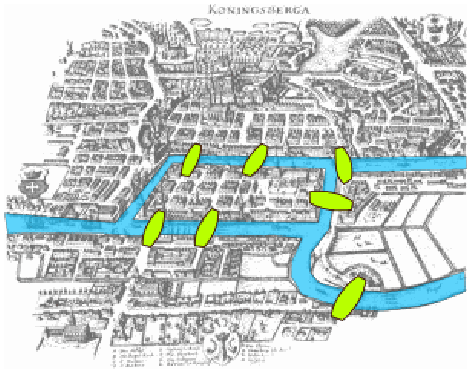
图的基本概念

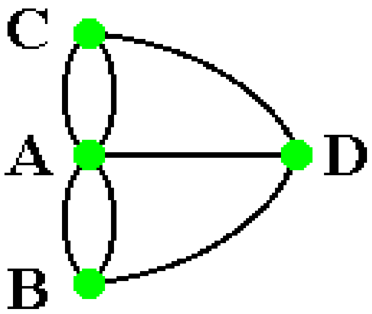
图论的研究对象

- 世界由事物组成,事物之间有联系.
- 图可以直观地描述事物及其间联系.
  - 用结点表示事物
  - 用边表示它们之间的联系
- 可见,图模型几乎可用于任何领域.
- 图论(*graph theory*)就是以这种结点和边构成的图为研究对象.

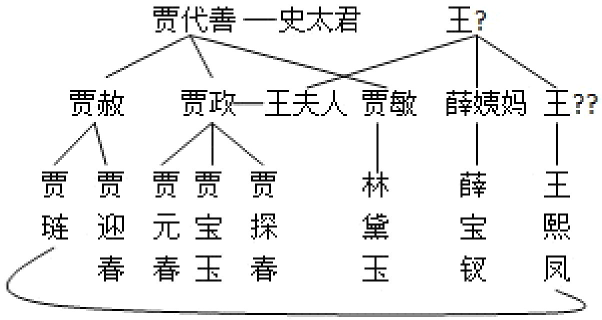
图的例子

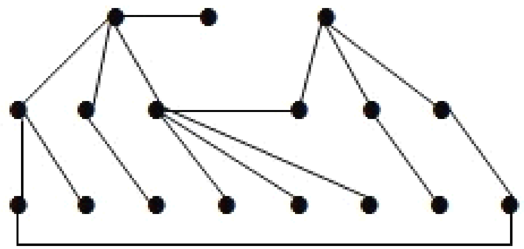
七桥问题



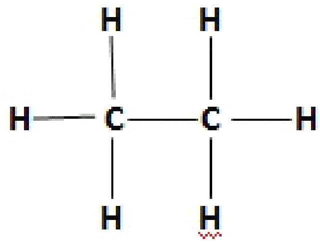


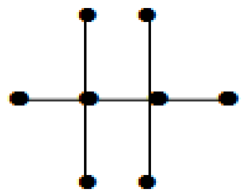
红楼家谱





乙烷



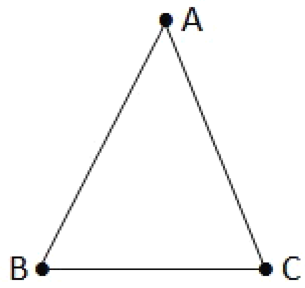


图的定义

- 定义:图(*graph*) G是一个二元组:$G=(V,E)$,其中V是非空结点(*vertex*)集合,E是边(*edge*)的集合,每条边与V中两个结点(可相同)相关联.
  - 对任意图G,约定用$V(G)$和$E(G)$表示该图的顶点集和边集.
- 例如右图G:

$V(G)=\{A,B,C\}$

$E(G)=\{AB, BC, AC\}$



有限图vs无限图

- 有限图:V和E是有限集合
- 无限图:V或E是无限集合
- 我们只讨论有限图:

$$V = \{v_1, v_2, \ldots, v_n\}$$

$$E = \{e_1, e_2, \ldots, e_m\}$$

  - e_k可记为无序或有序的顶点对(v_i, v_j).
    - 称e_k与v_i、v_j关联, v_i、v_j是e_k的端点
    - 称v_i和v_j相邻(*adjacent*或*neighbors*)
  - 以后不加说明时,都假定图有n个顶点,m条边.

无向图vs有向图

- 无向边(*undirected edge*):边无方向.
  - 对无向边$e_k = (v_i, v_j)$, v_i和v_j称为e_k的端点.
- 有向边(*directed edge*):边有方向.对$e_k = (v_i, v_j)$:
  - v_i称始点(*initial vertex*), v_j称终点(*terminal vertex*).
  - v_i是v_j的直接前趋, v_j是v_i的直接后继.
- 无向图(*undirected graph*):都是无向边.
- 有向图(*directed graph*):都是有向边.
- 混合图:既有无向边也有有向边.

简单图vs多重图

- 自环(*loop*):两端点重合的边.即$e_k = (v_i, v_i)$.
- 重边(*multiple edges*):两结点间的多条边.
- 多重图(*multigraph*):有重边的图.
- 简单图(*simple graph*):无重边无自环的无向图.
  - 空图(*null/empty graph*):无边的简单图,记作N_n.
    - 有的书定义空图是$(\varnothing,\varnothing)$.
  - 完全图(*complete graph*):任意两结点间都有边的简单图.n个顶点的完全图记作K_n.

结点的度

- 结点的度(*degree*):与结点v关联的边数,记作$d(v)$.
  - v上自环对$d(v)$的贡献为2.
  - 对有向图:$d(v) = d^+(v) + d^-(v)$
    - 正度(出度*out-degree*) $d^+(v)$ = 以v为始点的边数
    - 负度(入度*in-degree*) $d^-(v)$ = 以v为终点的边数
    - 自环对正度,负度各贡献1.
- 例如: K_n中各结点的度都为$n-1$.
- 度为0的顶点称为孤立点.

若干基本性质

(1)[握手定理] $G=(V,E)$,$|E|=m$.则 $\sum_{v\in V} d(v) = 2m$.

(2) 图G中度为奇数的结点必有偶数个.

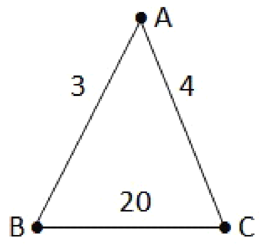
(3) 有向图中:正度之和=负度之和=边数.

(4) K_n的边数为$n(n-1)/2$.

(5) 非空简单图中一定存在度相同的结点.

赋权图

- 定义:如果给图$G=(V,E)$的每条边e_k都赋以一个实数w_k作为该边的权(*weight*),则称G是赋权图.
  - 特别地,如果权都是正数,称为正权图.
- 应用中往往是赋权图.权可以表示长度、时间、费用等.
- 例:



子图

- 定义:给定$G=(V,E)$, 如果$G' = (V',E')$满足$V' \subseteq V$, $E' \subseteq E$,则称G'是G的子图(*subgraph*),记作$G' \subseteq G$.
  - 如果$V'=V$,就称G'是G的支撑(*spanning*)子图或者生成子图;
  - 如果$V' \subseteq V$且对任何v_i, $v_i \in V'$,若$e_k=(v_i, v_j) \in E$则$e_k \in E'$,则称G'是G的导出(*induced*)子图.
  - 平凡子图: G和N_n

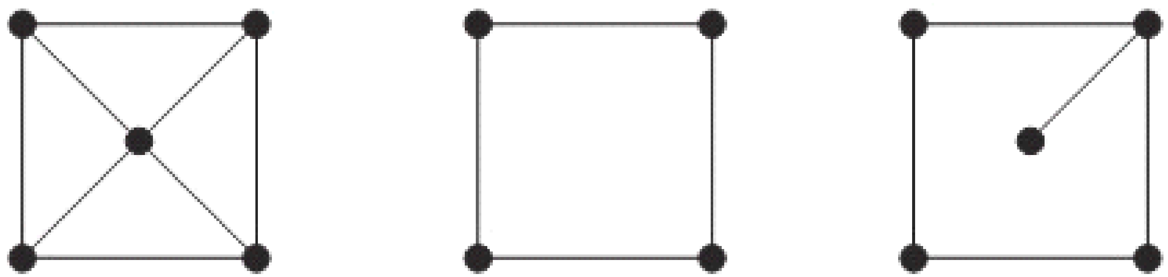
例:子图

- 下图中

  G和G'都是G的子图

  G是G的导出子图,而G'不是

  G'是G的支撑子图,而G不是



G G' G''

图的运算

- 定义$G_1=(V_1,E_1)$和$G_2=(V_2,E_2)$的
  - 并: $G_1\cup G_2=(V_1\cup V_2,\ E_1\cup E_2)$
  - 交: $G_1\cap G_2=(V_1\cap V_2,\ E_1\cap E_2)$
  - 对称差: $G_1\oplus G_2=(V_1\cup V_2,E_1\oplus E_2)$

$$=(V_1\cup V_2,\ (E_1-E_2)\cup(E_2-E_1))$$

图的运算(续)

- 若G_2是G_1的子图,则定义
  - 差: $G_1 - G_2 = (V_1, E_1 - E_2)$
  - n个结点的简单图G的补图$\overline{G}$: $K_n - G$
    - 显然:$\overline{\overline{G}} = G$
- 从G中删去结点v及其关联的边: $G - v$
  - 显然:$G - v$是G的导出子图
- 从G中删去边e: $G - e$
  - 显然:$G - e$是G的支撑子图
- 向G中增加边$e_{ij} = (v_i, v_j)$: $G + e_{ij}$

顶点的邻点

- 无向图G中,顶点v的邻点集定义为

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

- 有向图G中,顶点v的直接后继集或外邻集定义为

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

  v的直接前趋集或内邻集定义为

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

图的同构

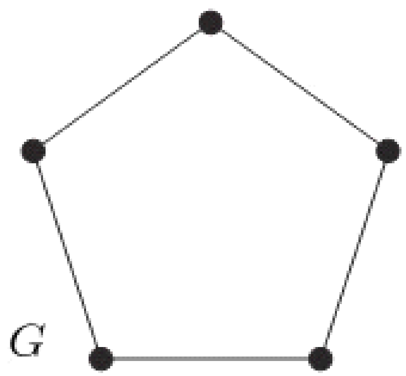
- 定义:给定两个图$G_1=(V_1,E_1)$和$G_2=(V_2,E_2)$.如果在V_1和V_2之间存在双射f使得

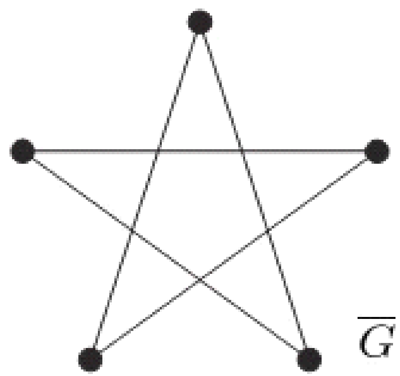
  $$(u,v)\in E_1 \ \ iff \ \ (f(u),f(v))\in E_2$$

  则称G_1和G_2同构(*isomorphic*),记作 $G_1 \cong G_2$.
- 若 $G_1 \cong G_2$,则有
  (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的任一导出子图在G_2中都有与之同构的导出子图;反之亦然.

例:同构

- 下图显示了图G与它的补图同构.





例题

证明:任意6个人中必有三人相互认识或者有三人互不相识.

证:作K_6并给边涂色:红=认识,蓝=不认识.只要证图中必有同色三角形.

v_1有5条边,由抽屜原则必有三边同色(设为红),这三边的另一顶点设为v_2, v_3, v_4.

$\triangle v_2v_3v_4$有一边为红色,则与v_1构成红色$\triangle$;

$\triangle v_2v_3v_4$的三边无红色,则构成蓝色$\triangle$.

图的表示法:邻接矩阵

- 图$G=(V,E)$的邻接矩阵(*adjacency matrix*)是一个$n\times n$矩阵A,其元素为:

$$a_{ij}=\begin{cases}1, & (v_i,v_j)\in E\\ 0, & \text{其他}\end{cases}$$

- 邻接矩阵可以表示自环,但不能表示重边.
- 对有向图: A的第i行之和是v_i的正度,第j列之和是v_j的负度.
- 对无向图: A是对称矩阵.

图的表示法:权矩阵

- 赋权图常用权矩阵表示:

$$a_{ij} = \begin{cases} w_{ij}, & (v_i, v_j) \in E \\ 0, & \text{其他} \end{cases}$$

  - 即将前面邻接矩阵的1改成权w_{ij}.
  - 可见,邻接矩阵是权矩阵的特例(所有边的权都是1).

图的表示法:关联矩阵

- 有向图的关联矩阵(*incidence matrix*)是一个$n\times m$阶矩阵B,其元素为

$$b_{ij}=\begin{cases}1, & e_j=(v_i,v_k)\in E\\ -1, & e_j=(v_k,v_i)\in E\\ 0, & \text{其他}\end{cases}$$

- 性质

  (1)每列只有两个非零元素:1和−1

  (2)第i行±1的数目是$d(v_i)$,其中1的数目是$d^+(v_i)$, −1的个数是$d^-(v_i)$.

  (3)能表示重边,但不能表示自环.

- 无向图的关联矩阵:类似,只是没有−1.

图的表示法:边列表

- 对关联矩阵的列进行压缩.
- 存储边的信息:分别用m维向量A和B存储每条边的始点编号和终点编号.如果是赋权图,再用m维向量C存储每条边的权.即:对每条边$e_k=(v_i, v_j)$,有

$$A[k]=i$$

$$B[k]=j$$

$$C[k]=w_k$$

图的表示法:正向表

- 对邻接矩阵的行进行压缩.
  - n维向量A: $A[i]$存储v_i的第一个直接后继的存储地址. ($B[A[i]]$是v_i的第一个直接后继)
  - m维向量B: 存储m个直接后继顶点的编号.
    - 同一个顶点的直接后继在B中连续存储.显然有:
    - v_i的后继: $B[A[i]], B[A[i]+1], \ldots, B[A[i+1]-1]$.
    - $d^+(v_i) = A[i+1] - A[i]$
    - $A[i] = d^+(v_1) + d^+(v_2) + \ldots + d^+(v_{i-1}) + 1$
  - 对赋权图,可再用一个m维向量C存储权值.
- 对无向图,B中存储邻点.由于v_i和v_j互为邻点,所以需要$2m$维向量.

图的表示法:逆向表

- 对邻接矩阵的列进行压缩

  n维向量A: $A[i]$存储v_i的第一个直接前趋的存储地址.

  m维向量B: 存储m个直接前趋顶点的编号.

图的表示法:邻接表

- 将正向表的m维向量B拆成n个列表,第i个列表存储v_i的所有后继.
- 列表通常采用链表结构.
- 链表中每个单元除了可以存储后继结点,还可以存储权值,形如:

End