
- 旅行商问题（便宜算法）
  - 不断加入距当前初级回路最近结点的启发式算法

最短树(3)

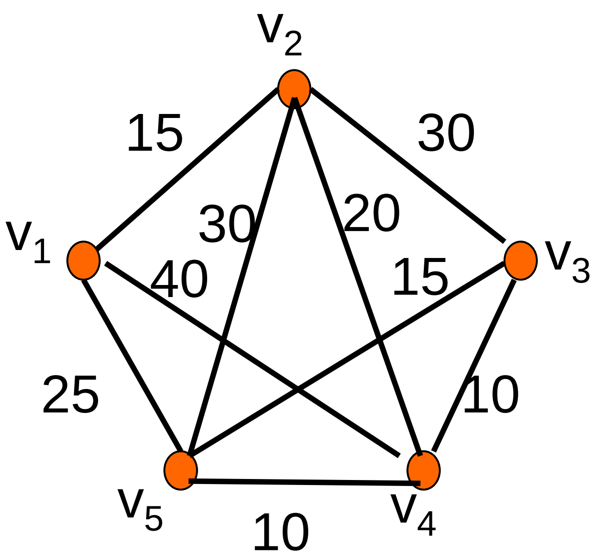
- Kruskal算法

1. T ← Φ
2. 当|T|<n-1且E≠Φ时，
3. Begin （迭代）
4. e ← E中最短边
5. E ← E-e
6. 若T+e无回路，则T ← T+e
7. end
8. 若|T|<n-1，则G非连通，否则输出最短树

算法执行过程中，不需要保证T是连通子图

最短树(4)

- 例3.7.1
  - 执行Kruskal算法的过程
    - T←(v_4, v_3)， (v_4, v_5)， (v_1, v_2)
    - 当加入(v_3, v_5)时构成回路，因此边(v_3, v_5)不加入T
    - 此后T ←T+ (v_2, v_4)
    - 这时|T|=n−1, T={(v_4, v_3)， (v_4, v_5)， (v_1, v_2)， (v_2, v_4)}， 结束



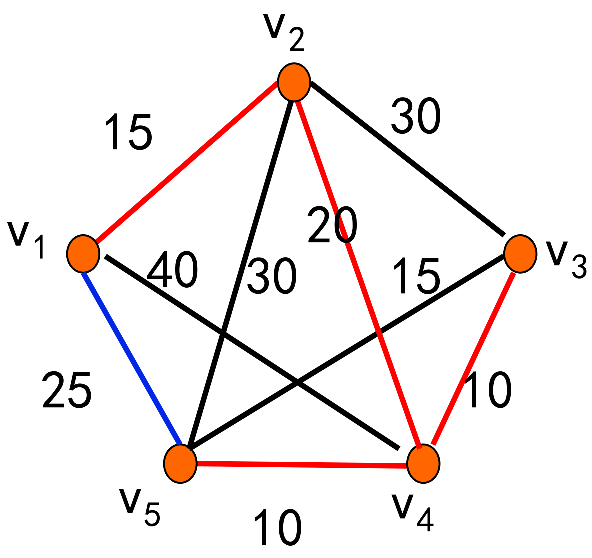
最短树(5)

- **Kruskal算法计算复杂度**
  - Kruskal算法的计算复杂性主要取决于步骤4和6
  - 对m条边的权采用堆结构存放，可以保证根结点是当前的最小权。
    - 堆结构是一种均衡二叉树，它满足对于任何一个父亲结点，其权都小于其左右儿子的权。
    - 建堆的计算复杂性是O(m)
  - 步骤4找最短边的计算复杂性是O(logm)
  - 步骤2-6迭代次数p
- **定理3.7.2**
  - Kruskal算法计算复杂性是O(m+plogm)

最短树(6)

最短树的判定和性质

- **定理3.7.1**
  - T=(V, E’)是赋权连通图G=(V, E)的最短树，当且仅当对任意的余树边e∈E−E’，满足其边权w(e) ≥w(a), 其中a为对应回路 $C^e(C^e \subseteq E'+e)$ 的任意树枝边a∈C^e（a≠e）



- **证明**
  - 必要性：（反证法）
    - 如果存在一条余树边e，满足w(e)<w(a), a∈C^e
    - 则T⊕{a, e}得到新树T’比T更短，与T是最短树矛盾

最短树(8)

- Kruskal算法
  - 不断的往T(非连通子图)中加入当前的最短边e，直到T中包含n-1条无回路的边
  - T是最短边的集合（构造过程不保证连通性）
- Prim算法?
  - 首先初始化集合V’为任选一结点v_0
  - 然后不断在V-V’中选一条距离V’中任意点(如点v)最近的节点u进入树T（连通子树），并令V’=V’+u,直至V’=V

最短树(9)

- **算法描述(初选v_1)：**

1. t ← v_1， T ← Φ ，U ←{t} //T为部分建成的连通子树，
   U为T的结点集，t为代表当前T的虚拟结点
2. while U≠V do // V为原图结点集
   begin
3. $w(t,u)=\min\limits_{v\in V-U}\{w(t,v)\}$ //找：到T的最近结点u
4. T ← T+e(t, u)
5. U ← U＋u
6. for v ∈ V-U do
7. w(t, v) ← min{w(t, v)， w(u, v)}
   //更新剩余各结点v到T的最短距离

   end

Prim算法的计算复杂度是多少？

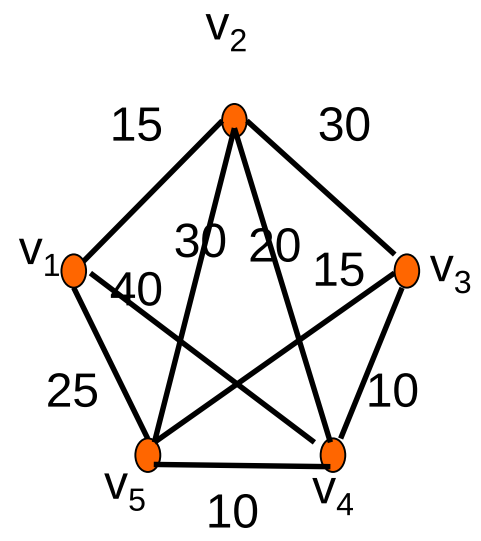
$O(n^2)$

每次查找连接两个点集的最短边

不是到树根的最短距离！

最短树(10)

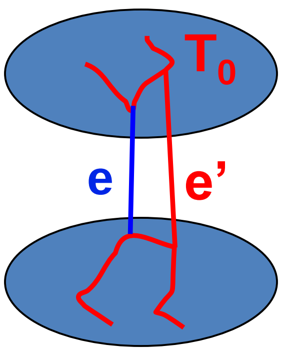
- **例3.24(设首选$U=\{v_1\}$)**
  - 3. $\min\{w(v_1,v_i)\}=w(v_1,v_2)=15$, $U=\{v_1\}+v_2$
  - 6. $w(t,v_3)=w(v_2,v_3)=30$, $w(t,v_4)=w(v_2,v_4)=20$, $w(t,v_5)=w(v_1,v_5)=25$
  - 3. $\min\{w(t,v_i)\}=w(v_2,v_4)=20$, $U=\{v_1,v_2\}+v_4$
  - 6. $w(t,v_3)=w(v_4,v_3)=10$, $w(t,v_5)=w(v_4,v_5)=10$
  - 3. $\min\{w(t,v_i)\}=w(v_4,v_3)=10$, $U=\{v_1,v_2,v_4\}+v_3$
  - 6. $w(t,v_5)=w(v_4,v_5)=10$
  - 3. $\min\{w(t,v_i)\}=w(v_4,v_5)=10$, $U=\{v_1,v_2,v_4,v_3\}+v_5=V$
  - 结束
  - 因此最短树$T=\{(v_1,v_2),(v_2,v_4),(v_4,v_3),(v_4,v_5)\}$



每次加入两个点集间的最短边，而不是便宜算法回路

最短树(11)

- 定理3.7.3
  - 设V’是赋权连通图G=(V,E)的结点真子集，e是两端点分跨在V’和V-V’的最短边，则G中一定存在包含e的最短树T
  - 证明(构造法，注意最短树不唯一)：
    - 设T_0是G的一棵最短树，若上述e不属于T_0，则T_0+e构成唯一回路
    - 该回路一定包含e和至少一条分跨在V’和V-V’的边e’=(u,v)，其中u∈V’，v ∈V-V’
    - 由已知条件w(e)≤ w(e’)，作$T_0 \oplus \{e, e'\}$，得到的仍然是最短树
    - 因此G中存在包含e的最短树T

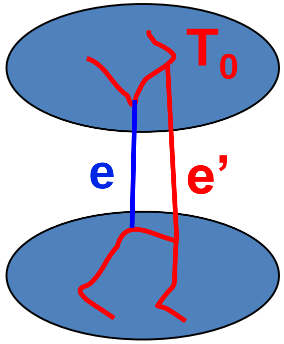


最短树(12)

- **定理3.7.4**
  - Prim算法的结果是赋权连通图G的一棵最短树
- **证明:**
  - 首先证明是一棵支撑树（n-1条边和n个节点）
    - 采用归纳法，初始$U=\{v_1\}$, $T=\Phi$, 它是由U导出的树
    - 设$|U|=i$，T是U导出的树
    - 则下一次迭代时，U中增加一新结点u, T中也加入一条与u 相连的边
    - 因此T连通，有$|U|-1$条边，它是由U导出的一棵树
    - 因此最终T是G的支撑树

最短树(13)

- 证（续）
  - 再证明Prim算法产生的树T是一棵最短树
    - 设T_0 是G的一棵最短树
    - 若$T \neq T_0$，将T_0变换为Prim算法产生的T
    - 对任意的$e \in T-T_0$，Prim算法加入的每条边e都是一条连接两个节点集的最短边
    - 由定理3.7.3，对任意的$e \in T-T_0$，一定能构造最短树$T' = T_0 \oplus (e, e')$，其中$e' \in C^e \cap T_0$
    - 继续对T'如此处理，直至最终T'=T，它仍然是最短树。



最短树(14)

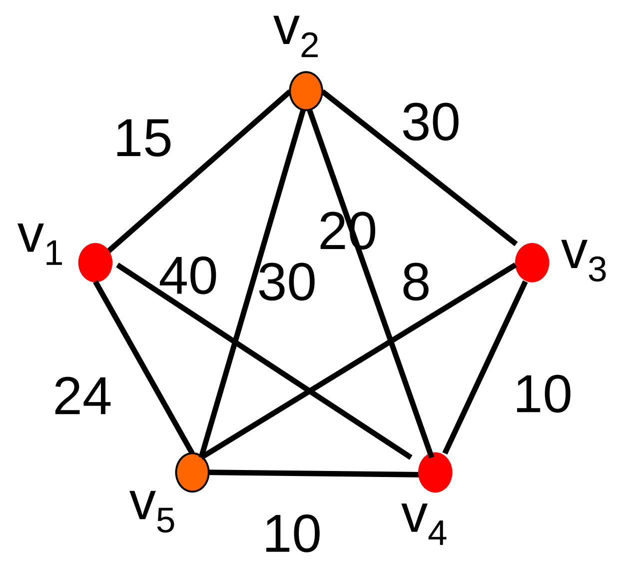
- 最短树算法怎么办？
  - Kruskal算法复杂性O(m+plogm)，与迭代次数p和边数m相关
  - Prim算法复杂性O(n^2)仅与节点数相关
  - 两个算法的适用范围？
  - 稀疏图与稠密图的不同选择

除了最短，还有什么？

- 最长树问题
  - 构造新图：给定大数减边权作为新权值
  - Kruskal算法(原为加入当前的最短边)：将加入树的边次序按边权构成非增序列
  - 最短路径->最长路径：边数不定，不适用大数减

最短树(15)

- **Dijkstra算法（最短路径树）**
- **互联网基本传输模式**
  - 单播：Unicast
  - 广播：Broadcast
  - 组播：Multicast
  - 任意播：Anycast



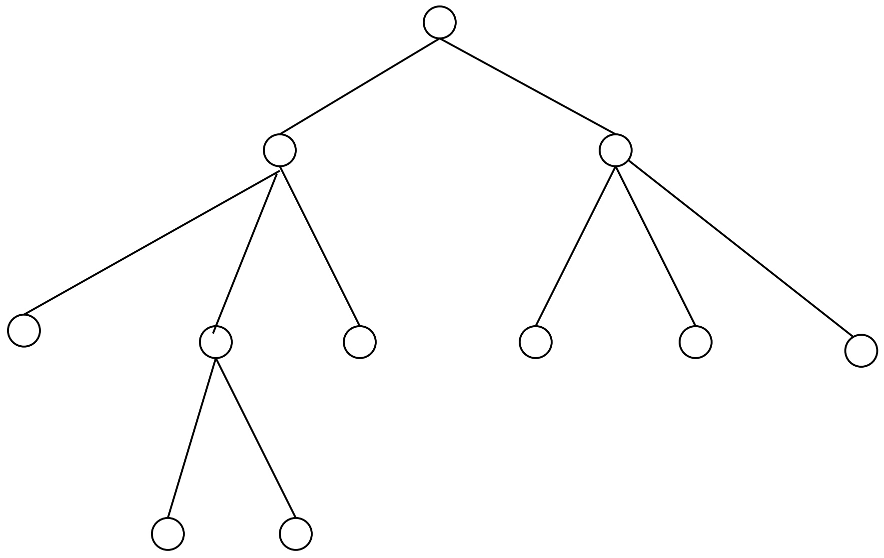
- **最优组播树(Steiner树)**
  - 不必包含所有节点而必须包含组播组成员的最短树
  - NPC问题
  - 启发式算法，满足三角不等式、节点度约束

启发式图搜索

- 优先扩展"最佳"节点
- 利用知识来引导搜索，达到减少搜索范围，提高搜索效率的目的。
- 启发信息的强度
  - 强：降低搜索工作量，但可能导致找不到最优解
  - 弱：一般导致工作量加大，极限情况下变为盲目搜索，但可能可以找到最优解

基本思想

- 引入启发知识，在保证找到最佳解的情况下，尽可能减少搜索范围，提高搜索效率。
- 定义一个评价函数f，对当前的搜索状态进行评估，找出一个最有希望的节点来扩展。



启发式搜索算法A（A算法）

- 评价函数的格式：

$$f(n) = g(n) + h(n)$$

f(n)：评价函数

h(n)：启发函数

一个A算法的例子

2	8	3
1	6	4
7		5

$\Rightarrow$

1	2	3
8		4
7	6	5

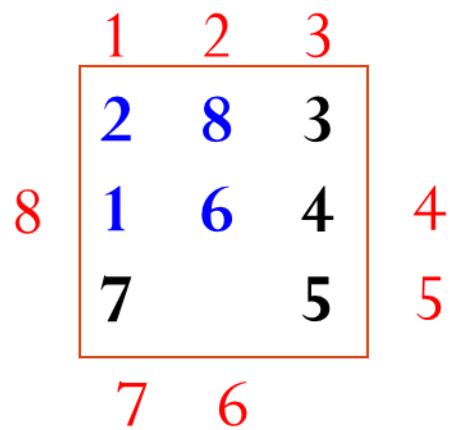
定义评价函数：

$f(n) = g(n) + h(n)$

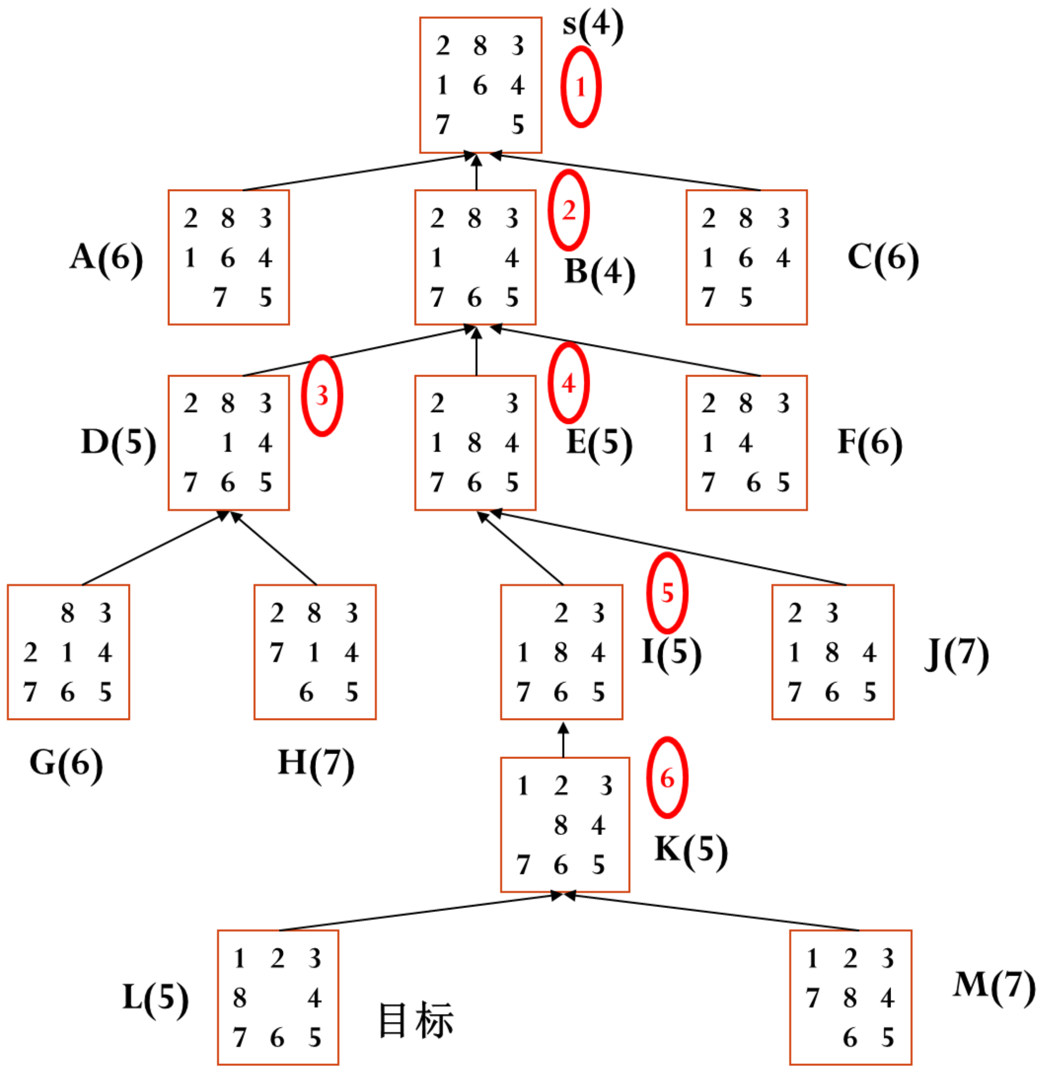
$g(n)$为从初始节点到当前节点的耗散值

$h(n)$为当前节点“不在位”的将牌数

h计算举例



h（n） =4



s(4)
2 8 3
1 6 4
7 5
1
A(6)
2 8 3
1 6 4
7 5
2
B(4)
2 8 3
1 4
7 6 5
C(6)
2 8 3
1 6 4
7 5
3
D(5)
2 8 3
1 4
7 6 5
4
E(5)
2 3
1 8 4
7 6 5
F(6)
2 8 3
1 4
7 6 5
8 3
2 1 4
7 6 5
G(6)
2 8 3
7 1 4
6 5
H(7)
2 3
1 8 4
7 6 5
5
I(5)
2 3
1 8 4
7 6 5
J(7)
6
1 2 3
8 4
7 6 5
K(5)
L(5)
1 2 3
8 4
7 6 5
目标
1 2 3
7 8 4
6 5
M(7)

Questions?