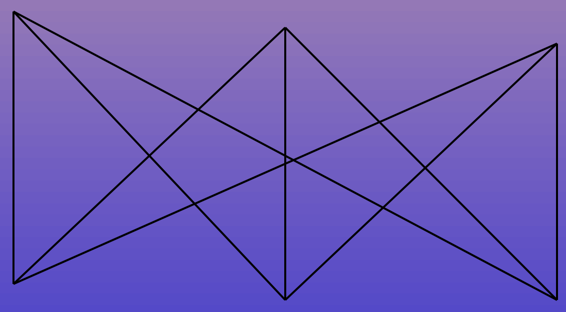
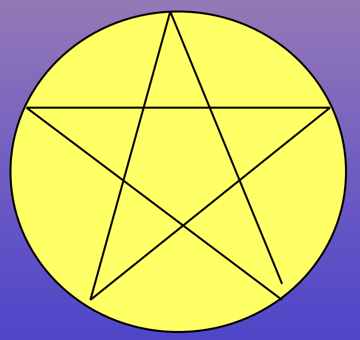
第八章

图论

Graph Theory





图论 (Graph Theory)

图论是个应用十分广泛而又极其有趣数学分支, 物理、学、生物、经济、管理科学、信息论、计算机等各个领域都可以找到图论的足迹.

历史上很多数学家对图论的形成作出过贡献, 特别要提到的欧拉 (Euler)、基尔霍夫(Kirchhoff)与凯莱(Cayley).

欧拉在1736年发表了第一篇图论的论文, 解决了著名的七桥问题. 拓扑学中著名的欧拉公式,也是图论中的重要公式.

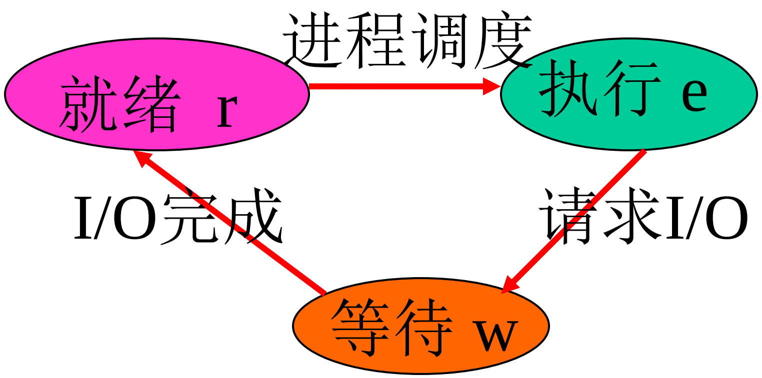
基尔霍夫对电路网络的研究(基尔霍夫定律)以及凯莱在有机化学计算中应用了树和生成树等概念.

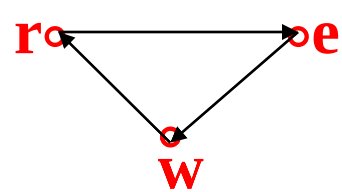
很多有趣的数学游戏也促进了图论的发展,如汉米尔顿周游世界游戏, 四色定理等, 都促进了图论的发展.

8-1. 图的基本概念

例1. 多用户操作系统中的进程状态变换图:

(进程:一个业务可以分成若干个阶段,每个阶段看成一个进程. 一个进程有三种状态.)





V={r,e,w}

E={<r,e>,<e,w>,<w,r>}

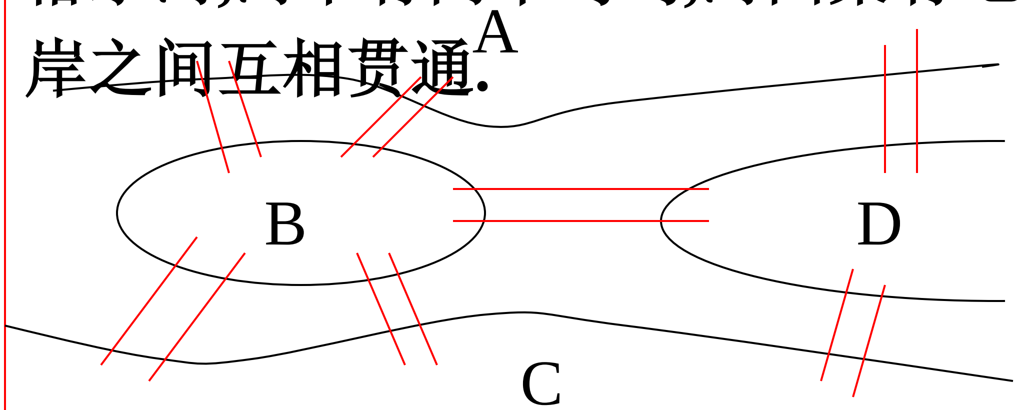
就绪状态:进程具备执行条件,因CPU少,要排队等待分配CPU.

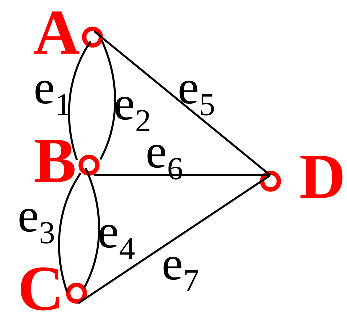
执行状态:进程已经分配到CPU,它的程序正被执行.

等待状态:进程等待某事件(如I/O完成),此时就是给它CPU也不能执行..

例2.“七桥问题” 十八世纪,哥尼斯堡城内有一条河----普雷

格尔河,河中有两个岛屿,河面架有七座桥,使得岛屿与两岸之间互相贯通.





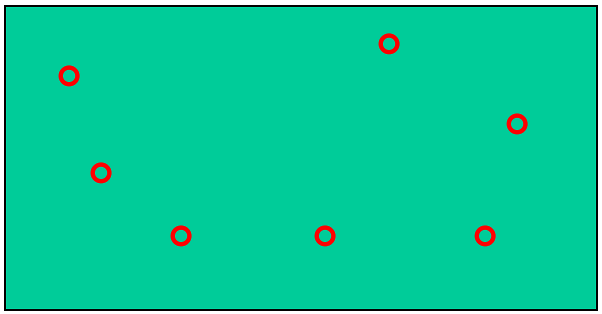
人们茶余饭后经常到桥上散步,从而提出这样问题:是否可以从某地出发，每座桥都走一次,再回到出发点. 很多人试图找出这样的路径, 都没有找到. 后来欧拉证明这样的路径根本不存在.

此图可以抽象为上边右图.

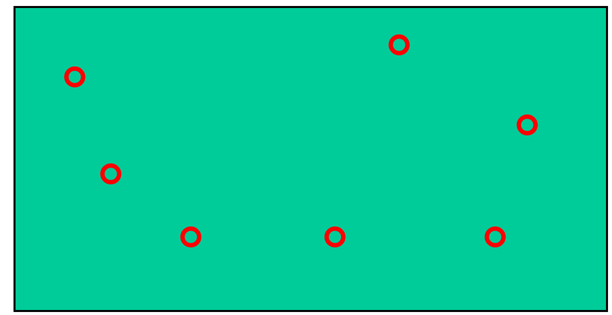
$V=\{A,B,C,D\}$　　$E=\{e_1, e_2, e_3, e_4, e_5\ e_6, e_7\}$

例3. 在机械加工中,经常需要在一块金属薄板上钻若干孔如何确定钻孔的次序,使之加工的时间最短.

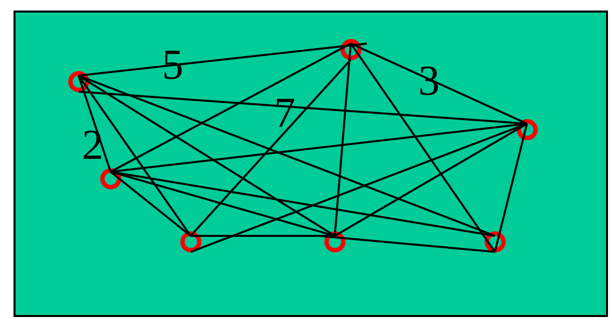
(或者是机械手在印刷电路板上安装电子元件)如图所示:



这样钻孔显然是不合适的:



这个问题可以抽象为在一个图上求从某一个结点出发,经过所有结点一次, 使得此路径最短. 如何找到此路径.



类似的: 旅行最优问题, 工程最优问题, 成本(费用)最低. ……

一．图的概念

一个图 $G=<V(G),E(G)>$，其中

V(G): 是G的结点的非空集合．(V(G)≠Φ),简记成V.

E(G): 是G的边的集合. 有时简记成E.

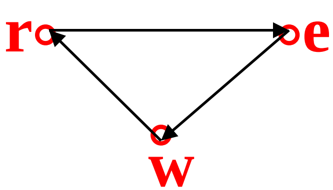
结点(Vertices): 用 ∘ 表示, 旁边标上该结点的名称.

边(Edges):

有向边: 带箭头的弧线．从u到v的边表示成 $<u,v>$

无向边:不带箭头的弧线．u和v间的边表示成 (u,v)

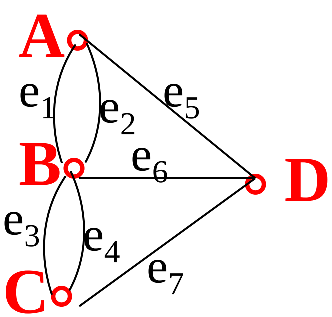
G_1 :



$V(G_1)=\{r,e,w\}$

$E(G_1)=\{<r,e>,<e,w>,<w,r>\}$

G_2 :



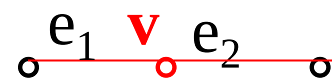
$V(G_2)=\{A,B,C,D\}$

$E(G_2)=\{e_1,e_2,e_3,e_4,e_5,e_6,e_7\}$

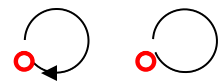
在图中, 结点的相对位置不同, 边的曲直、长短无关紧要.

- 邻接点: 与一边关联的两个结点.

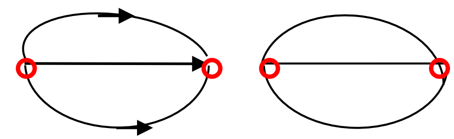
- 邻接边: 关联同一个结点的两条边.



- 环:只关联一个结点的边.



- 平行边:关联于同一对结点的若干条边.



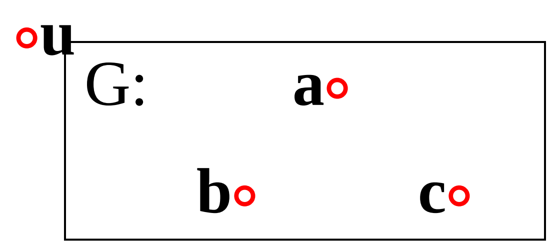
二. 有向图与无向图

有向图:只有有向边的图.

无向图:只有无向边的图.

三. 零图与平凡图

孤立结点:不与任何边关联的结点.



零图:仅由一些孤立结点构成的图.

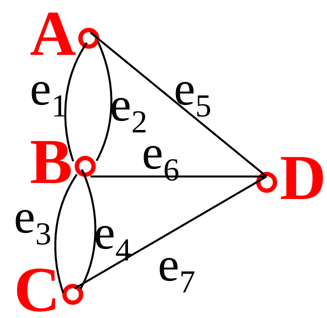
即此图的边的集合E=Φ

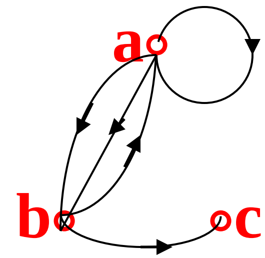
平凡图:仅由一个孤立结点构成的零图. |V(G)|=1,|E(G)|=0

四. 简单图与多重图

简单图:不含有环和平行边的图.

多重图: 含有平行边的图.

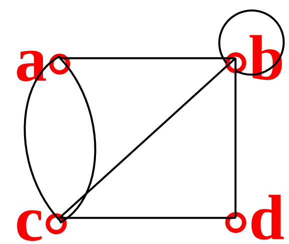




五. 无向图结点v的度(degree):

1.定义:G是个无向图, v∈V(G), 结点v所关联边数,称之为结点v的度. 记作 deg(v).(或d(v)).

deg(a)=3 deg(b)=5 deg(c)=4 deg(d)=2



一个环给结点的度是2.

2.无向图的结点度序列:

令G=<V,E>是无向图, $V=\{v_1,v_2,v_3,\ldots,v_n\}$, 则称:

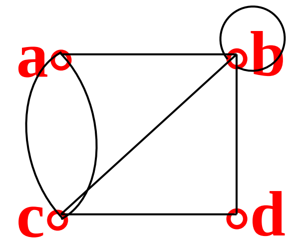
$(deg(v_1), deg(v_2),deg(v_3), \ldots,deg(v_n))$ 为图G的结点度序列.

例如上图的结点度序列为:(3,5,4,2)

3.图的最大度Δ(G)与最小度δ(G) :G=<V,E>是无向图,

$\Delta(G) = \max\{deg(v)|v\in G\}$ $\delta(G) = \min\{deg(v)|v\in G\}$

右图中$\Delta(G)=5$ $\delta(G)=2$



4. 定理8-1.1 每个无向图所有结点度总和等于边数的2倍.即 $\sum_{v\in V}\deg(v)=2|E|$

证明:因为图中每条边关联两个结点,因此每条边给予它所关联的两个结点的度各是1, 即一条边对应的度数是2, 所以整个图的度数总和为边数的2倍.

定理8-1.2(握手定理)每个无向图中,奇数度的结点必为偶数个.(在一次集会中,与奇数个人握手的人,共有偶数个.)

证明:令G=<V,E>是无向图,将V分成两个子集V_1和V_2,

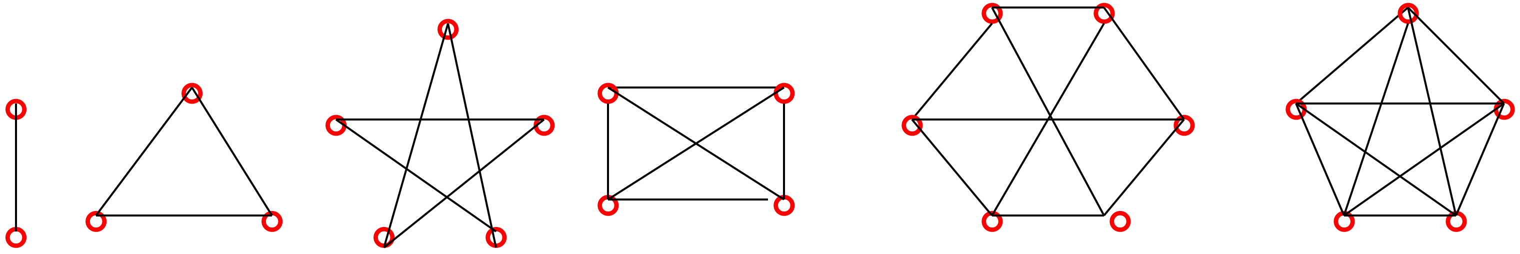
其中 V_1 ---是度数是奇数的结点集合,

V_2 ---是度数是偶数的结点集合

$\sum_{v\in V_1}\deg(v)+\sum_{v\in V_2}\deg(v)=2|E|$ 而 $\sum_{v\in V_2}\deg(v)$ 是偶数.

所以 $\sum_{v\in V_1}\deg(v)$ 也是偶数, 于是奇数度的结点数是偶数.

六. k-正则图:一个无向简单图G中,如果$\Delta(G)=\delta(G)=k$
则称G为k-正则图.



课堂练习:

1.下面哪些数的序列,可能是一个图的度数序列?
如果可能,请试画出它的图. 哪些可能不是简单图?

a) (1,2,3,4,5)

b) (2,2,2,2,2)　　　c) (1,2,3,2,4)

d) (1,1,1,1,4)　　　e) (1,2, 2,4,5)

2.已知无向简单图G中,有10条边,4个3度结点,其余结点的度均小于或等于2,问G中至少有多少个结点?为什么?

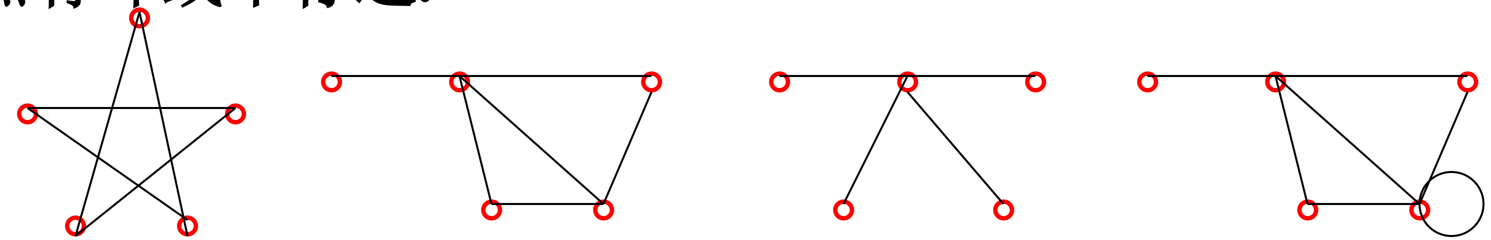
1. a) (1,2,3,4,5) b) (2,2,2,2,2) c) (1,2,3,2,4)

d) (1,1,1,1,4) e) (1,2, 2,4,5)

解:a)不是, 因为有三个数字是奇数.

b) c) d)是.

e) 不是简单图,因为它有5个结点, 有一个结点度为5, 必然有环或平行边.



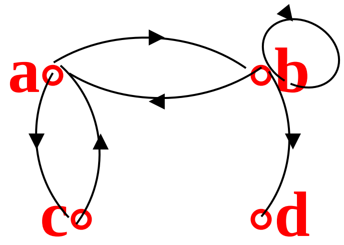
2.解:已知边数|E|=10, ∑deg(v)=2|E|=20

其中有4个3度结点, 余下结点度之和为: 20-3×4=8

因为G是简单图, 其余每个结点度数≤2, 所以至少还有4个结点. 所以G中至少有8个结点.

七. 有向图结点的出度和入度:(in degree out degree)

G=<V,E>是有向图,v∈V



v的出度: 从结点v射出的边数.

记作$deg^+(v)$ 或 dego(v)

v的入度: 射入结点v的边数. 记作$deg^-(v)$ 或 degi(v)

一个结点v的出度与入度之和，称之为结点v的度。

degi(a)=2 degi(b)=2 degi(c)=1 degi(d)=1

dego(a)=2 dego(b)=3 dego(c)=1 dego(d)=0

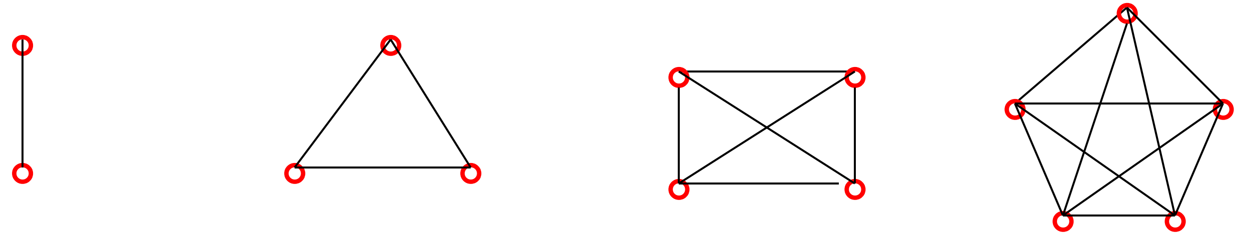
定理8-1.3 G=<V,E>是有向图, 则G的所有结点的出度之和等于入度之和.

证明: 因为图中每条边对应一个出度和一个入度. 所以所有结点的出度之和与所有结点的入度之和都等于有向边数. 必然有所有结点的出度之和等于入度之和.

八. 完全图

1.无向完全图

定义:G是个简单图, 如果每对不同结点之间都有边相连则称G是个无向完全图. 如果G有n个结点, 则记作Kn.



K_2　　K_3　　K_4　　K_5

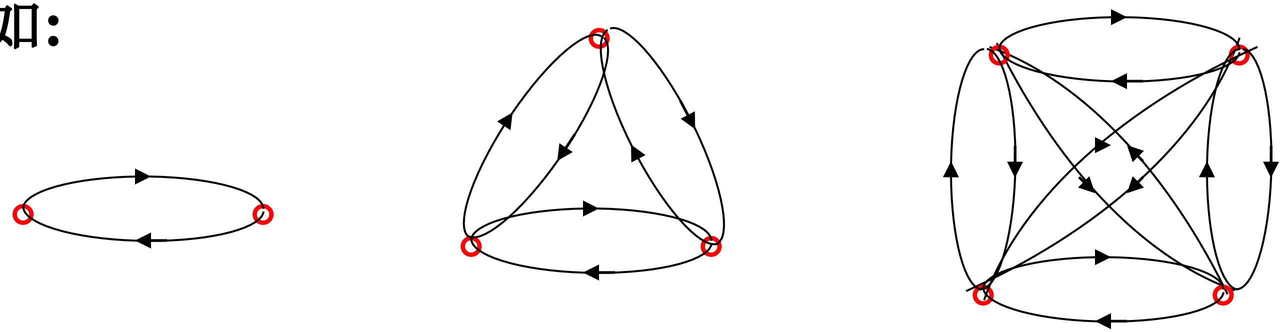
定理8-1.4 无向完全图Kn, 有边数 $\frac{1}{2}n(n-1)$

证明: 因为Kn中每个结点都与其余n-1个结点关联, 即每个结点的度均为n-1, 所以Kn的所有结点度数总和为n(n-1), 设边数为|E|, 于是n(n-1)=2|E| 所以|E|= $\frac{1}{2}n(n-1)$

2. 有向图的完全图 (注:这里的定义与教材不同)

1).有向简单完全图:G是个有向简单图,如果任何两个不同结点之间都有相互可达的边,则称它是有向简单完全图.

例如:



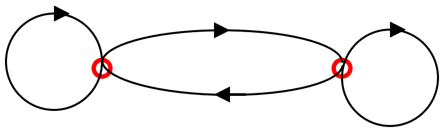
定理8-1.5: 有n个结点的有向简单完全图有边数为n(n-1).

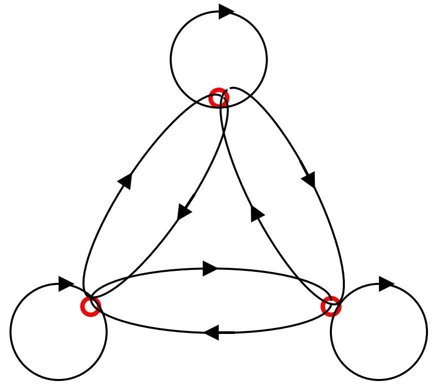
证明: 显然它的边数是K_n边数的2倍.所以是n(n-1).

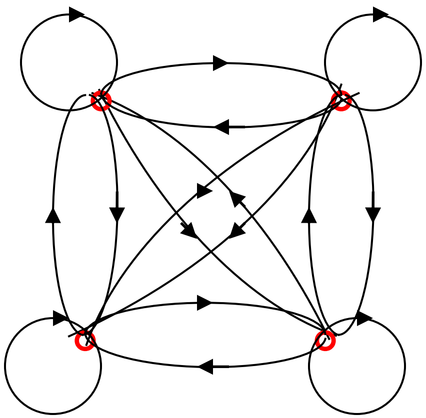
2).有向完全图(有向全图) (它与完全关系图一致)

G是个有向图如果任何两个结点之间都有相互可达的边,则称它是有向完全图.

其图形如下:



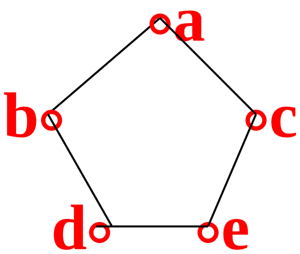




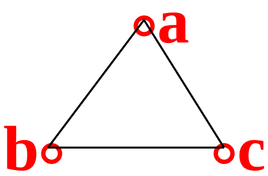
所以有n个结点的有向完全图, 有边数 n^2.

九.子图和生成子图

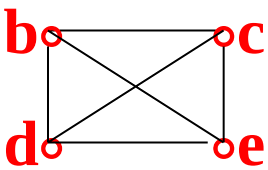
1.子图:设$G=\langle V,E\rangle$是图,如果$G'=\langle V',E'\rangle$且$V'\subseteq V$, $V'\neq\Phi$, $E'\subseteq E$, 则称G'是G的子图.



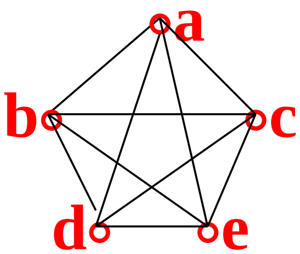
G_1



G_2



G_3



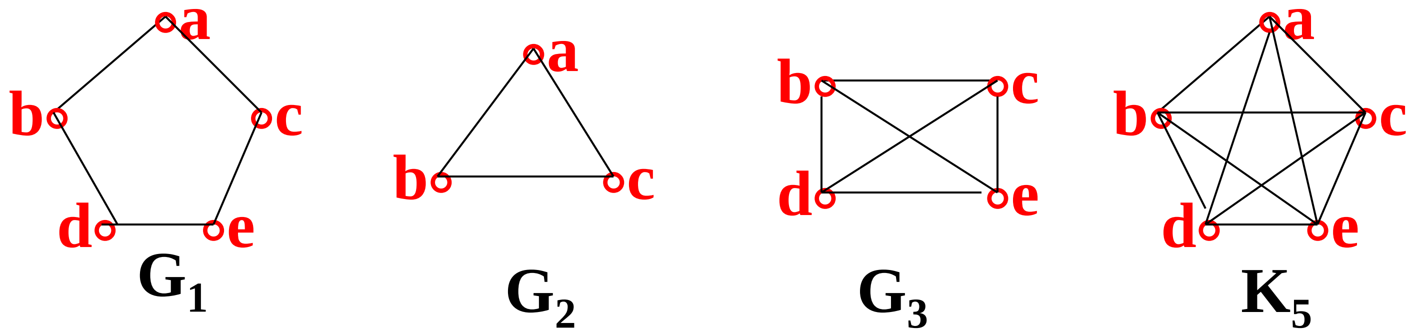
K_5

可见G_1,G_2,G_3都是K_5的子图.

2. 生成子图（支撑子图）

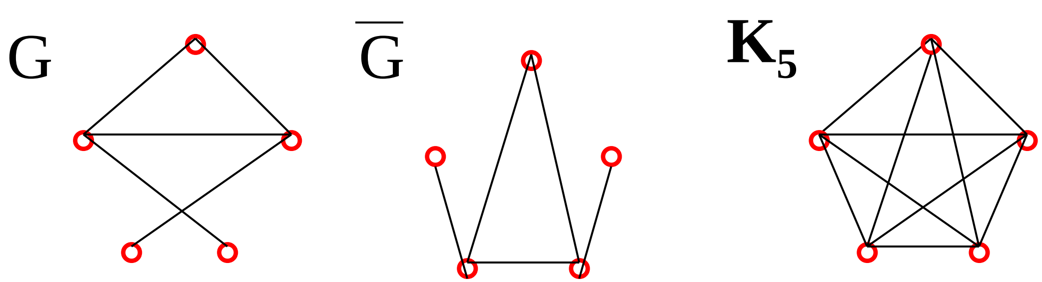
设G=<V,E>是图, G’=<V’,E’>, G’是G的子图,如果V’=V, 则称G’是G的生成子图。

上例中, G_1是K_5的生成子图。



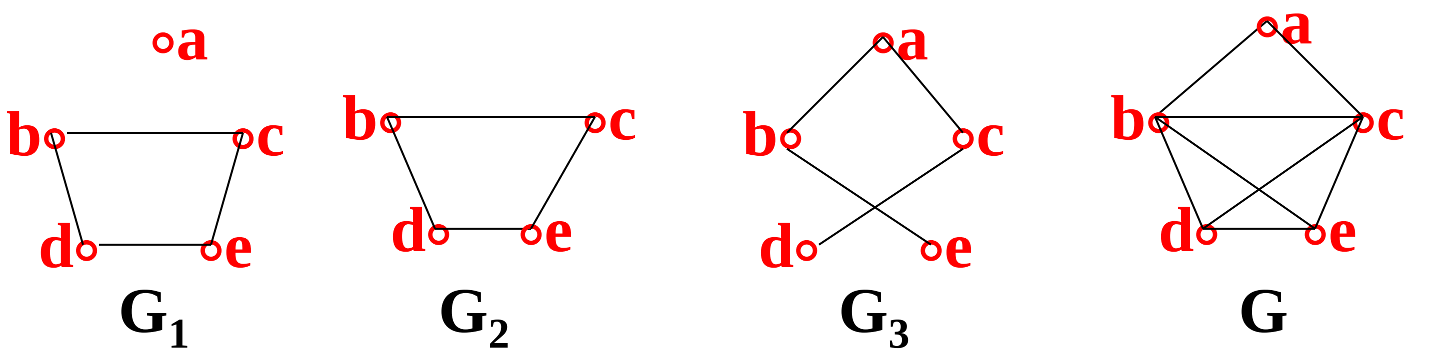
十. 补图

由G的所有结点和为使G变成完全图,所需要添加的那些边组成的图, 称之为G相对完全图的补图,简称G的补图, 记作 $\overline{G}$ 。



*十一.相对补图

设$G_1=<V_1,E_1>$是图$G=<V,E>$的子图,如果有$G_2=<V_2,E_2>$使得$E_2=E-E_1$且V_2中仅包含E_2中的边所关联的结点,则称G_2是G_1相对G的补图.



可见G_2是G_3相对G的补图. G_3也是G_2相对G的补图.

而G_1不是G_3相对G的补图(多了一个结点).

但是G_3是G_1相对G的补图.

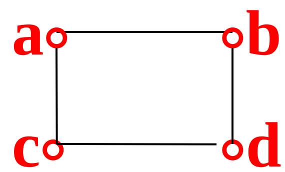
可见: 相对补图无相互性.

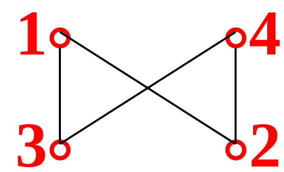
十二. 图的同构

设G=<V,E>和G'=<V',E'>是图,如果存在双射$f:V\rightarrow V'$ 且任何$v_i,v_j\in V$,若边$(v_i,v_j)\in E$,当且仅当 边$(f(v_i),f(v_j))\in E'$,(或若边$\langle v_i,v_j\rangle\in E$,当且仅当 边$\langle f(v_i),f(v_j)\rangle\in E'$),则称G与G'同构,记作$G\cong G'$. (同构图要保持边的"关联"关系)

例如:右边所示的两个图:

G=<V,E> G'=<V',E'>





构造映射$f:V\rightarrow V'$

$a\rightarrow 1$
$b\rightarrow 2$
$c\rightarrow 3$
$d\rightarrow 4$

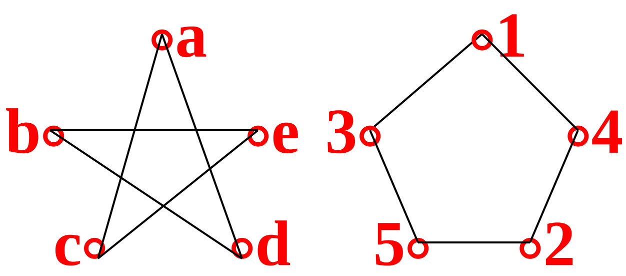
$a\rightarrow 1$
$b\rightarrow 2$
$c\rightarrow 3$
$d\rightarrow 4$

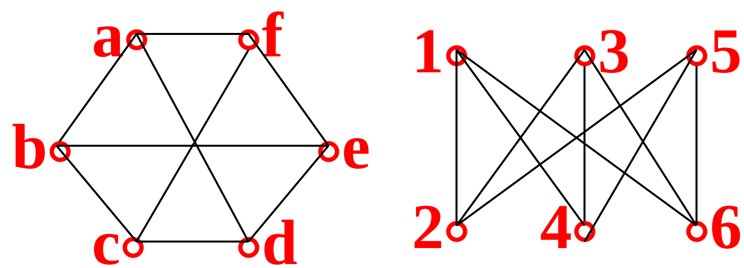
两个图同构的必要条件:

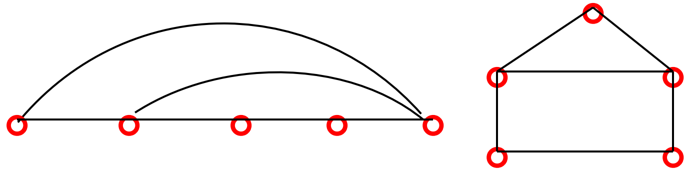
1.结点个数相等. 2.边数相等.

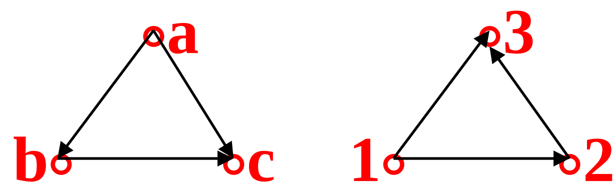
3.度数相同的结点数相等. 4. 对应的结点的度数相等.

下面是同构的图:

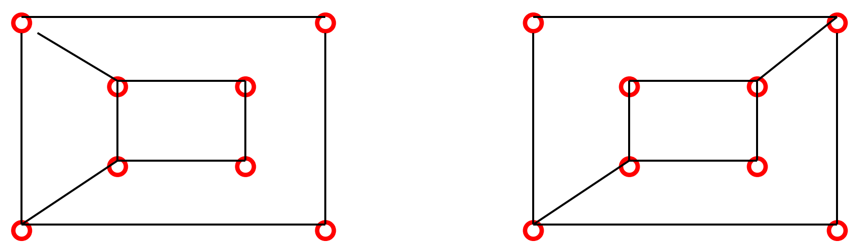






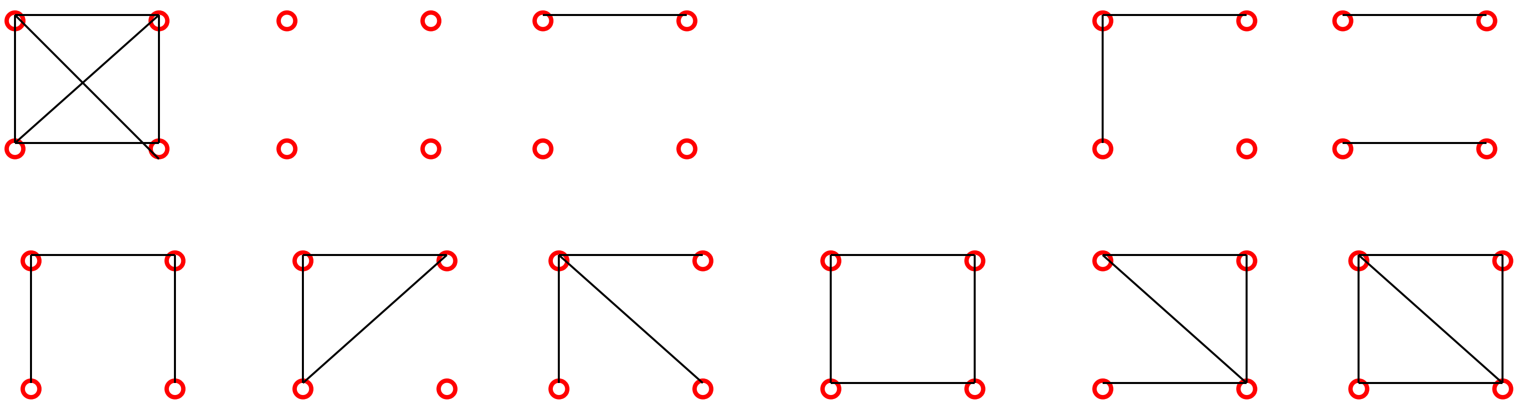


右面两个图不同构:
左图中四个3度结点
构成四边形,而右图,
则不然.



课堂练习:请画出K_4的所有不同构的生成子图.

练习:请画出K_4的所有不同构的生成子图.



本节要求:准确掌握如下基本概念和定理:

1.有向边,无向边,孤立结点,平行边,环.

2.有向图,无向图,零图,平凡图,简单图,多重图,完全图,子图,生成子图,补图,相对补图

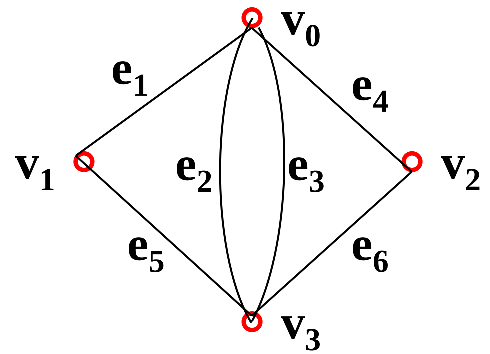
3.四个定理(关于结点度,以及结点度与边数关系)

4.图的同构 (会判断).

作业: **P279 (1) (2) (4) (5)**

8-2. 路与回路

在实际应用中,比如在市内乘出租车去参观一个博览会,一定要司机选一条最短的路. 到博览会后, 最好选一条这样到路径,使得每个展台都参观一次后,再回到原来存包处. 这就是路与回路的问题.



一. 路的概念

1.路的定义: 给定图G=<V,E >

设$v_0, v_1, v_2, \ldots, v_n \in V$, $e_1, e_2, \ldots, e_n \in E$

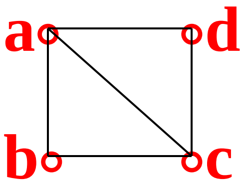
其中e_i是关联v_{i-1}, v_i的边, 则称结点和边的交叉序列

$v_0 e_1 v_1 e_2 v_2 \ldots e_n v_n$是连接$v_0$到$v_n$的路. v_0是此路的起点,v_n是此路的终点. 路中含有的边数 n称之为路的长度.

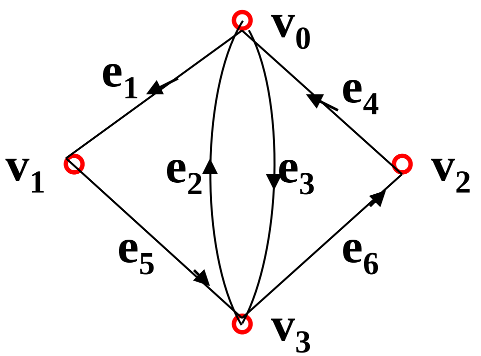
例如上图中: $v_0 e_2 v_3 e_6 v_2$是一条长度为2的路.

如果图是个简单图, 则路可以只用结点序列表示.

如右图中, 路:abcad



如果图是个有向图, 则路可以只用边序列表示.

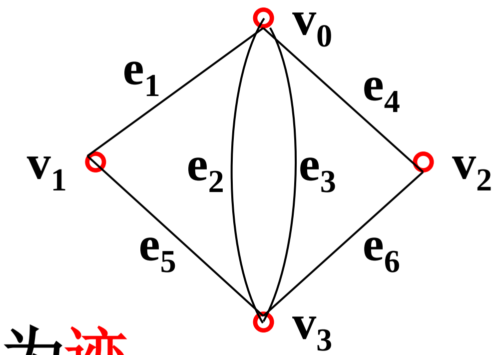


如有向图中 $e_1\, e_5 e_2 e_3\, e_6$ 是一条路.

2. 回路:如果一条路的起点和终点是一个结点,则称此路是一个回路.

如右图中的 $L_1=v_0\, e_1 v_1\, e_5 v_3\, e_6 v_2 e_4 v_0$

$L_2= v_0\, e_1 v_1\, e_5 v_3 e_2 v_0$



3. 迹与闭迹

如果一条路中,所有边都不同,则称此路为迹.

如果一条回路中,所有边都不同,则称此回路为闭迹.

4. 通路与圈

如果一条路中,所有结点都不同,则称此路为通路.

如果一条回路中,除起点和终点外,其余结点都不同,则称此回路为圈.

例如右图中:

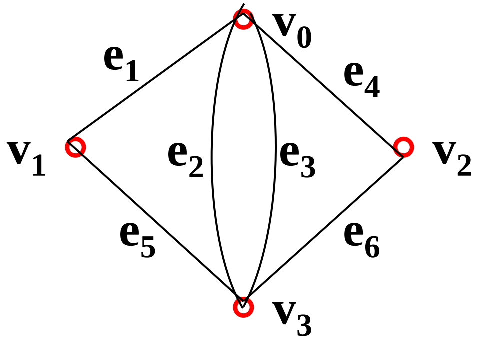
$L_1=v_0 e_1 v_1 e_5 v_3 e_6 v_2 e_4 v_0$

$L_2= v_0 e_1 v_1 e_5 v_3 e_2 v_0$

$L_3=v_0 e_1 v_1 e_5 v_3 e_2 v_0 e_3 v_3 e_6 v_2 e_4 v_0$

L_1和L_2是闭迹, 也是圈.

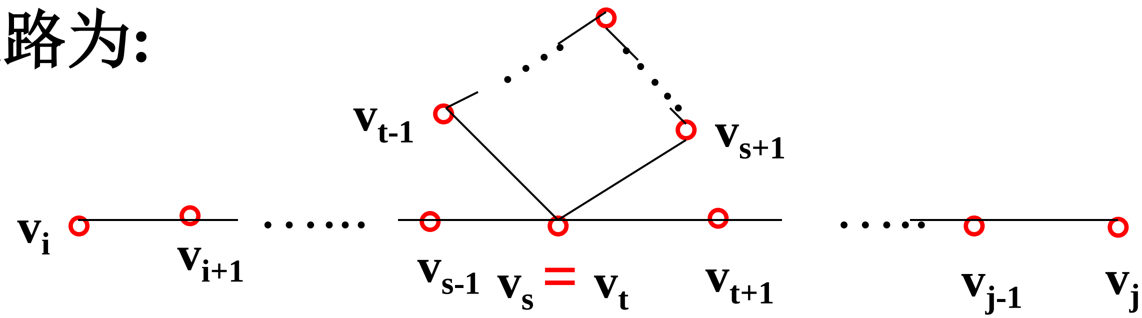
L_3是闭迹,而不是圈.



定理8-2.1 在一个有n个结点的图中,如果从结点v_i到v_j存在一条路,则从v_i到v_j必存在一条长度不多于n-1的路.

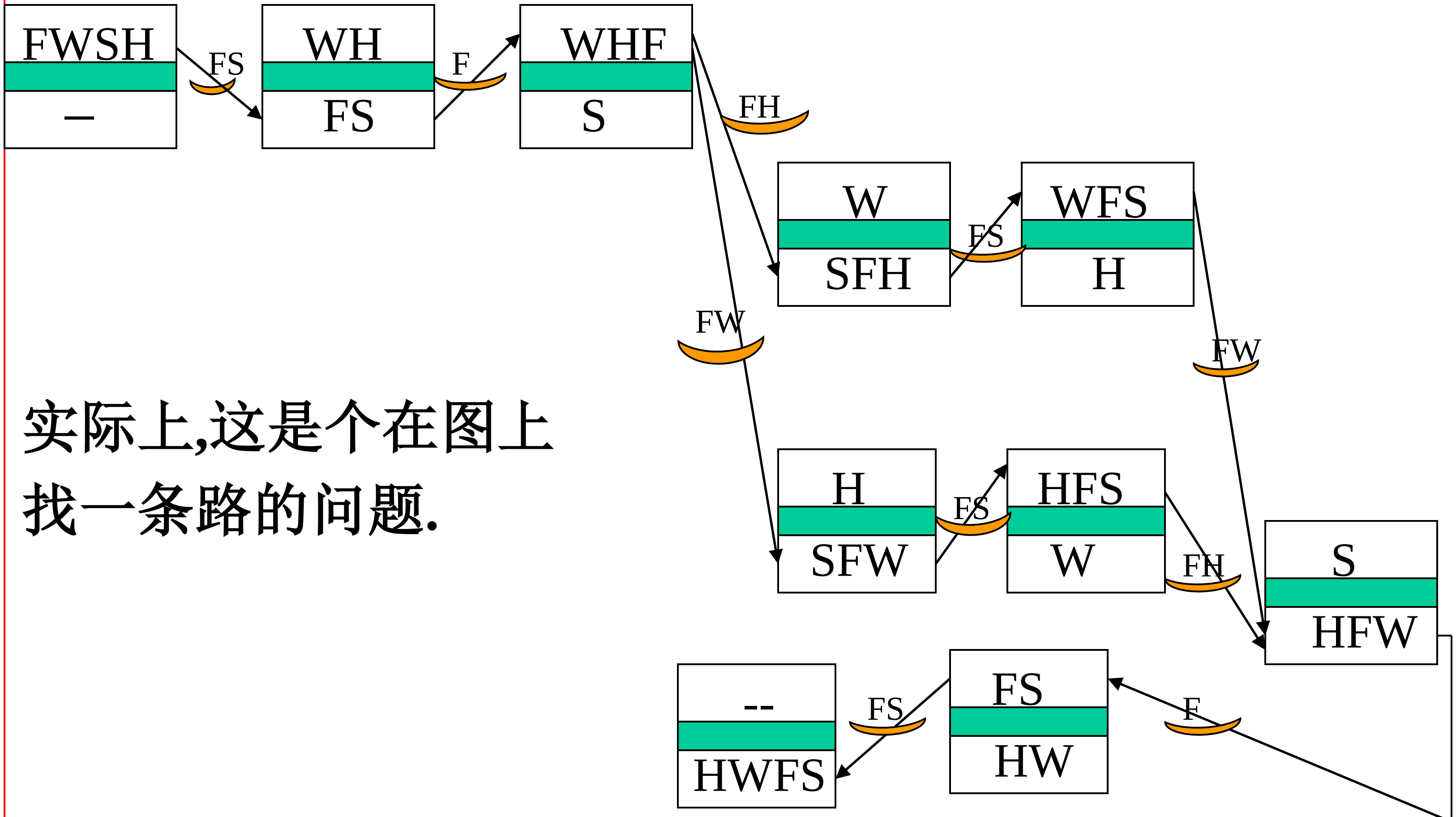
***证明**: 设v_i到v_j存在一条路: $v_i v_{i+1} v_{i+2}, \ldots, v_j$,
设此路的长度为k.

假设k>n-1, 则此路中有 k+1个结点, k+1>n, 而G中只有n个结点, 所以此路中必有两个结点相同, 假设$v_s=v_t$, (t>s)
于是此路为:



从图看出,此路中有一个从v_s到v_t的回路, 此回路中,有t-s条边(t-s>1), 如果删去这个回路, 就得到一条v_i到v_j更短的路. 如果新的路长度还大于n-1, 说明此路中还有回路,再删去回路, 如此进行下去. 最后必可找到长度小于n-1的路.

应用:摆渡人Ferryman,狼Wolf,羊Sheep,干草Hay过河问题．如何摆渡使得它们不能互相伤害.



实际上,这是个在图上找一条路的问题.

二. 无向图的连通性

1.两个结点是连通的: 在无向图中,结点u和v之间如果存在一条路, 则称u与v是连通的.

我们规定: 对任何结点u, **u与u是连通的**.

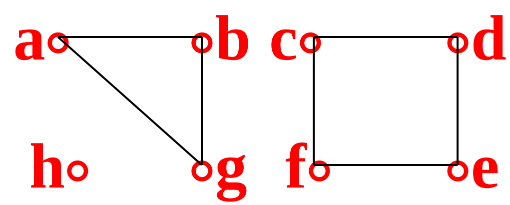
2.结点之间的连通关系是个等价关系.

令G=<V,E>是无向图, R是V上连通关系, 即

R={<u,v>|u和v是连通的}

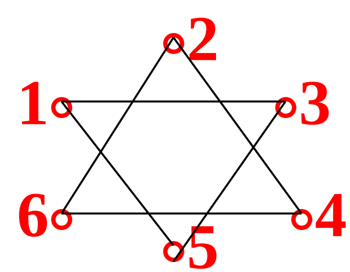
显然R具有自反、对称和传递性.

于是可以求商集V/R.



例1. 给定图G_1如右上图所示:

V/R={{a,b,g},{c,d,e,f},{h}}

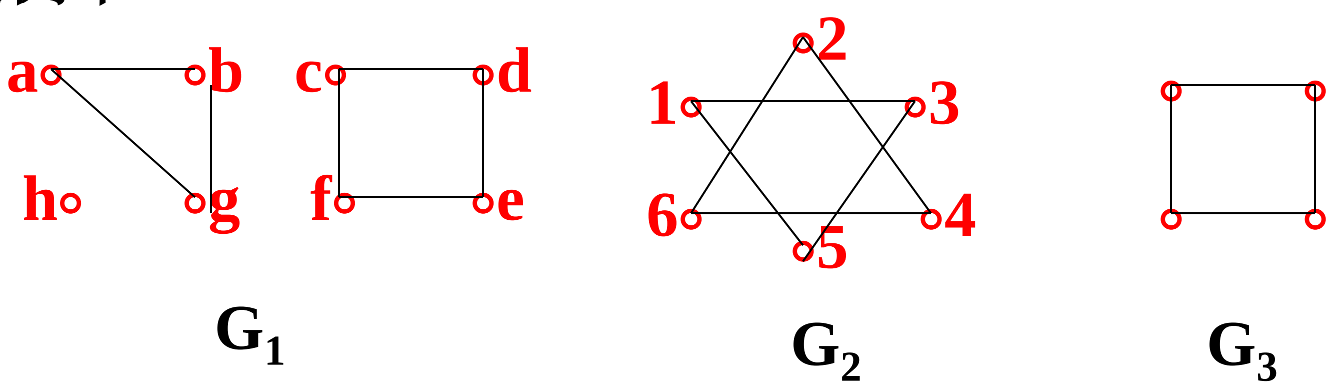


例2.给定图G_2如右下图所示:

V/R={{1,3,5},{2,4,6}}

3.连通分支:令$G=<V,E>$是无向图, R是V上连通关系, 设R对V的商集中有等价类$V_1,V_2,V_3,\ldots, V_n$,这n个等价类构成的n个子图分别记作$G(V_1),G(V_2),G(V_3),\ldots, G(V_n)$,并称它们为G的连通分支. 并用$W(G)$表示G中连通分支数.

下边例中



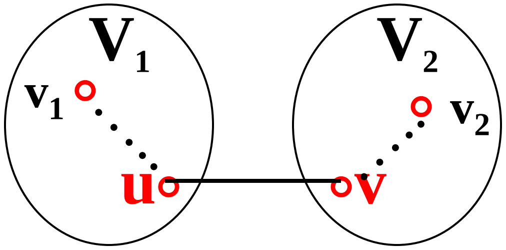
$W(G_1)=3$ $W(G_2)=2$ $W(G_3)=1$

4.连通图: 如果一个图G只有一个连通分支($W(G)=1$),则称G是连通图.

$W(G_3)=1$, G_3是连通图

定理8-2.2: 图G=<V,E>是连通的,当且仅当 对V的任何分成V_1、V_2的划分,恒存在一条边, 使得它的两个端点分别属于V_1和V_2.

*证明:必要性. 已知G是连通的. 令$\{V_1,V_2\}$是V的一个划分. 任取$v_1 \in V_1$, $v_2 \in V_2$, 由于G是连通的, 必存在一条路 $v_1 \ldots\ldots v_2$, 在此路上必存在结点u和v, 使得$u \in V_1$, $v \in V_2$,且(u,v)是此路中的一条边.

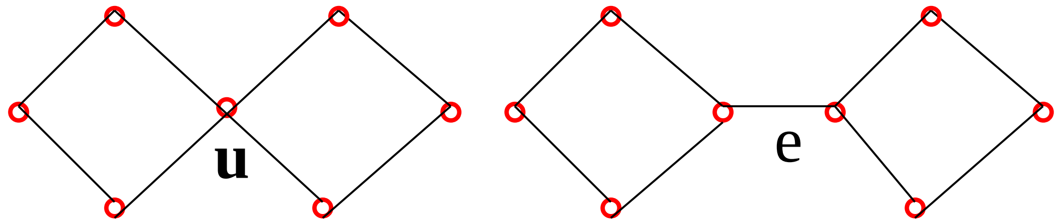


充分性:已知对V的任何分成V_1、V_2的划分,恒存在一条边, (反证法)假设G不是连通的. 则G至少有两个连通分支G_1、G_2,令$V_1=V(G_1)$ $V_2=V-V(G_1)$, 根据连通分支定义知, 不存在端点分别属于V_1和V_2的边, 与已知矛盾. 所以G是连通的.

三. 割集 (Cut Set)

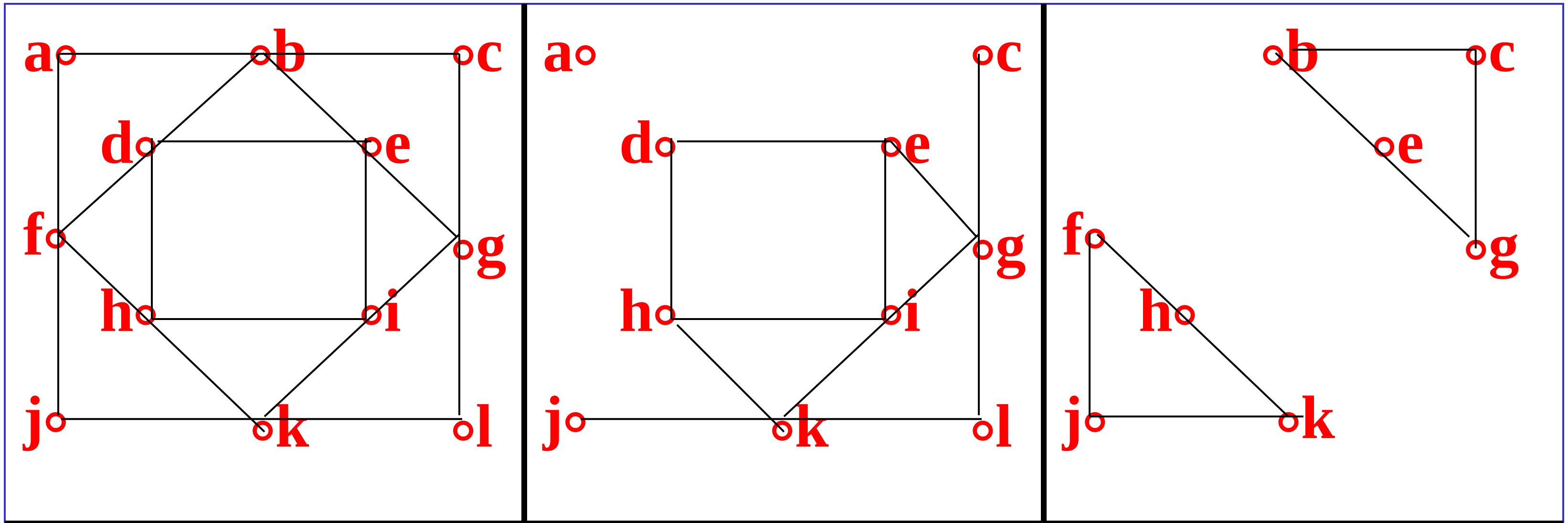
割集在图论中是个重要概念, 在图论的理论和应用中, 都具有重要地位.

比如有交通图: 结点**u**, 边**e**就是至关重要的.



割集就是使得原来连通的图, 变成不连通, 需要删去的结点集合或边的集合.

1.点割集与割点:令$G=\langle V,E\rangle$是连通无向图, 结点集合V_1 , $V_1 \subseteq V$, 如果删去V_1中所有结点后,G就变得不连通了, 而删去V_1的任何真子集中的所有结点,得到的子图仍然连通.则称V_1是G的一个点割集. 如果点割集V_1中只有一个结点, 则称此结点为割点.



右图中:{b,f}, {b,g}, {f,k},{k,g}以及{a,d,i,l}是点割集.

不存在割点.

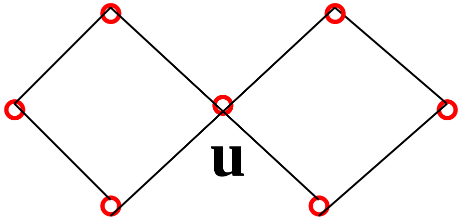
2. 点连通度:若G不是完全图, 定义:

$k(G)=\min\{\,|V_1|\mid V_1$是G的点割集$\}$ 为G的点连通度.

点连通度k(G)是表示使G不连通,至少要删去的结点数.

上例中 k(G)=2

具有割点图的点连通度 k(G)=1



定理8-2.3 :一个连通图中结点v是割点的充分且必要条件是存在两个结点u和w, 使得从u到w的任何路都通过 v .

证明:略

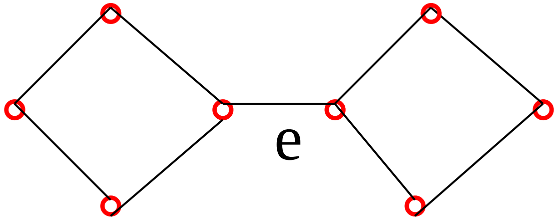
上边是通过删去结点的办法使连通图变得不连通的.

也可以通过删去边的办法使连通图变得不连通.

3. 边割集与割边(桥)

令$G=\langle V,E\rangle$是连通无向图, 边的集合E_1,$E_1 \subseteq E$, 如果删去E_1中所有边后,G就变得不连通了, 而删去E_1的任何真子集中的所有边,得到的子图仍然连通.则称E_1是G的一个边割集. 如果边割集E_1中只有一条边, 则称此边为割边, 也称之为桥.

右图中, e就是桥.



4.边连通度:若G不是平凡图, 定义:

$\lambda(G)=\min\{\, |E_1| \mid E_1$是G的边割集$\}$为图G的边连通度.

边连通度λ(G)是表示使G不连通,至少要删去的边数.

显然,如果G不是连通图, 则 $k(G)=\lambda(G)=0$

*定理8-2.4 G是无向图,则 $k(G)\le\lambda(G)\le\delta(G)$。

证明:

当G是不连通时, 显然有$k(G)=\lambda(G)=0$，所以 $k(G)\le\lambda(G)\le\delta(G)$成立。

当G是连通时:

1)先证 $\lambda(G)\le\delta(G)$

如果G是平凡图，$\lambda(G)=0\le\delta(G)$

如果G是非平凡图，因为每个结点所关联的边必包含一个边割集，所以$\lambda(G)\le\delta(G)$。

2)再证$k(G)\le\lambda(G)$

(1)如果G存在割边，即$\lambda(G)=1$，显然这时$k(G)=1$。所以$k(G)\le\lambda(G)$。

(2)如果$\lambda(G)\ge2$，则可删去$\lambda(G)$ 条边，使G不连通；而删去其中的$\lambda(G)-1$条边后，G仍然连通，此时就有一条桥$e=(u,v)$。对$\lambda(G)-1$条边中的每一条边都取一个不同于u,v的端点，把这些端点删去，则至少删去$\lambda(G)-1$条边。

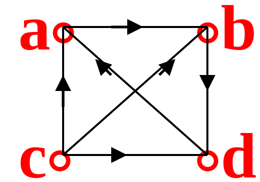
a.如果这样产生的图是不连通的，则$k(G)\le\lambda(G)-1<\lambda(G)$。 ($\lambda(G)-1$是删去的结点数)

b.如果这样产生的图仍然是连通的，则e仍是桥，此时再删去u或v，就必产生一个不连通的图，所以$k(G)\le\lambda(G)$ 。

最后由1)和2)得$k(G)\le\lambda(G)\le\delta(G)$。

四. 有向图的连通性

1.结点间的可达性: G=<V,E>是有向图, u,v∈V, 如果从u到v有一条路, 则称从u到v可达.



右图中: a可达b和d, 但是a不可达c.

显然结点间的可达关系,具有自反性和传递性.

2. 结点u到v的距离: 如果u可达v, 可能从u到v有多条路,其中最短的路的长度,称之为从u到v的距离.记作d<u,v>.

上例中 d<a,b>=1　d<a,d>=2　d<a,a>=0　d<b,c>=∞

3. 可达性的性质:

1). d<u,v>≥0　　2) d<u,u>=0

3) d<u,v> + d<v,w>≥d<u,w>　(如上图的c,a,b)

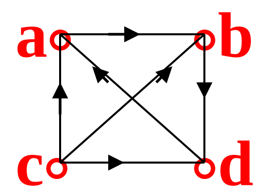
4) 如果从u到v不可达,则d<u,v>=∞ (如b,c)　(如a,d)

5)如从u可达v,从v也可达u, 但d<u,v>不一定等于d<v,u>

4. 图的直径: G是个有向图, 定义

$$D=\max_{u,v\in V}\{d\langle u,v\rangle\}$$

为图G的直径.



上图中, 图的直径$D=\infty$ (因为$d\langle b,c\rangle=\infty$)

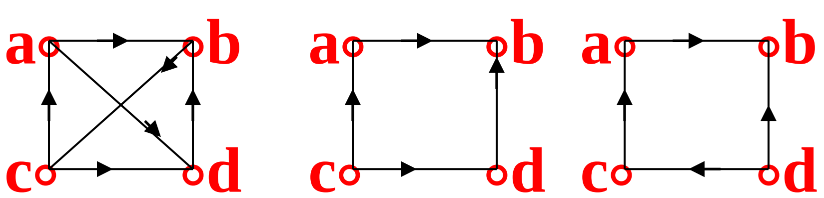
5. 强连通、单侧连通和弱连通

在简单有向图G中,如果任何两个结点间相互可达, 则称G是**强连通**. 如果任何一对结点间, 至少有一个结点到另一个结点可达, 则称G是**单侧连通**. 如果将G看成无向图后(即把有向边看成无向边)是连通的,则称G是**弱连通**.

(a)有回路adbca,强连通.

(b)a到d, d到a, 都不可达 是弱连通.

(c)单侧连通.



(a) (b) (c)

定理8-2.5:一个有向图G是强连通的,当且仅当G中有一个回路, 此回路至少包含每个结点一次.

证明:充分性: 显然成立. 因为如果G中有一个回路, 它至少包含每个结点一次, 就使得任何两个结点间相互可达, 所以G是强连通的.

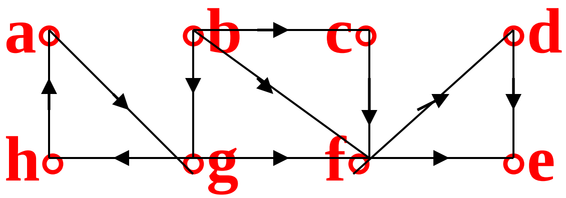
必要性:如果G是强连通的,则任何两个结点间相互可达. 所以可以构造一个回路经过所有结点. 否则必有一个回路不包含某个结点v, 所以v与回路上的各结点都不相互可达.这与G是强分图矛盾. 所以G必有回路至少包含每个结点一次.

所以可以应用此定理判断G是否为强连通, 就是看它是否有包含每个结点的回路.

6. 强分图、单侧分图和弱分图

在简单有向图中,具有强连通的最大子图,称为强分图.具有单侧连通的最大子图,称为单侧分图. 具有弱连通的最大子图,称为弱分图.

这些分图用结点的集合表示.



例如,给定有向图G,如图所示:求它的强分图、单侧分图和弱分图.

解: 强分图:由{a,g,h}{b}{c} {d}{e}{f}导出的子图.

单侧分图:由{a,g,h,b,f,d,e}{b,c,f,d,e}导出的子图.

弱分图:G本身是弱分图.

在有向图中,每个结点必位于一个且只位于一个强分图中

定理8-2.6 在有向图中,每个结点必位于一个且只位于一个强分图中.

证明:令G=<V,E>是有向图, 任取结点$v \in V$, 令S是所有与v相互可达的结点集合, 当然$v \in S$ $\therefore S \neq \Phi$, 而S是一个强分图, 所以v必位于一个强分图中.

如果v位于两个不同的强分图S_1、S_2中, 于是 v与S_1中每个结点相互可达, v也与 S_2中每个结点相互可达, 所以S_1中每个结点都与S_2中每个结点通过v相互可达, 这说明S_1与S_2是一个强分图, 与已知S_1、S_2是两个不同的强分图矛盾.

所以每个结点必位于一个且只位于一个强分图中.

在给定的简单有向图中找强分图----回路中的结点构成一个强分图, 不在回路中的结点,自己构成一个强分图.

作业 P286 (3) (5) (8)

8-3. 图的矩阵表示

图的矩阵表示不仅是给出图的一种表示方法, 还可以通过这些矩阵讨论有关图的若干性质, 更重要的是可以用矩阵形式将图存入计算机中, 在计算机中对图作处理.

这里主要讨论图的三种矩阵.

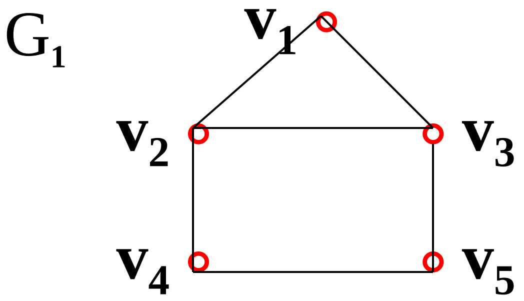
一. 邻接矩阵

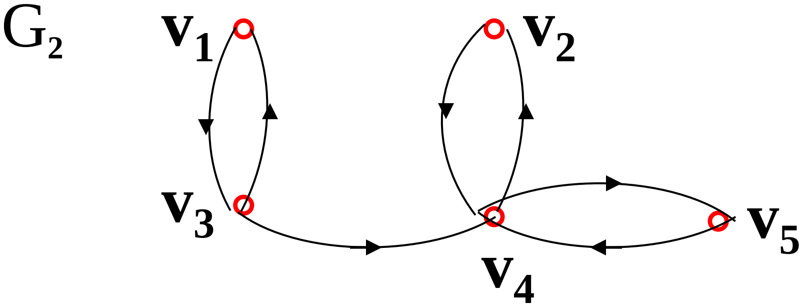
这是以结点与结点之间的邻接关系确定的矩阵.

1.定义:设G=<V,E>是个简单图,$V=\{v_1,v_2,v_3,\ldots,v_n\}$, 一个n×n阶矩阵$A=(a_{ij})$称为G的邻接矩阵. 其中:

$$a_{ij}=\begin{cases}1 & v_i与v_j邻接,\ 即(v_i,v_j)\in E\ 或<v_i,v_j>\in E\\ 0 & 否则\end{cases}$$

例如, 给定无向图G_1和有向图G_2如图所示:





$$A(G_1)=\begin{bmatrix}0&1&1&0&0\\1&0&1&1&0\\1&1&0&0&1\\0&1&0&0&1\\0&0&1&1&0\end{bmatrix}\qquad A(G_2)=\begin{bmatrix}0&0&1&0&0\\0&0&0&1&0\\1&0&0&1&0\\0&1&0&0&1\\0&0&0&1&0\end{bmatrix}$$

2.从邻接矩阵看图的性质:

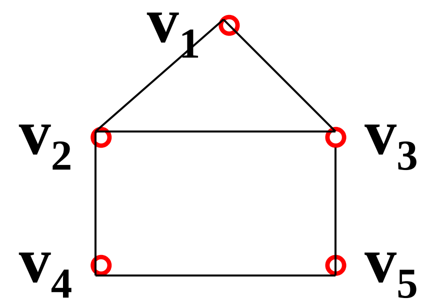
无向图: 每行1的个数=每列1的个数=对应结点的度

有向图: 每行1的个数=对应结点的出度

每列1的个数=对应结点的入度

3.邻接矩阵的乘积

$$A(G_1)=\begin{bmatrix}0&1&1&0&0\\1&0&1&1&0\\1&1&0&0&1\\0&1&0&0&1\\0&0&1&1&0\end{bmatrix}\qquad (A(G_1))^2=\begin{bmatrix}2&1&1&1&1\\1&3&1&0&2\\1&1&3&2&0\\1&0&2&2&0\\1&2&0&0&2\end{bmatrix}$$



G_1

$$(A(G_1))^3=\begin{bmatrix}2&1&1&1&1\\1&3&1&0&2\\1&1&3&2&0\\1&0&2&2&0\\1&2&0&0&2\end{bmatrix}\begin{bmatrix}0&1&1&0&0\\1&0&1&1&0\\1&1&0&0&1\\0&1&0&0&1\\0&0&1&1&0\end{bmatrix}=\begin{bmatrix}2&4&4&2&2\\4&2&6&5&1\\4&6&2&1&5\\2&5&1&0&4\\2&1&5&4&0\end{bmatrix}$$

$v_3v_2v_4$，$v_3v_5v_4$

在$(A(G_1))^2$中$a_{34}^{\ 2}=2$表示从v_3到v_4有长度为2的路有2条:

在$(A(G_1))^3$中$a_{23}^{\ 3}=6$表示从v_2到v_3有长度为3的路有6条:

$v_2v_1v_2v_3$，$v_2v_4v_2v_3$，$v_2v_3v_2v_3$，$v_2v_3v_1v_3$，$v_2v_3v_5v_3$，$v_2v_4v_5v_3$。

定理8-3.1设$G=<V,E>$是简单图,令$V=\{v_1,v_2,v_3,\ldots,v_n\}$, G的邻接矩阵$(A(G))^k$中的第 i行第j列元素$a_{ij}^k=m$, 表示在图G中从$v_i$到$v_j$长度为k的路有m条.
可以用归纳法证明.(见教材P290)

在实际应用中,有时只关心从一个结点到另一个结点是否有路,而不关心路有多长,比如电话网络. 这就促使我们定义可达矩阵.

二.可达性矩阵

1.定义:设$G=<V,E>$是个简单图,$V=\{v_1,v_2,v_3,\ldots,v_n\}$, 一个$n\times n$阶矩阵$P=(p_{ij})$称为G的可达性矩阵. 其中:

$$p_{ij}=\begin{cases}1 & v_i\text{到}v_j\text{可达, (至少有一条路)}\\ 0 & \text{否则}\end{cases}$$

2.求可达矩阵

可以根据邻接矩阵A求可达矩阵. 设$|V(G)|=n$

令$A^{(k)}$是将A^k中的非0元素都写成1,而得到的只含有0和1的0-1矩阵.于是可达矩阵P为:

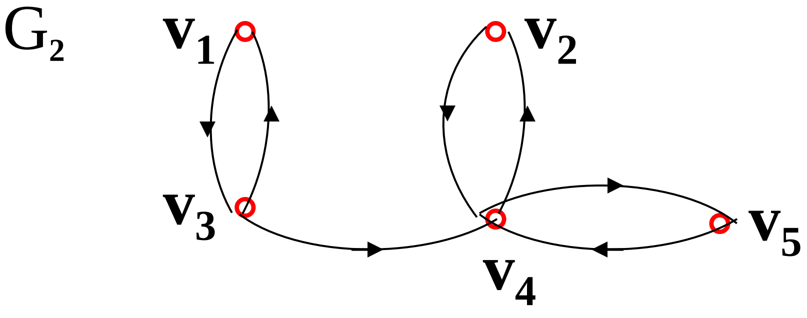
$P=A\vee A^{(2)}\vee A^{(3)}\vee\ldots\vee A^{(n)}$ 其中$\vee$是逻辑或.

有两种方法求P

方法1. 按照矩阵相乘分别求出$A^{(k)}$ $(k\geq 2)$, 然后再$\vee$.

方法2.用求传递闭包的Warshall算法,见P124.

例如,G_2如图所示, 求它的可达矩阵P.

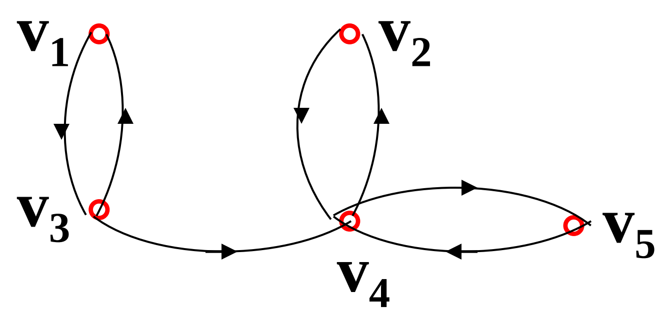


$$A=\begin{bmatrix}00100\\00010\\10010\\01001\\00010\end{bmatrix}\quad A^{(2)}=\begin{bmatrix}10010\\01011\\01101\\00010\\01001\end{bmatrix}\quad A^{(3)}=\begin{bmatrix}01101\\01011\\10010\\01000\\00010\end{bmatrix}\quad A^{(4)}=\begin{bmatrix}10010\\01011\\01101\\00010\\01001\end{bmatrix}=A^{(2)}$$

$$\mathbf{P=A\vee A^{(2)}\vee A^{(3)}\vee A^{(4)}\vee A^{(5)}}\qquad A^{(5)}=A^{(3)}$$

$$P=\begin{bmatrix}11111\\01011\\11111\\01011\\01011\end{bmatrix}$$

G_2



3.用可达矩阵求强分图.

以G_2为例,

从图看出有两个强分图:$\{v_1,v_3\}$和$\{v_2,v_4,v_5\}$

下面看怎样用P求强分图.

先将$P=(p_{ij})$转置得$P^T=(p^T_{ij})$, 如果v_i与v_j相互可达,则

$p_{ij}= p^T_{ij} =1$

$$P=\begin{bmatrix}11111\\01011\\11111\\01011\\01011\end{bmatrix}\quad P^T=\begin{bmatrix}10100\\11111\\10100\\11111\\11111\end{bmatrix}\quad P\wedge P^T=\begin{bmatrix}10100\\01011\\10100\\01011\\01011\end{bmatrix}\ \text{初等变换得}\ \begin{array}{c}v_1v_3v_2v_4v_5\\ \begin{bmatrix}11000\\11000\\00111\\00111\\00111\end{bmatrix}\end{array}$$

对$P\wedge P^T$进行初等变换, 第2行与第3行交换,再第2列与第3列交换, 最后得两个强分图:$\{v_1,v_3\}$和$\{v_2,v_4,v_5\}$

三.完全关联矩阵

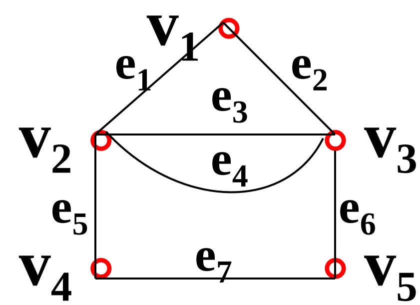
此矩阵是按照结点与边之间的关联关系确定的矩阵.

1.无向图的完全关联矩阵

1.无向图的完全关联矩阵

1).定义:设$G=<V,E>$是个无向图,$V=\{v_1,v_2,v_3,\ldots,v_m\}$, $E=\{e_1,e_2,e_3,\ldots,e_n\}$,一个$m\times n$阶矩阵$M=(m_{ij})$称为G的完全关联矩阵. 其中:

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



2).从关联矩阵看图的性质:

a)每列只有二个1.

(因为每条边只关联两个结点)

b)每行中1的个数为对应结点的度数.

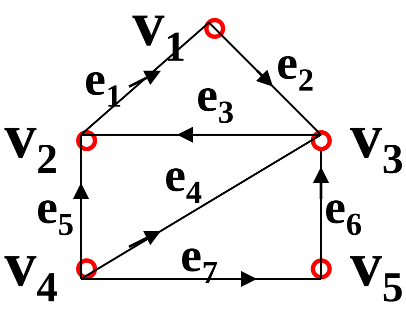
c)如果两列相同,则说明对应的两条边是平行边.

$$M=\begin{array}{c}\\ v_1\\ v_2\\ v_3\\ v_4\\ v_5\end{array}\begin{array}{c}\begin{array}{ccccccc}e_1&e_2&e_3&e_4&e_5&e_6&e_7\end{array}\\ \begin{bmatrix}1&1&0&0&0&0&0\\1&0&1&1&1&0&0\\0&1&1&1&0&1&0\\0&0&0&0&1&0&1\\0&0&0&0&0&1&1\end{bmatrix}\end{array}$$

2.有向图的完全关联矩阵

1).定义:设$G=\langle V,E\rangle$是个简单有向图,$V=\{v_1,v_2,v_3,\ldots,v_m\}$,$E=\{e_1,e_2,e_3,\ldots,e_n\}$,一个$m\times n$阶矩阵$M=(m_{ij})$称为G的完全关联矩阵. 其中:

$$m_{ij}=\begin{cases} 1 & v_i\text{是}e_j\text{的起点} \\ -1 & v_i\text{是}e_j\text{的终点} \\ 0 & v_i\text{与}e_j\text{不关联} \end{cases}$$



$$M=\begin{array}{c} \\ v_1\\ v_2\\ v_3\\ v_4\\ v_5 \end{array}\begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \end{array}$$

2).从关联矩阵看图的性质:

a)每列只有一个1和一个-1.

(每条边有一个起点一个终点)

b)每行中1的个数为对应结点的出度 -1个数是结点入度

本节重点掌握:

图的三个矩阵的求法

由图的矩阵,看图的性质.

作业 P300 (3)

*8-4. 带权图的最短路与关键路

在实际应用中,一些图的边上标有数字, 用以表示两结点间的距离、或路费等等. 然后求两点间的最短路径.这是很有意义的问题.

一.带权图(赋权图) Weighted Graph

1.定义:设G=<V,E,W>,是个图,如果G的每条边e上都标有实数c(e) (c(e)∈W),称这个数为边e的权, 称此图为带权图.

规定: u,v∈V, 边(u,v)的权记作 c(u,v)

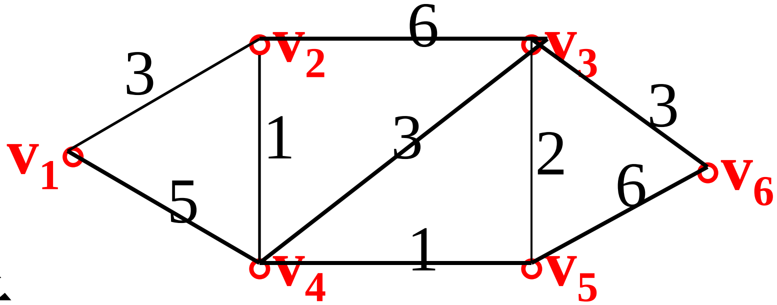
1) $c(u,u)=0$

2) 如果结点u与v之间无边相连,则 $c(u,v)=\infty$

2.带权图的路长:结点u与v之间的路长是指该路所包含的各边权的总和.

例如右图中

$v_1\,v_2v_3\,v_6$ 的路长为12.



3.带权图的两点间距离:

结点**u**与**v**之间的 最短路的长

称为结点**u**与**v**之间的距离. 记作**d(u,v).**

如果**G**是有向带权图,称为结点**u**到**v**的距离,记作**d<u, v>**

例如上图中 **$d(v_2,v_5)=2$**

4.带权图中求一个结点到各点的最短路的算法:

此算法是于**1959**年由**E.W.Dijkstra**提出的.

基本思想:若使 **$(u_0,u_1,u_2,\ldots,u_{n-1},u_n)$**最短, 就要使 **$(u_0,u_1,u_2,\ldots,u_{n-1})$**最短, 即保证从**$u_0$**到以后各点的路都是最短的.

令图$G=\langle V,E,W\rangle$, 集合$S_i \subseteq V$ $S_i'=V-S_i$，令$|V|=n$

$S_i=\{u|$从u_0到u的最短路已求出$\}$

$S_i'=\{u'|$从u_0到u'的最短路未求出$\}$

Dijkstra算法:(求从u_0到各点u的最短路长)

第一步. 置初值: $d(u_0,u_0)=0$ $d(u_0,v)=\infty$ (其中$v\neq u_0$)

$i=0$ $S_0=\{u_0\}$ $S_0'=V-S_0$,

第二步.若 $i=n-1$ 则停. 否则转第三步

第三步. 对每个$u'\in S_i'$

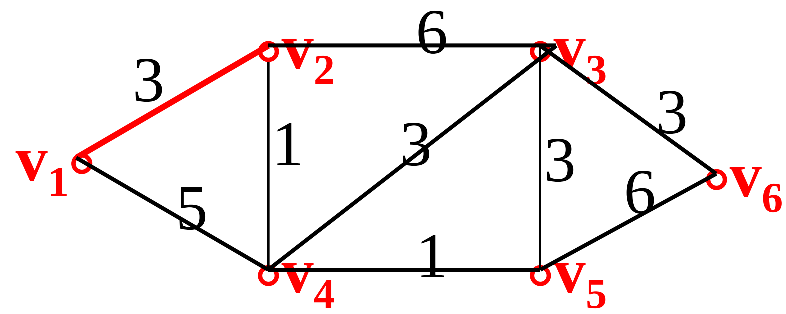
计算 $d(u_0,u')=\min\limits_{u_i\in S_i}\{d(u_0,u'), d(u_0,u_i)+c(u_i,u')\}$

计算 $\min\limits_{u'\in S_i'}\{d(u_0,u')\}$

并用u_{i+1}记下达到该最小值的那个结点u'

置$S_{i+1}=S_i\cup\{u_{i+1}\}$ $i=i+1$ $S_i'=V-S_i$，转第二步.

例.求右图中从v_1到v_6的最短路



1.置初值: $u_0=v_1$

$d(u_0,u_0)=0$

$d(u_0,v_2)=d(u_0,v_3)=d(u_0,v_4)=d(u_0,v_5)=d(u_0,v_6)=\infty$

2.3. $i=0$ $S_0=\{v_1\}$ $S_0'=\{v_2,v_3,v_4,v_5,v_6\}$

$d(u_0,v_2)=\min\{d(u_0,v_2), d(u_0,u_0)+c(u_0,v_2)\}=\min\{\infty,0+3\}=3$

$d(u_0,v_3)=\min\{d(u_0,v_3),d(u_0,u_0)+c(u_0,v_3)\}=\min\{\infty,0+\infty\}=\infty$

$d(u_0,v_4)=\min\{d(u_0,v_4), d(u_0,u_0)+c(u_0,v_4)\}=\min\{\infty,0+5\}=5$

$d(u_0,v_5)=\min\{d(u_0,v_5),d(u_0,u_0)+c(u_0,v_5)\}=\min\{\infty,0+\infty\}=\infty$

$d(u_0,v_6)=\min\{d(u_0,v_6),d(u_0,u_0)+c(u_0,v_6)\}=\min\{\infty,0+\infty\}=\infty$

$\min\{3,\infty,5,\infty,\infty\}=3$

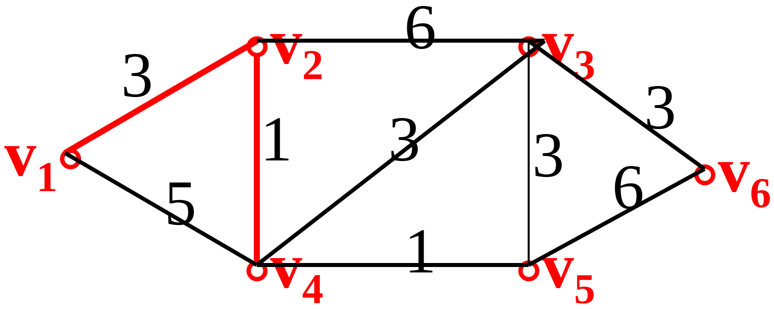
$u_{i+1}=u_1=v_2$, 实际已求出$d(u_0,v_2)=3$, 路是u_0v_2

i=1 $S_1=\{v_1, v_2\}$

$S_1'=\{v_3,v_4,v_5,v_6\}$

$u_1=v_2$

$d(u_0,u_1)=3$



$d(u_0,v_3)=\min\{d(u_0,v_3),d(u_0,u_1)+c(u_1,v_3)\}=\min\{\infty,3+6\}=9$

$d(u_0,v_4)=\min\{d(u_0,v_4), d(u_0,u_1)+c(u_1,v_4)\}=\min\{5,3+1\}=4$

$d(u_0,v_5)=\min\{d(u_0,v_5),d(u_0,u_1)+c(u_1,v_5)\}=\min\{\infty,3+\infty\}=\infty$

$d(u_0,v_6)=\min\{d(u_0,v_6),d(u_0,u_1)+c(u_1,v_6)\}=\min\{\infty,3+\infty\}=\infty$

$\min\{9,4,\infty,\infty\}=4$

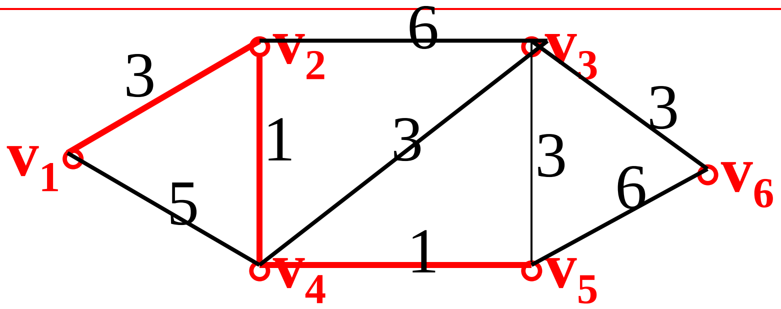
$u_{i+1}=u_2=v_4$， 实际已求出$d(u_0,v_4)=4$, 路是$u_0v_2v_4$

i=2 $S_2=\{v_1, v_2, v_4\}$

$S_2'=\{v_3,v_5,v_6\}$

$u_2=v_4$

$d(u_0,u_2)=4$



$d(u_0,v_3)=\min\{d(u_0,v_3), d(u_0,u_2)+c(u_2,v_3)\}=\min\{9 ,4+3\}=7$

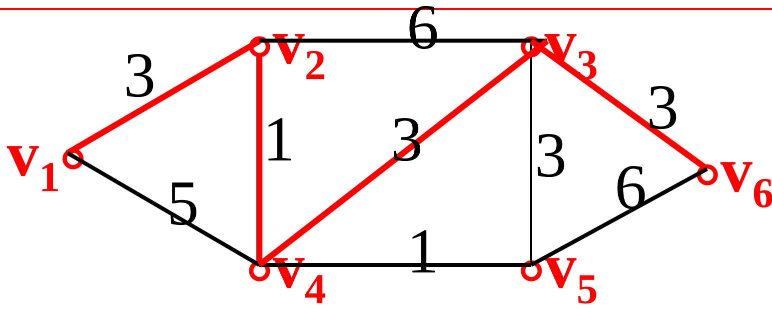
$d(u_0,v_5)=\min\{d(u_0,v_5), d(u_0,u_2)+c(u_2,v_5)\}=\min\{\infty,4+1\}=5$

$d(u_0,v_6)=\min\{d(u_0,v_6), d(u_0,u_2)+c(u_2,v_6)\}=\min\{\infty,4+\infty\}=\infty$

$\min\{7,5,\infty\}=5$

$u_{i+1}=u_3=v_5$, 实际已求出$d(u_0,v_5)=5$, 路是$u_0v_2v_4v_5$

$i=3$ $S_3=\{v_1, v_2, v_4, v_5\}$



$S_3'=\{v_3, v_6\}$

$u_3=v_5$

$d(u_0,u_3)=5$

$d(u_0,v_3)=\min\{d(u_0,v_3),d(u_0,u_3)+c(u_3,v_3)\}=\min\{7,5+3\}=7$

$d(u_0,v_6)=\min\{d(u_0,v_6),d(u_0,u_3)+c(u_3,v_6)\}=\min\{\infty,5+6\}=11$

$\min\{7,11\}=7$

$u_{i+1}=u_4=v_3$, 实际已求出$d(u_0,v_3)=7$, 路是$u_0v_2v_4v_3$

$i=4$ $S_3=\{v_1, v_2, v_4, v_5, v_3\}$ $S_3'=\{v_6\}$ $u_4=v_3$ $d(u_0,u_4)=7$

$d(u_0,v_6)=\min\{d(u_0,v_6),d(u_0,u_4)+c(u_4,v_6)\}=\min\{11,7+3\}=10$

$\min\{10\}=10$

$u_{i+1}=u_5=v_6$, 实际已求出$d(u_0,v_6)=10$, 路是$u_0v_2v_4v_3v_6$

$i=5$ $(n-1)$ 时 算法停止.

二.求关键路径问题

实施一项工程计划时, 若将整个工程分成若干个工序, 有些工序可以同时实施, 有些工序必须在另一些工序完成之后才能实施, 工序之间的次序关系用有带权的有向图表示, 这种有向图称为**PERT图(计划评审技术图).**

(Program Evaluation and Review Technique)

1. PERT图定义:

$D=\langle V,E,W\rangle$是有向带权图, $|V|=n$, 如果满足:

(1). D是简单图.

(2). D中无回路.

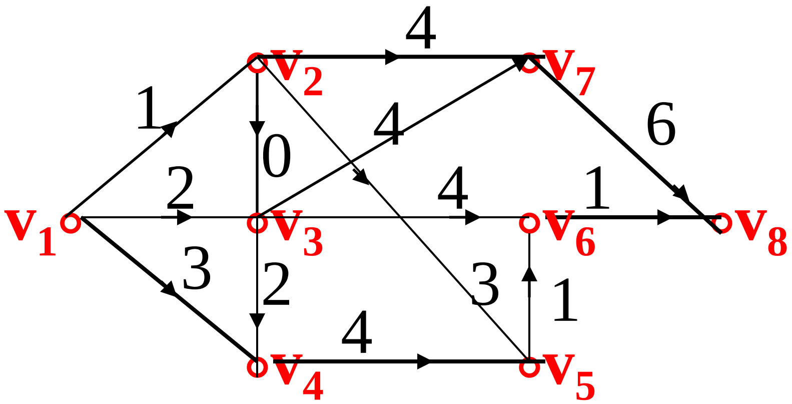
(3).有一个结点入度为0, 称此结点为发点; 有一个结点出度为0, 称此 结点为收点.

(4).边$\langle v_i,v_j\rangle$带的权记作w_{ij}, (通常表示时间.)

称此图D为PERT图.

如右图就是个PERT图.

在PERT图中，要找出关键路径, 即是影响工程工期的关键路径.



就是通过求从发点到收点的一条最长路径, 通过求各个结点的最早完成时间和最迟完成时间, 来求关键路径. 为此先给出两个概念:

2.结点v的先驱元集: 令D=<V,E>为有向图, v∈V,称集合

$$\Gamma_D^-(v)=\{x \mid x\in V \wedge <x,v>\in E\}$$

为v的先驱元集.

3.结点v的后继元集: 令D=<V,E>为有向图, v∈V,称集合

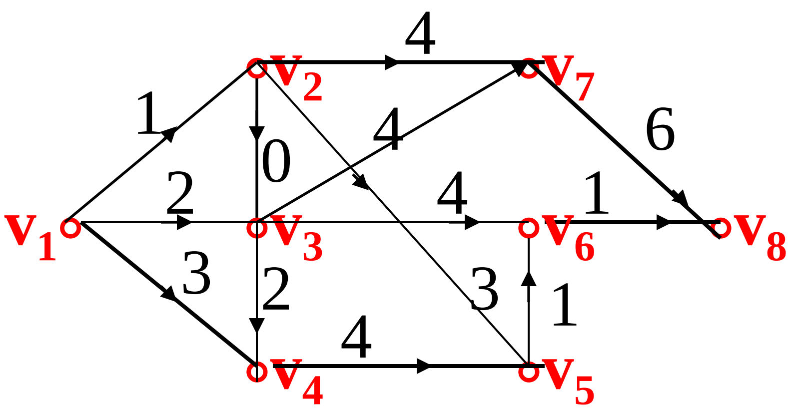
$$\Gamma_D^+(v)=\{x \mid x\in V \wedge <v,x>\in E\}$$

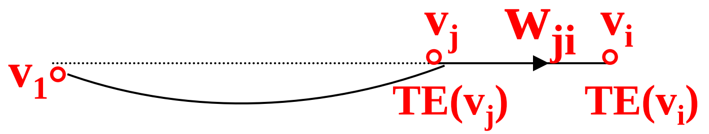
为v的后继元集.

4. 最早完成时间:

自发点(记作v_1)开始沿**最长**路径(按权计算)到达v_i，称为v_i的最早完成时间，记作$TE(v_i)$，$i=1,2,\ldots,n$ 。

显然$TE(v_1)=0$,





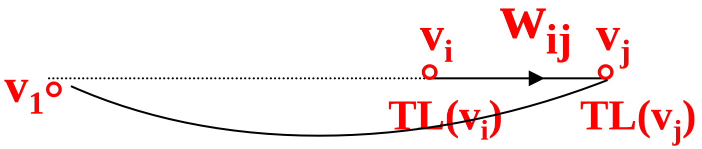
$$TE(v_i) = \max_{v_j \in \Gamma_D^-(v_i)} \{TE(v_j) + w_{ji}\}, \quad i = 2,3,\ldots,n \quad (1)$$

收点v_n最早完成时间$TE(v_n)$就是从v_1到v_n的最长路径的长.

5. 最晚完成时间:

在保证收点v_n的最早完成时间不增加的条件下,自发点(记作v_1)最迟到达v_i的时间,称为v_i的最晚完成时间, 记作$TL(v_i)$, $i=1,2,\ldots,n$

显然$TL(v_n)=TE(v_n)$,



$$TL(v_i) = \min_{v_j \in \Gamma_D^+(v_i)} \{TL(v_j) - w_{ij}\}, \quad i = 2,3,\ldots,n-1 \quad (2)$$

6.缓冲时间:

称 $TS(v_i)=TL(v_i))-TE(v_i)\quad i=1,2,\ldots n$

为v_i的缓冲时间.

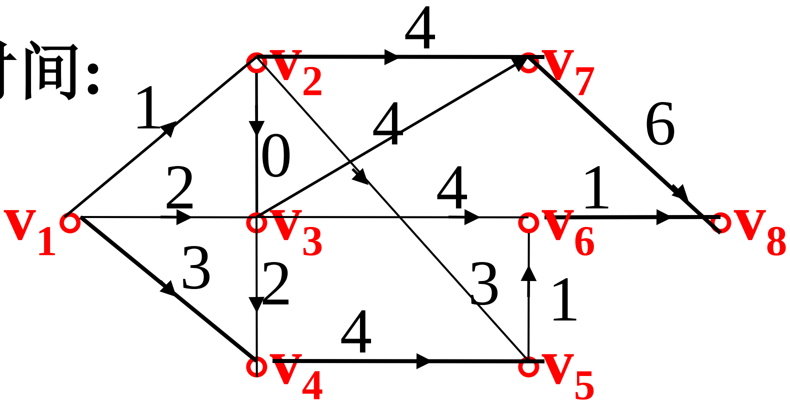
7.关键路径:就是各个结点的缓冲时间均为0的路径.

可见在关键路径上,如果一个工序增加了时间t, 则整个工程就推迟了时间t. 所以才称之为关键路径.

例如, 求右图的关键路径

(1)求各个结点的最早完成时间:

计算时，从前向后逐个结点计算。



$TE(v_1)=0$

$TE(v_2)=\max\{0+1\}=1$　　　　$(v_1 — v_2$的先驱结点$)$

$TE(v_3)=\max\{0+2 ,1+0\}=2$　$(v_1v_2 — v_3$的先驱结点$)$

$TE(v_4)=\max\{0+3,2+2\}=4$　$(v_1v_3 — v_4$的先驱结点$)$

$TE(v_5)=\max\{1+3,4+4\}=8$　$(v_2v_4 — v_5$的先驱结点$)$

$TE(v_6)=\max\{2+4,8+1\}=9$　$(v_3v_5 — v_6$的先驱结点$)$

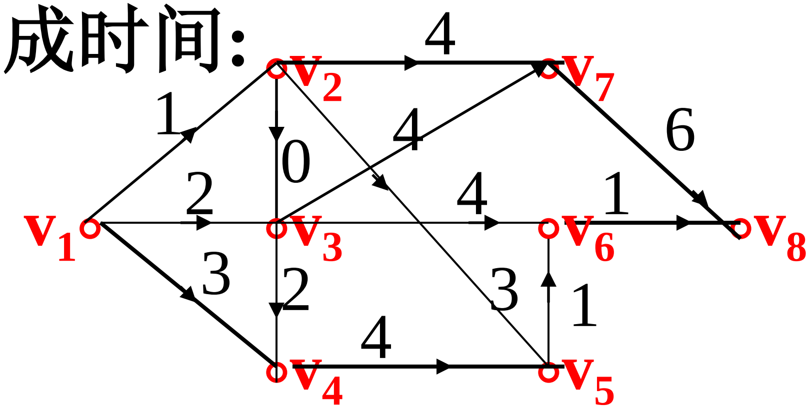
$TE(v_7)=\max\{1+4,2+4\}=6$　$(v_2v_3 — v_7$的先驱结点$)$

$TE(v_8)=\max\{9+1,6+6\}=12$　$(v_6v_7 — v_8$的先驱结点$)$

(2)求各个结点的最晚完成时间:

计算时，从后向前

逐个结点计算。



(括号内是前面结点的后继结点)

$TL(v_8)=TE(v_8)=12$

$TL(v_7)=\min\{12-6\}=6$ (v_8)

$TL(v_6)=\min\{12-1\}=11$ (v_8)

$TL(v_5)=\min\{11-1\}=10$ (v_6)

$TL(v_4)=\min\{10-4\}=6$ (v_5)

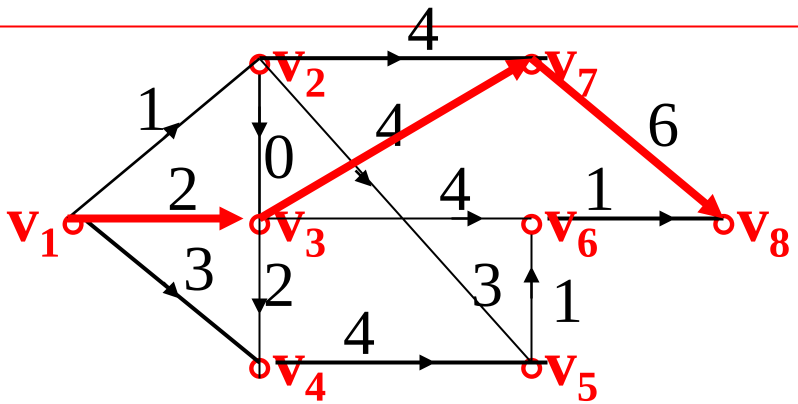
$TL(v_3)=\min\{6-2, 11-4, 6-4\}=2$ $(v_4v_6v_7)$

$TL(v_2)=\min\{2-0, 10-3, 6-4\}=2$ $(v_3v_5v_7)$

$TL(v_1)=\min\{2-1, 2-2, 6-3\}=0$ $(v_2v_3v_4)$

(3)求各个结点的缓冲时间

$TS(v_i)=TL(v_i)-TE(v_i)$



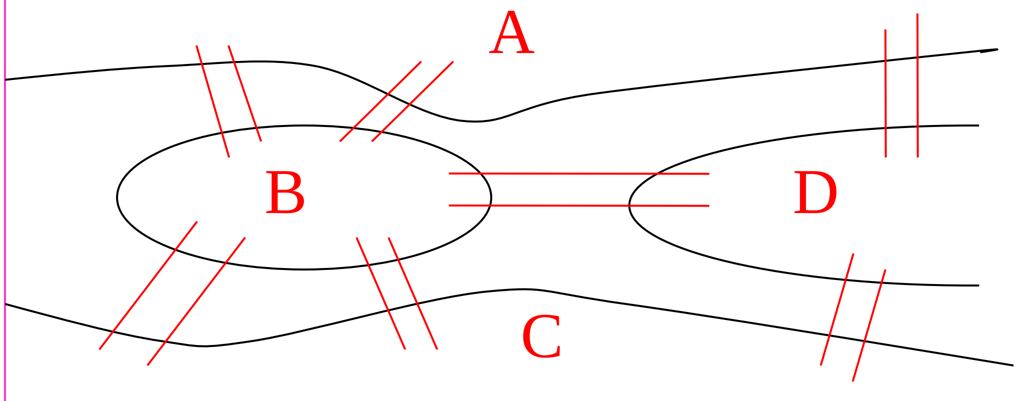
v_i	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
$TL(v_i)$	0	2	2	6	10	11	6	12
$TE(v_i)$	0	1	2	4	8	9	6	12
$TS(v_i)$	0	1	0	2	2	2	0	0

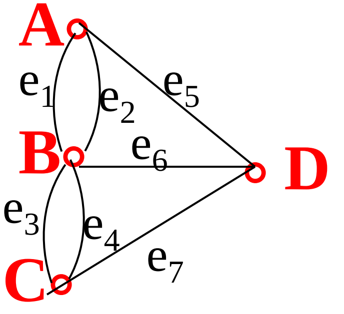
关键路径为: $v_1v_3v_7v_8$

8-5. 欧拉图与汉密尔顿图

这里主要讨论图的遍历问题,一个是遍历过程中要求经过的所有边都不同;一个是遍历过程中要求经过的所有结点都不同.

欧拉在1736年发表了第一篇关于图论的论文, 就是就七桥问题.





一.欧拉图:

1.欧拉路:在无孤立结点的图G中,如果存在一条路,它经过图中每条边一次且仅一次, 称此路为**欧拉路**.

2.欧拉回路:在无孤立结点的图G中,若存在一条回路,它经过图中每条边一次且仅一次,称此回路为**欧拉回路**.

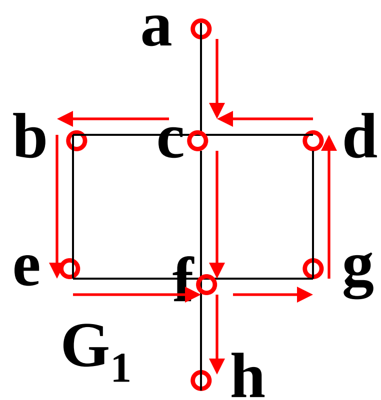
称此图为**欧拉图**,或**E图**.(Euler)

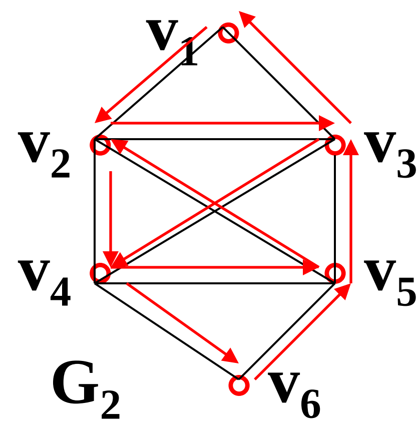
在G_1中:有欧拉路:

acbefgdcfh

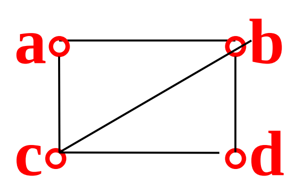
在G_2中:有欧拉回路:

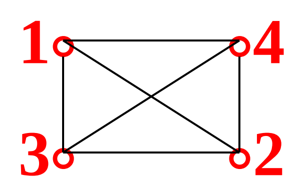
$v_1v_2v_3v_4v_5v_2v_4v_6v_5v_3v_1$





下面两个图中是否有欧拉路,或有欧拉回路?如何判断?





3.有欧拉路与有欧拉回路的判定:

定理8-5.1: 无向图G具有欧拉路,当且仅当G是连通的,且有零个或两个奇数度的结点.

证明:必要性: 设G有欧拉路.设此路为$v_0\ e_1 v_1\ e_2 v_2 \ldots v_i \ldots\ e_n v_k$.其中的结点可能重复，但边不重复，因为G无孤立结点，此路又包含了G的所有边，所以此路必包含了G的所有结点, 所以图G是连通的。

因为对此路中任何非端点的结点v_i，每当它出现一次,必关联两条边，故它虽可重复出现，但始终$deg(v_i)$ 是偶数。

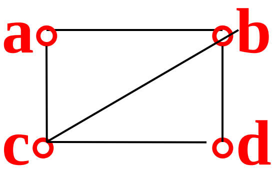
对于端点v_0和v_k:

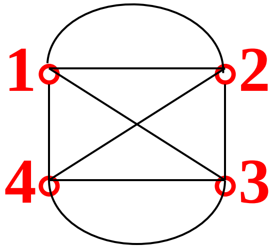
如果$v_0 = v_k$,则 $deg(v_0)$是偶数。则G中无奇数度结点。

如果$v_0 \neq v_k$, 则$deg(v_0)$为奇数， $deg(v_k)$也为奇数。则G中有两个奇数度结点。

充分性：(证明的过程就是一个构造欧拉路的过程)

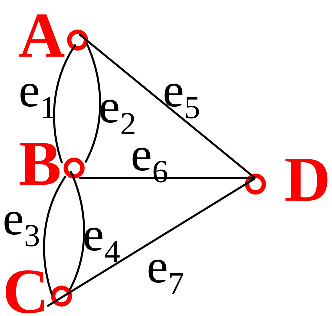
如果G有两个奇数度结点:就从一个奇数度结点出发,每当到达一个偶数度结点,必然可以再经过另一条边离开此结点,如此重复下去, 经过所有边后到达另一个奇数度结点。如果G无奇数度结点,则可以从任何一个结点出发,去构造一条欧拉路.



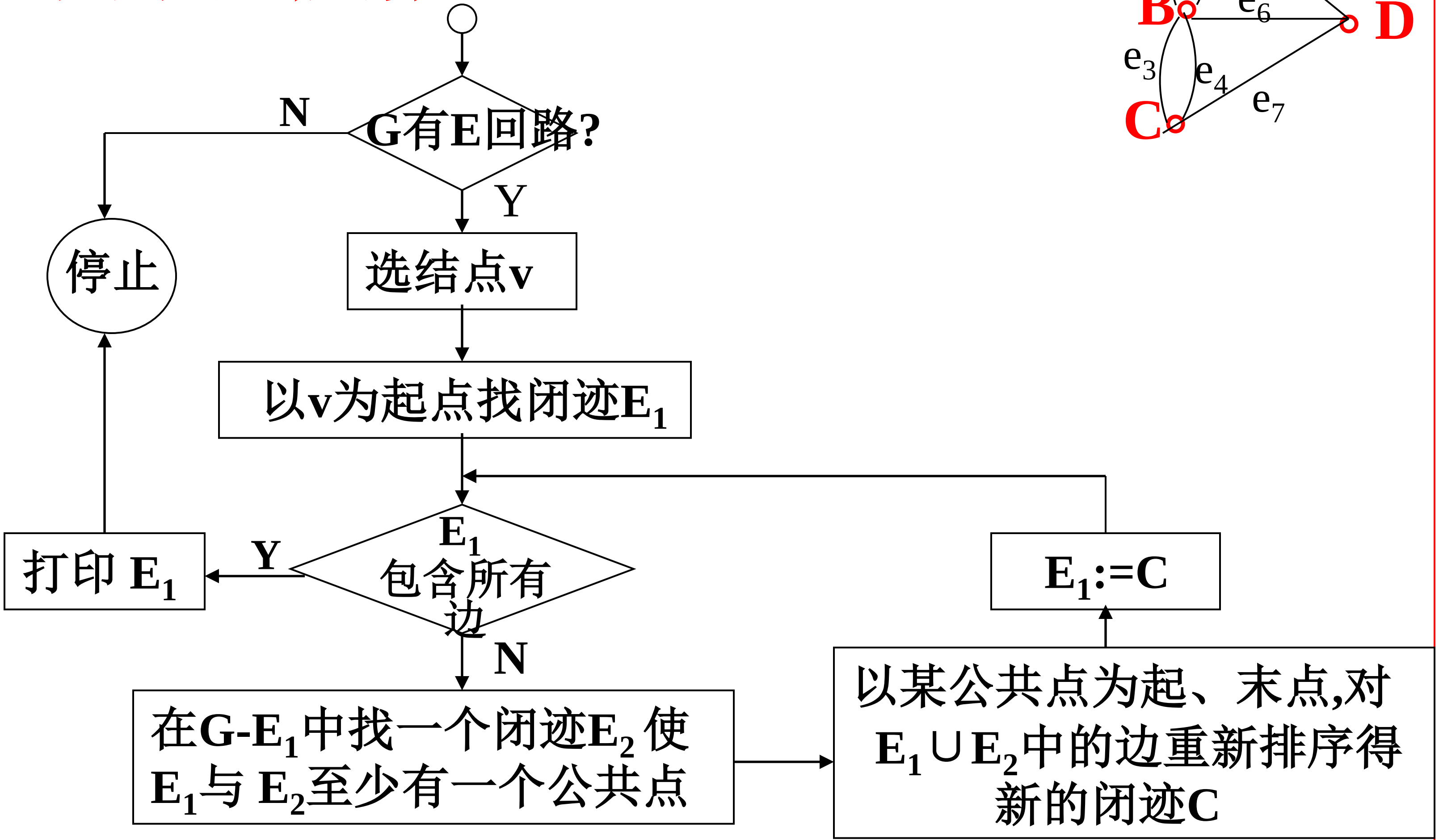


推论:无向图G具有欧拉回路,
当且仅当G是连通的,且所有结点的度都是偶数.

从此推论得知,七桥问题的图不是欧拉图.

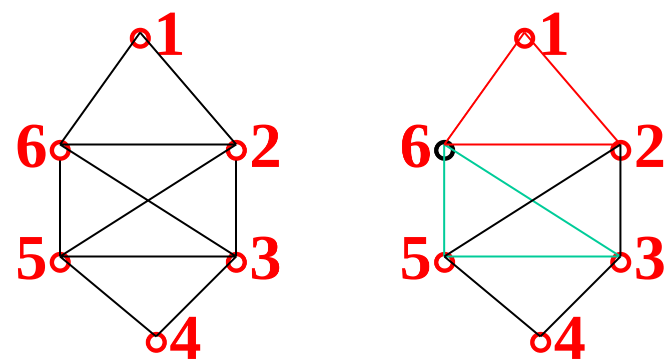


4.求欧拉回路的算法:



用上述算法求右图中欧拉回路.

此图中所有结点度均为偶数,所以有欧拉回路.



a) 选以1为起点的闭迹E_1:1261

b) E_1不包含所有边.

c) 在G- E_1中找新闭迹E_2: 6356 (6是E_1与E_2的公共点)

d)以公共点6为起点,对$E_1 \cup E_2$中的边排序:C=6356126

e) E_1 := C

f) E_1不包含所有边.

g) 在G- E_1中找新闭迹E_2: 52345 (5是E_1与E_2的公共点)

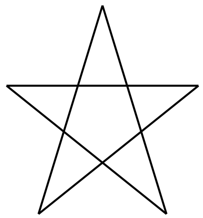
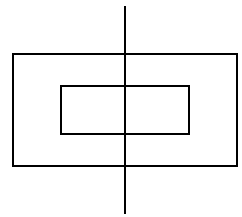
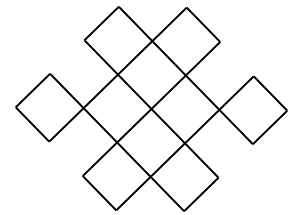
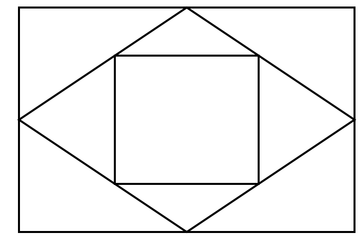
h)以公共点5为起点,对$E_1 \cup E_2$中的边排序:

C=52345612635

i) E_1 := C

j) E_1包含所有边. k)打印E_1 =52345612635 l)停止.

欧拉路与欧拉回路问题, 也称一笔画问题.

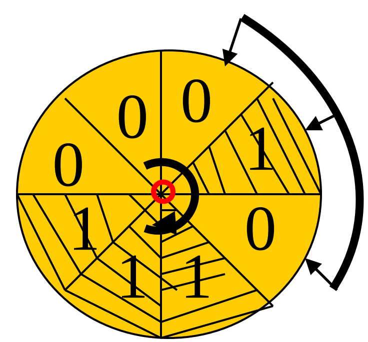





***5.欧拉图的应用**----计算机鼓轮的设计

早期向计算机输入数据, 为简单,以输入八进制数为例

(0,1,2,3,4,5,6,7,即000,001,010,011,100,101,110,111)

鼓轮表面分成2^3等分,每一等分分别用绝缘体或导体组成,绝缘部分输出0,导体部分输出1. 有三个触点分别与三个部分接触,以读取三个数字. 如图所示:

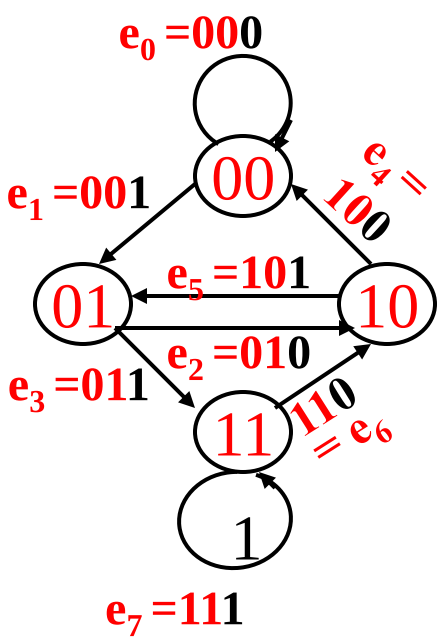


转动鼓轮,分别输出8个数:

000,001,010,011,100,101,110,111

下面介绍此鼓轮的设计过程:

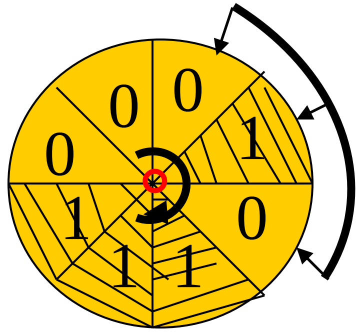
此轮的设计:以两位二进制数 $V=\{00,01,10,11\}$为结点,画带权图(即边上标有数字--称为边的权), 从任何$a_1a_2 \in V$结点画有向边,标的权0(或1), 该边指向结点a_20(或a_21),于是构成边a_1a_20, (或a_1a_21),这八条边分别表示八个二进制数:



000,001,010,011,100,101,110,111

从此图上取一个回路: $e_0e_1e_2e_5\ e_3e_7e_6e_4$

将上述各边的末位数字写成序列:01011100,于是就按照此序列将鼓轮进行加工,标0部分用绝缘体,标1部分用导体.

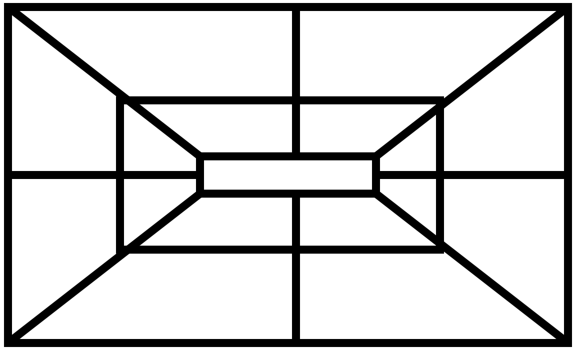


二. 汉密尔顿图(H图) (Hamilton图)

Hamilton是英国数学家,在1959年,他提出Hamilton回路.

H图起源于一种游戏,这个游戏就是所谓周游世界问题.

例如,某个城市的街道如图所示:



该城市的所有交叉路口都有形象各异的精美的雕塑,吸引着许多游客,人人都想找到这样的路径:游遍各个景点再回到出发点----H回路.

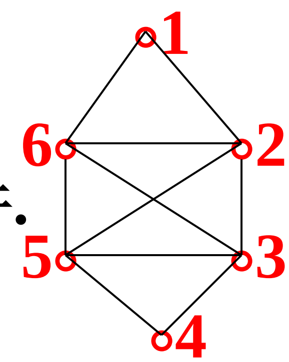
1.定义:设G=<V,E>是个无向有限图,

汉密尔顿路:通过G中每个结点恰好一次的路.

汉密尔顿回路(H回路):通过G中每个结点恰好一次的回路.

汉密尔顿图(H图):具有汉密尔顿回路(H回路)的图.

例如右图中,就是H图,因为它有H回路:1234561

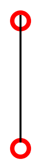


2.汉密尔顿图的判定:

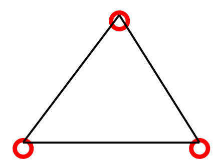
到目前为止并没有判定H图的充分和必要条件.

定理8-5.2 (充分条件):G是完全图,则G是H图.

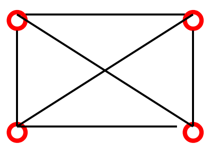
证明:略



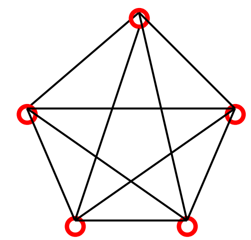
K_2



K_3



K_4



K_5

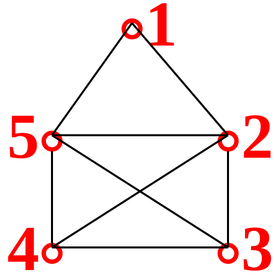
定理8-5.3(充分条件)设G是有n个结点的简单图,若G中每对结点度数之和大于等于n-1(n),则G有一条H路(H回路)

证明: 先证明G是连通的.(反证法) 见书P307

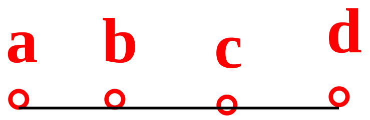
再构造H路(H回路)

在图G_1中, 满足充分条件$\Delta(G)=4$ $\delta(G)=2$

任意两个结点度数之和大于等于5,所以是H图.



G_1



G_2

注意:上述条件只是充分条件,而不是必要条件, 即不满足这个条件的, 也可能有H路.

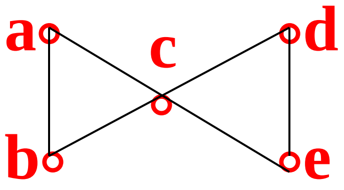
例如:在图G_2中, 并不满足任意两个结点度数之和大于3, 但是却有H路.

定理8-5.4:(必要条件) 若图$G=\langle V,E\rangle$有H回路,则对V的任何非空子有限集S, 均有$W(G-S)\le|S|$, 其中$W(G-S)$是从G中删去S中所有结点及与这些结点关联的边所得到的子图的连通分支数.

证明: 设C是图G的一条H回路,则对于V的任何非空子集S, 在C中删去S中任意一个结点v_1后, 则$C-\{v_1\}$仍是连通的路, 若再删去S中的另一个结点v_2, 则$W(C-\{v_1,v_2\})\le 2$, 若$|S|=n$则删去S中的n个结点，有$W(C-\{v_1,v_2,\ldots,v_n\})\le n$, 所以$W(C-\{v_1,v_2,\ldots,v_n\})\le|S|$. 因为C是H回路, 所以它包含了G的所有结点, 即C是G的生成子图. 所以C-S 也是G-S 的生成子图, 故 $W(G-S)\le W(C-S)\le|S|$.

用此定理可以判断一个图不是H图.

如右图G, 取 $S=\{c\}$ 不满足 $W(G-S)\le|S|$.



3.用“最邻近法”求H回路

如果已经确定图G有H回路.

a). 任选初始结点u,找一个最邻近的(边的权最小)结点x.

b). 设x是新加入到这条路的结点, 再从不在此路上的结点中找到一个与 x邻近的(边的权最小)结点,加到此路中.

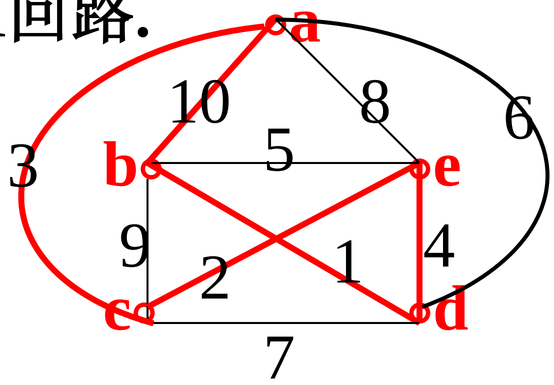
c).重复b), 直到G中所有结点都在此路上.

d).最后再回到起点, 构成回路. 就是H回路.

例如右图

初始结点为a,

逐渐选邻近结点c,e,d,b,a.



得H回路acedba. 此回路的总权为:20

但是对带权图来说,此方法求的H回路不一定是最短的.

例如,实际上此图最短的H回路是 acebda 总权为17

本节重点掌握:

　　欧拉路及欧拉回路的判定, 能求欧拉路和欧拉回路.

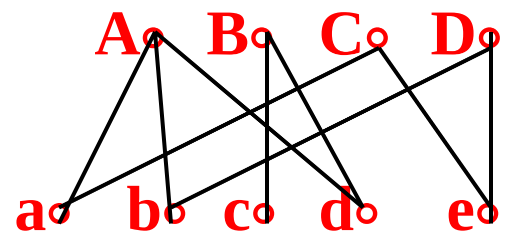
　　H路及H回路的判定, 能求H路和H回路.

以及欧拉图和汉密尔顿图的应用.

作业 P311 (1) (2) (6)

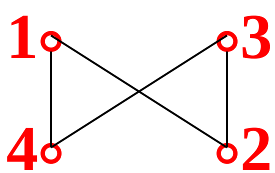
*8-6 二部图 Bipartite Graph

在实际应用中,有些如对策、二人对弈、分配工作等问题。例如有A,B,C,D四个人,有a,b,c,d,e五种工作,如果某人可以做某种工作,则它们之间连一直线.
请给他们安排工作.

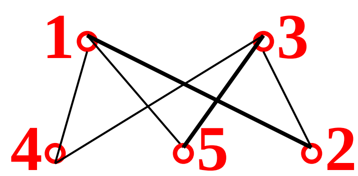


1.定义:令$G=<V,E>$是无向图,
如果可以将V划分成两个子集V_1,V_2,
使得任何边$(v_i,v_j)\in E$, $v_i\in V_1$, $v_j\in V_2$, 则称G是二部图,也称二分图. 并称V_1,V_2是G的互补的结点子集。

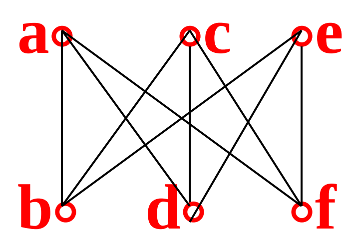
2.完全二部图:令$G=\langle V,E\rangle$是以V_1,V_2为互补的结点子集的二部图,如果V_1中的每个结点都与V_2中每个结点相邻接,则称G是完全二部图. 如果$|V_1|=m$, $|V_2|=n$ 则G记作$K_{m,n}$



$K_{2,2}$

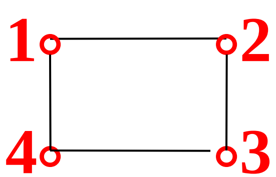


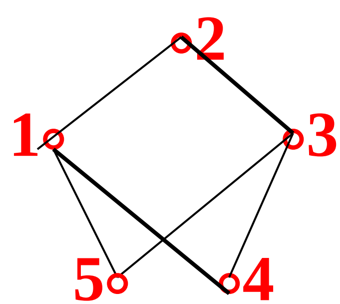
$K_{2,3}$

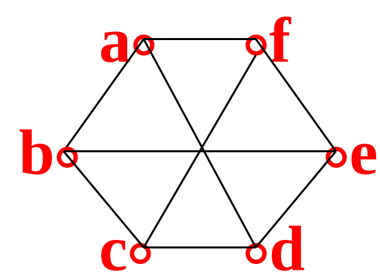


$K_{3,3}$

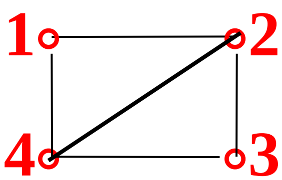
下面的图是否可以画成二部图?

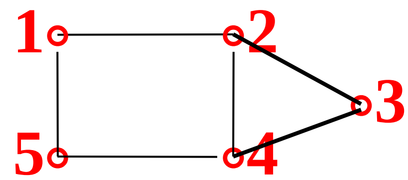






下面两个图, 是否可以画成二部图?





V_1={1,3} V_2={2,4} ,

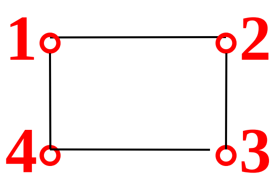
边(2,4)如何画?

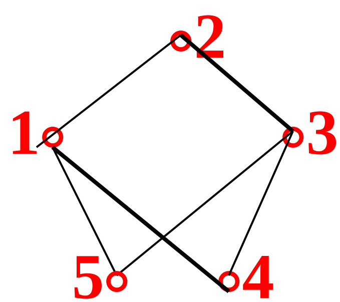
V_1={1,4} V_2={2,5}

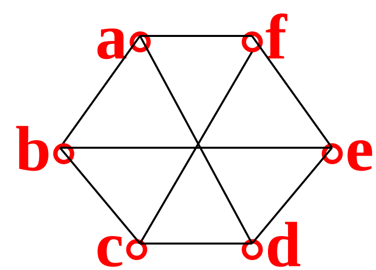
结点3放哪里? 边(2,3)(3,4)?

•什么样的图可以画成二部图?

再看看前面三个图:







3.二部图的判定:

定理8-6.1 $G=\langle V,E\rangle$是二部图 当且仅当它的所有回路的长度都是偶数.

证明:必要性: 假设G是个以V_1,V_2为互补的结点子集的二部图, 设任意长度为n的回路: $v_{i0}v_{i1}v_{i2},\ldots,v_{in-1}v_{i0}$,

假设 $v_{i0}v_{i2}v_{i4},\ldots,v_{in-2} \in V_1$, $v_{i1}v_{i3}v_{i5},\ldots,v_{in-1} \in V_2$,

可见n-1是奇数, 所以n是偶数.

充分性: 设G的每个回路长度均为偶数,

a)设G是连通的. 定义结点集合:

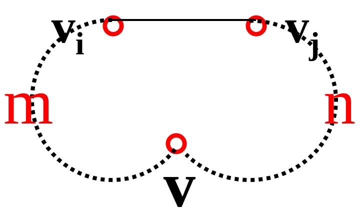
$V_1=\{v_i \mid v_i$和某个固定结点v之间的距离为偶数$\}$

$V_2=V-V_1=\{v_i \mid v_i$和某个固定结点v之间的距离为奇数$\}$

下面证明E中任何边, 其端点分别属于V_1和V_2.

(用反证法)假设有一条边$(v_i, v_j) \in E$, 使得$v_i \in V_1$, $v_j \in V_1$

由V_1的定义可知:结点 v到v_i有最短路长为m(是偶数),v到v_j有最短路长为n(是偶数), 所以再加上边(v_i,v_j), 就得到一条长度为(m+n+1)的回路, 可是此回路长为奇数, 与已知条件矛盾. 所以v_i和v_j不能属于同一个结点子集.



类似地,假设有一条边$(v_i, v_j)\in E$,

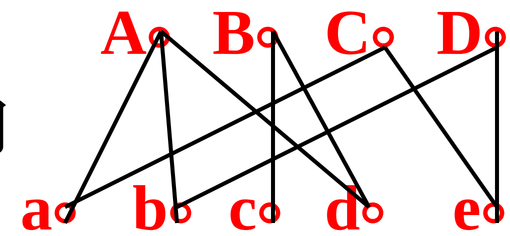
使得$v_i\in V_2$, $v_j\in V_2$,

由V_2的定义可知:结点 v到v_i有最短路长为m(是奇数),v到v_j有最短路长为n(是奇数), 所以再加上边(v_i,v_j), 就得到一条长度为(m+n+1)的回路, 可是此回路长为奇数, 与已知条件矛盾. 所以v_i和v_j不能属于同一个结点子集.

G是个以V_1,V_2为互补的结点子集的二部图,

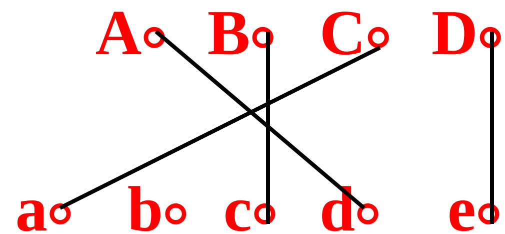
b)如果G不是连通图时, 可以对G的每个分图,重复上述证明,得到同样结论.　　最后得, G必是二部图.

例如有A,B,C,D四个人,有a,b,c,d,e五种工作,如果某人可以做某种工作,则它们之间连一直线.请给他们安排工作.



就是匹配问题.

4. 匹配



假设G=<V,E>是个以V_1,V_2为互补的结点子集的二部图,

令$V_1=\{v_1,v_2,v_3,\ldots,v_k\}$, V_1对V_2的一个匹配 是G的一个子图,它由k条边(v_1,v_1'), (v_2,v_2'), …,(v_k,v_k')组成, 其中$v_1',v_2',\ldots,v_k'$是V_2中k个不同元素.

如果 $|V_1|=|V_2|$时, 此匹配称为完美匹配.

(匹配相当于一个入射)　(完美匹配相当于一个双射)

5.求一个匹配的算法：设$G_0=<V_0, E_0>$是二部图，

(1)置初值: $V_1=V_0$　$E_1=\Phi$　$G_1=G_0$

(2)如果G_1是零图, 则结束,得E_1。否则在V_1中选取度最小的结点, 不妨设这个结点是u, 且与u相邻接的一个结点为v, 取边(u,v) ,

$$E_1=E_1\cup\{(u,v)\}$$

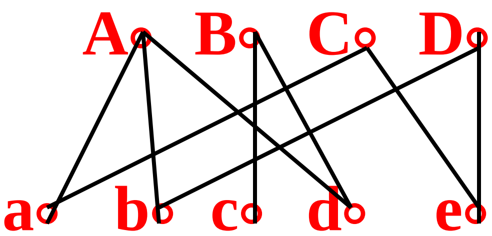
(3)从图G_1中删去结点u,v。即 $V_1=V_1-\{u,v\}$), 得到图G_1

(4)转到(2)。

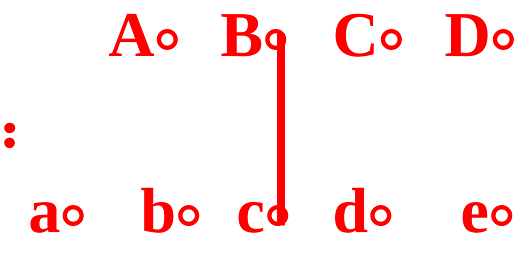
对于上例,执行此算法:

a) 度数最小的结点是 c , $E_1=\{(B,c)\}$

G_1

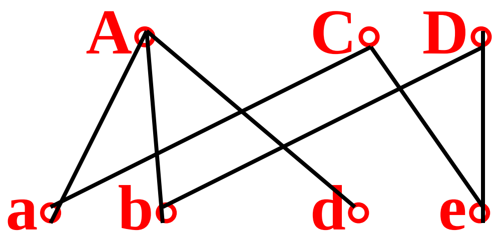


匹配:

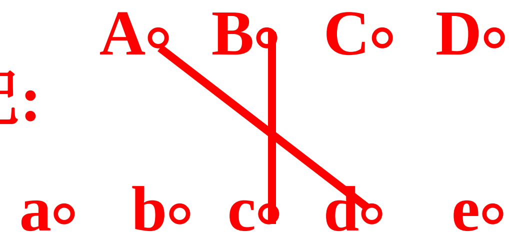


b) G_1中度数最小的结点是 d， $E_1=\{(B,c),(A,d)\}$

G_1

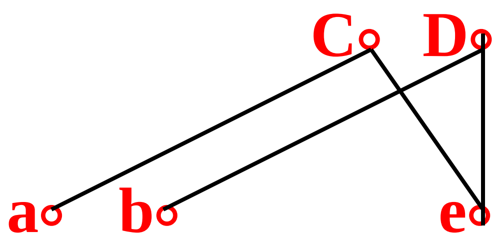


匹配:

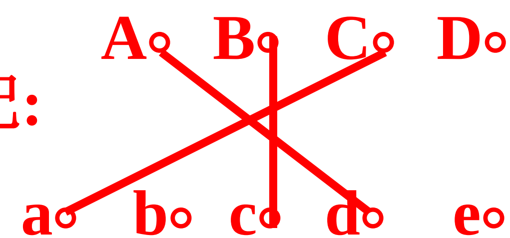


c) G_1中度数最小的结点是 a， $E_1=\{(B,c),(A,d)(C,a)\}$

G_1

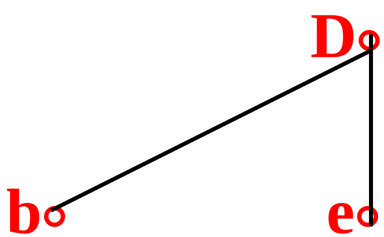


匹配:

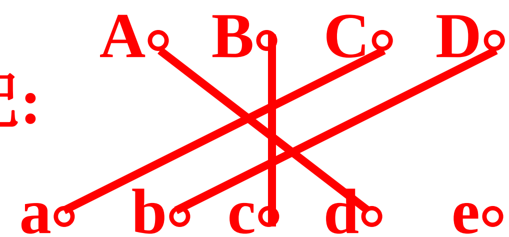


d) G_1中度数最小的结点是 b， $E_1=\{(B,c),(A,d),(C,a),(D,b)\}$

G_1



匹配:

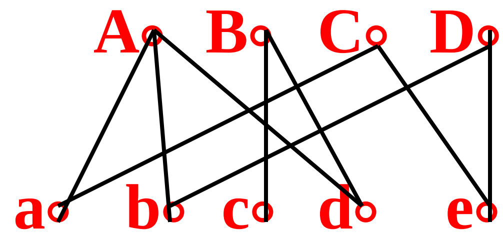


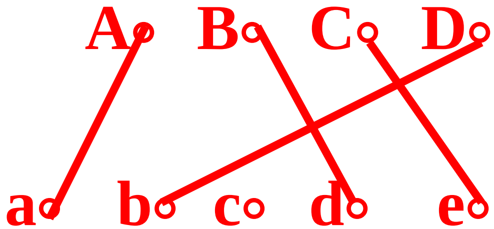
e) G_1是零图，匹配终止。最后得E_1是一个匹配。

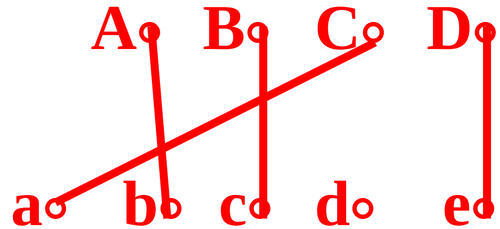
G_1

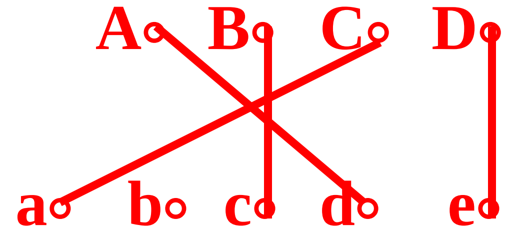
$E_1=\{(B,c),(A,d),(C,a),(D,b)\}$

当然执行此算法,不一定得到所有匹配。
可见下面也都是匹配。









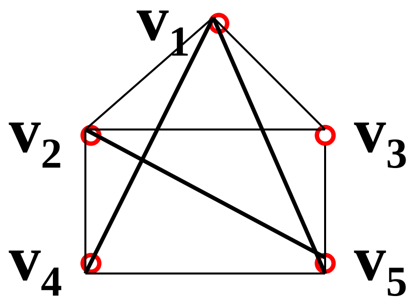
8-7 平面图 Plane Graph

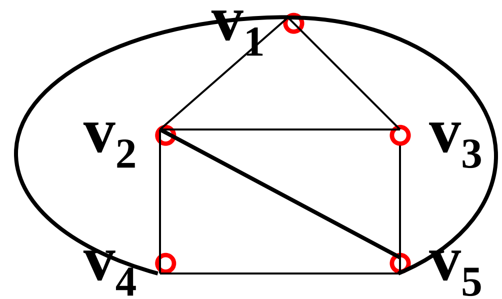
在实际应用中,如高速公路设计、印刷电路设计,都要求线路不交叉,这就是平面图, 一个图能否画在一个平面上, 且任何边都不交叉, 这就是图的平面化问题. 这个问题在近些年来, 特别是大规模集成电路的发展进一步促进了对平面图的研究.

1. 定义

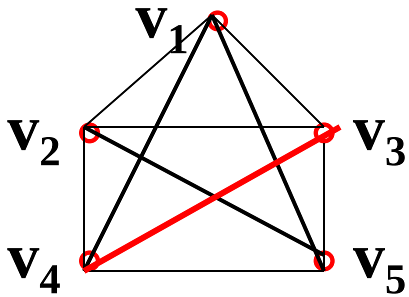
设G是无向图, 如果能将G的所有结点和边都画在一个平面上,且使得任何两条边除了端点外没有其它交点, 则称G是个**平面图**。 一个图表面上是个非平面图, 如果通过改变边的位置就变成平面图, 称此图是**可平面化的**。

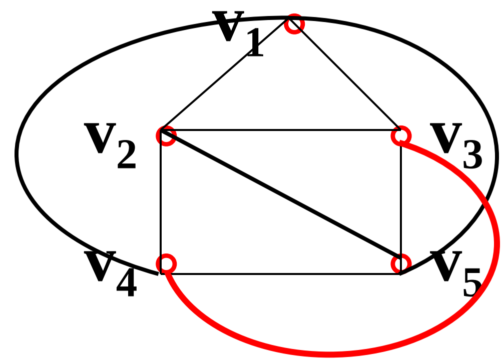
例如右图.就是
可平面化的图.

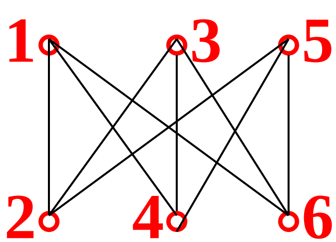


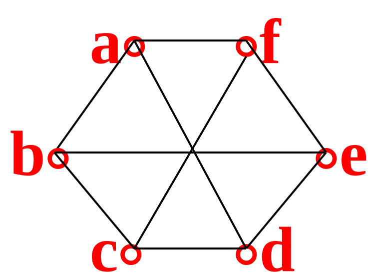


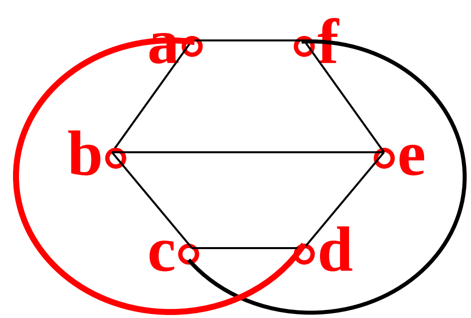
下面是两个
重要的非平面图:
K_5和$K_{3,3}$



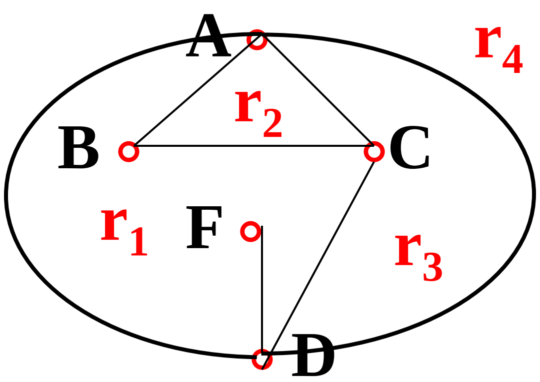








2. 平面图的面、边界及面的次数



设G是个平面图, 图中边围成的区域,其内部不含有结点, 也不含有边,称这样区域为G的一个面.

面的边界:围成一个面r的所有边构成的回路,称之为这个r面的边界. 此回路中的边数,称之为r面的次数,记作deg(r).

有限面与无限面: 面的面积有限称为有限面, 反之称为无限面. 所有平面图的外侧都有一个无限面.

例如,上图中　r_1: 边界:ABCDFDA　$\deg(r_1)=6$

r_2: 边界:ABCA　$\deg(r_2)=3$

r_3: 边界:ACDA　$\deg(r_3)=3$

r_4: 边界:ADA　$\deg(r_4)=2$

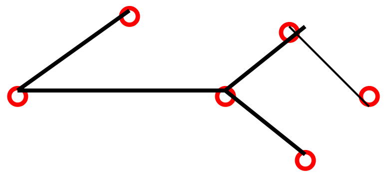
3.欧拉公式

定理8-7.1 G是个连通的平面图, 设v、e、r分别表示G中结点数、边数、面数, 则有 $v-e+r=2$. 称此式为欧拉公式.

证明: (对面数r归纳证明)

⑴当r=1 时, 此时图是连通无回路的—树, 则总是有 $e=v-1$, 于是

$v-e+r=v-(v-1)+1=2$ 结论成立.



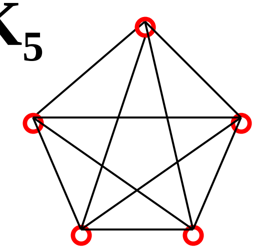
⑵假设当G有 $r \le k-1$ 个面时, 结论成立.

⑶当G有r=k 个面且是连通图时, 当 $k \ge 2$ 时, 至少有一个回路, 所以去掉此回路中的一条边后得到子图G', G'中有k-1个面, 结点数同G中结点数, 由⑵得 $v-(e-1)+(k-1)=2$

整理得 $v-e+k=2$ 即 $v-e+r=2$ 定理得证.

4.平面图的判定

K_5



*定理8-7.2(必要条件) 设G是有v 个结点、e条边的连通简单平面图, 若v≥3, 则 e≤3v-6.

证明:(1) 当e=2 时, 因为G是简单连通图, 所以v=3, 显然有

$2\leq 3\times 3-6$ 即$e\leq 3v-6$

(2)当e>2时, (通过计算每个面的边界来证明)

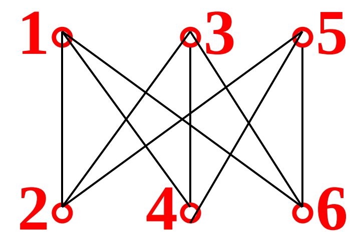
设G有r个面, 因为G是简单图, 所以每个面至少由三条边围成, 所以r个面的总边界数≥3r, 另外由于每条边在两个面的边界中出现, 所以所有面的边界总数=2e, 所以有:

2e=(r个面边界总数)≥ 3r, 即2e≥3r 所以$r\leq \frac{2}{3}e$

由欧拉公式: v-e+r=2 得 $v-e+\frac{2}{3}e\geq 2$ 整理得 $e\leq 3v-6$

用此定理可以判定一个图不是平面图, 例如证明K_5不是平面图: K_5中有v=5 e=10 $3v-6=3\times 5-6=9$ 不满足$e\leq 3v-6$, 所以K_5不是平面图.

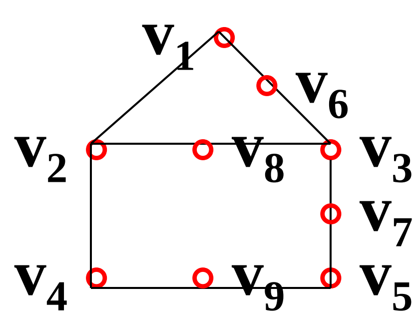
上面定理是判定平面图的必要条件,而不是充分条件. 即如果一个图 满足e≤3v-6, 它不一定是平面图. 例如, $K_{3,3}$中 v=6 e=9 9≤3×6-6 满足e≤3v-6, 但它不是平面图.

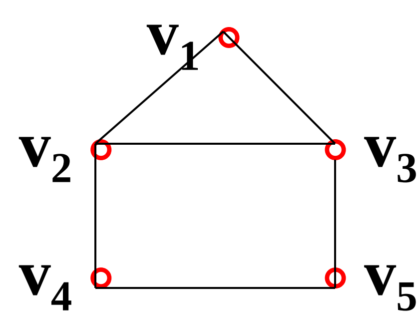


下面要介绍一个判定一个平面图的充分且必要条件, 即Kuratowski(库拉托斯基)定理. 在此之前先介绍一个新概念----在2度结点内同构(同胚).

在一个图中有2度结点, 则这些结点不影响平面的面数, 例如下面两个图:

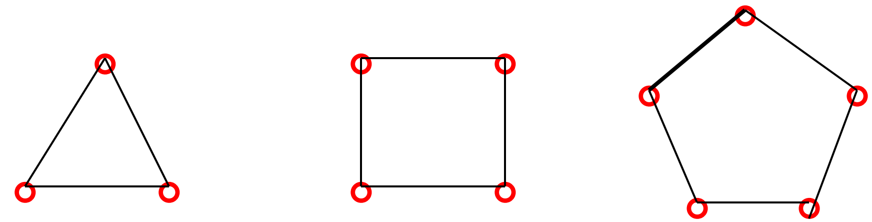
我们称这两个图是在2度结点内同构的图.



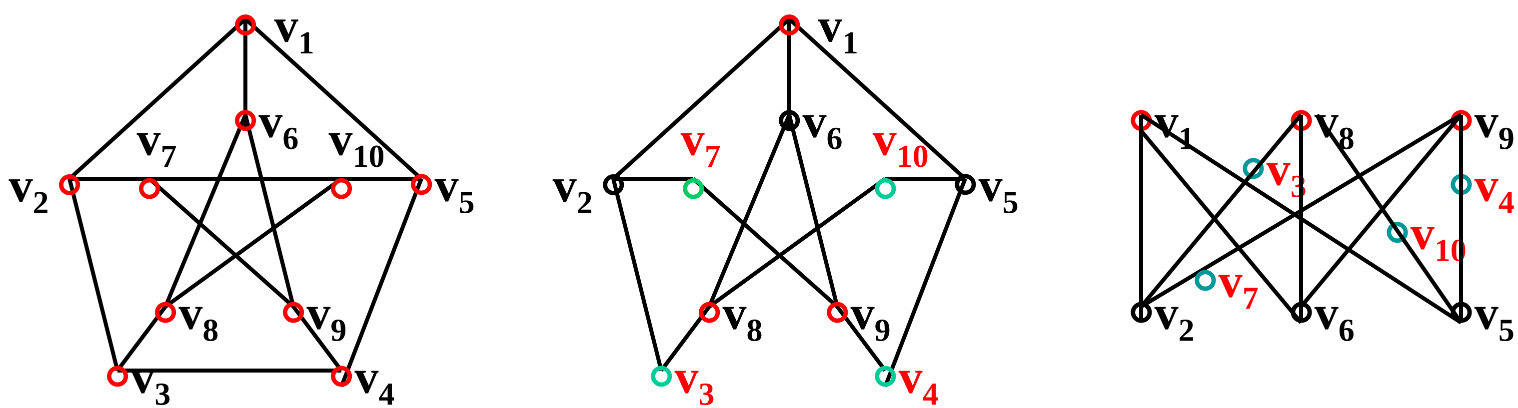


定义:如果G_1和G_2是同构的,或者通过反复插入或删去度数为2的结点, 使得它们变成同构的图, 称G_1和G_2 是在2度结点内同构.

例如右边3个图就是在2度结点内同构.

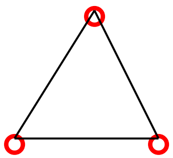


定理8-7.3 (Kuratowski定理)一个图是平面图的**充分且必要条件**是它不含有任何与K_5、$K_{3,3}$在2度结点内同构的子图. (此定理证明略.) 判断下面**彼得森(Peterser)图**:

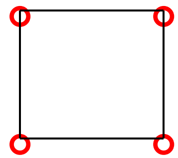


5.极大平面图

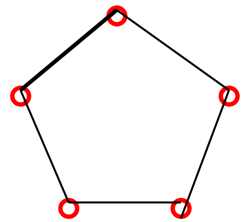
定义：设G是个简单图，令u、v是不邻接的结点，如果不能在u、v之间增加一条边而不破坏图的平面性时，则称G是极大平面图。(如果再加一条边，就不是平面图了)



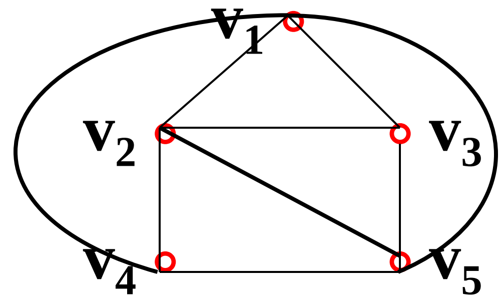
1



2

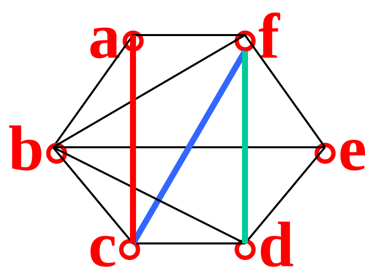


3

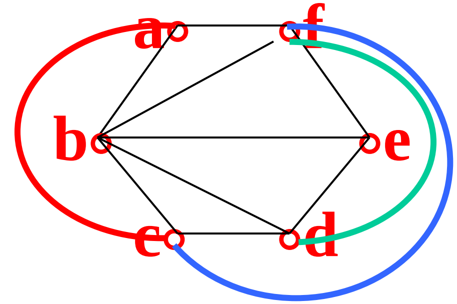


4

可见1、4、5、6是极大平面图。
而2、3不是极大平面图。



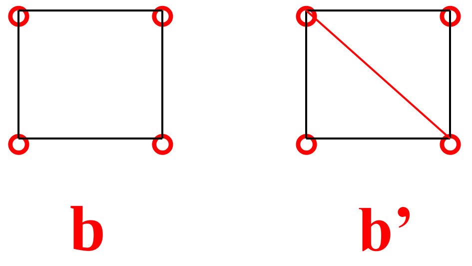
5



6

显然，极大平面图的每个面都是由三条边围成。因为如果含有四条或者四条以上的边围成的面，那么至少可以增加一条边，而不改变可平面性。

如图b和b’所示。



定理 设G是个极大平面图，且有v个结点，e条边，r个面，则

$e=3v-6$, $r=2v-4$。

证明：因为G是极大平面图，每个面由3条边围成。所以有 $3r=2e$, $r=(2/3)e$ 由欧拉公式 $v-e+r=2$，得 $r=e-v+2$，于是 $(2/3)e=e-v+2$，$2e=3e-3v+6$，所以 $e=3v-6$。

进而得 $r=e-v+2=3v-6-v+2=2v-4$。

定理得证。

本节要求掌握:

平面图的概念, 平面图的边界,

欧拉公式及其应用

平面图的判定.

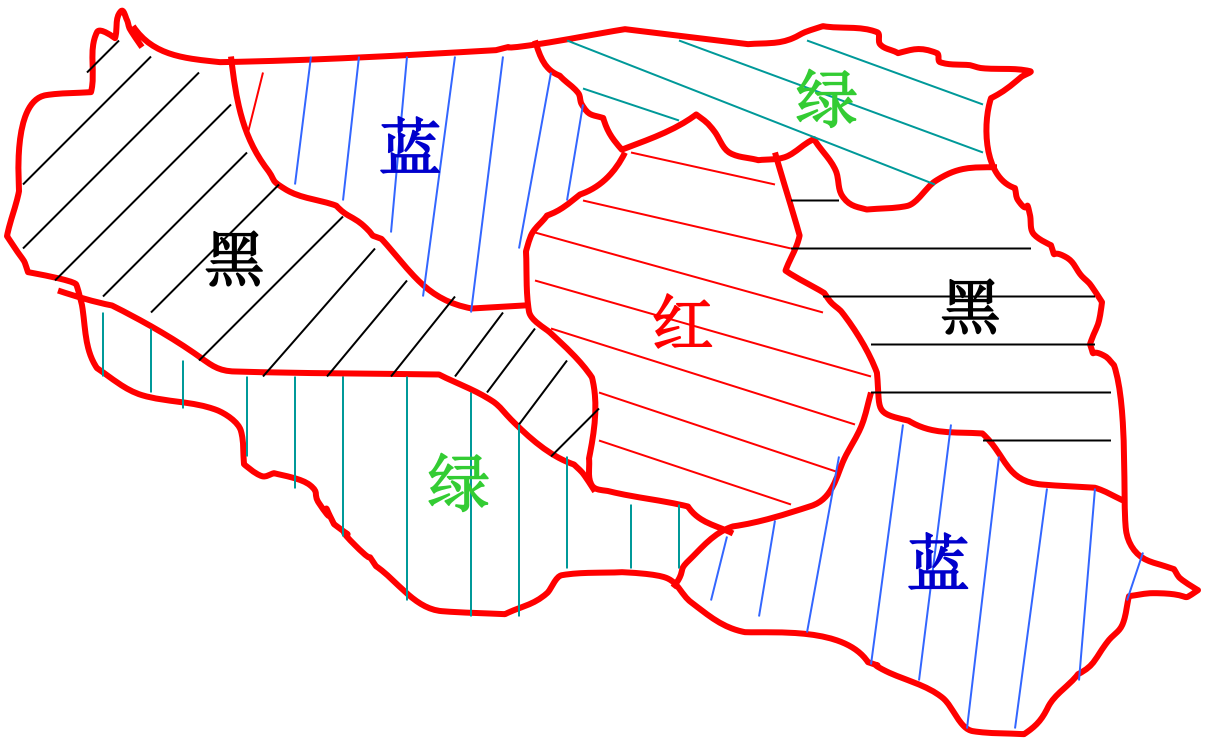
作业: P317 (1) (3) (5) (7)

8-8 着色与对偶图

着色问题起源于对地图着色, 使得相邻(不是如图所示)国家用不同颜色,需要多少种不同的颜色?

- 一百多年前,英国数学家格色里(**Guthrie**)提出了“四色猜想”, 这一猜想在图论发展史上曾起过巨大的推动作用.
- **1879**年肯普(**Kempe**)
  给出了这个猜想的第一个证明.
- 但是到**1890**年,希伍德(**Hewood**)发现肯普的证明是错误的, 可是他指出肯普的方法,虽然不能证明地图着色用四种颜色,但可以证明五种颜色就够了.

- 直到1976年美国伊利诺斯(Illinois)大学的阿佩尔(K.Appel)和黑肯(W.Haken)把四色问题归结为2000个不同的组合结构图形,利用三台高速IBM360计算机对这些图形进行分析用了1200机时,近百亿次逻辑判断, 证明了“四色定理”.



一.对偶图(偶图)的定义:

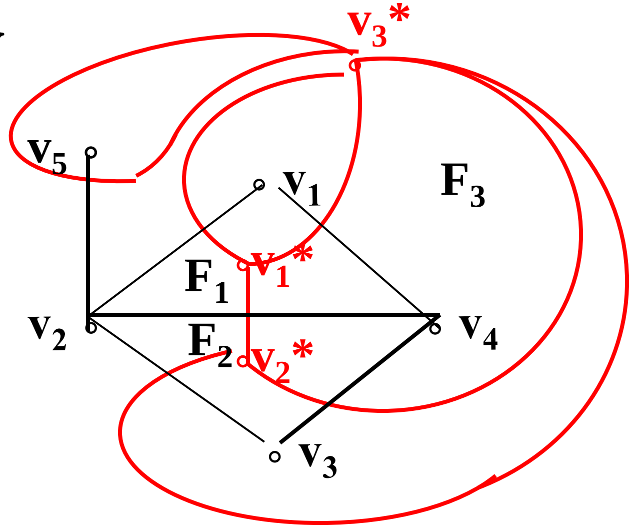
给定平面图G=<V,E>, 具有平面$F_1,F_2,F_3,\ldots,F_n$. 如果有图G*=<V*,E*>, 满足下面条件:

(1)对于G的任意平面F_i 的内部有且仅有一个结点$v_i^* \in V^*$.

(2)对于图G的面F_i与 F_j的公共边界e_k ,有且仅有一条边 $e_k^* \in E^*$,使得 $e_k^*=(v_i^*,v_j^*)$, 且e_k^*与e_k相交.(v_i^*在F_i内, v_j^*在F_j内)

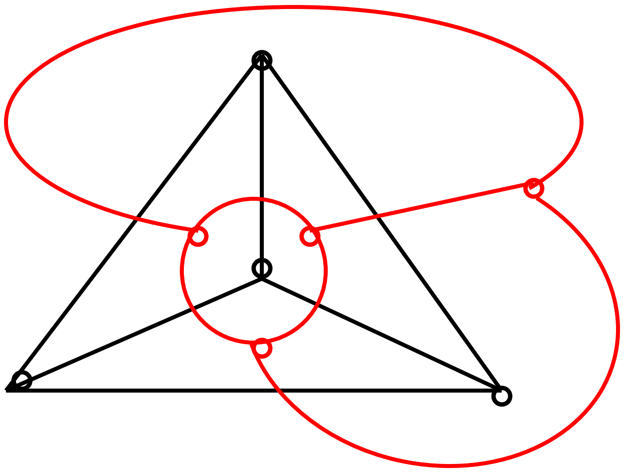
(3)当且仅当e_k只是一个面F_i的边界时, v_i^*上有一个环e_k^* 与e_k相交.

则称图G*是G的对偶图.



可见G*中的结点数等于G中的面数.

二. 自对偶图:如果图G对偶图G*与G同构,则称G是自对偶图. (如下图)



三.对偶图与平面图着色的关系:

对平面图的相邻面用不同颜色的着色问题,可以归结到对其对偶图的相邻接的结点着不同颜色。

四. 图G的正常着色(简称着色):

1. 对G的每个结点指定一种颜色,使得相邻接的两个结点着不同颜色. 如果G着色用了n种颜色,称G是 n-色的。

2.对G着色时,需要的最少颜色数,称为G的着色数,记作 $\chi(G)$ 。

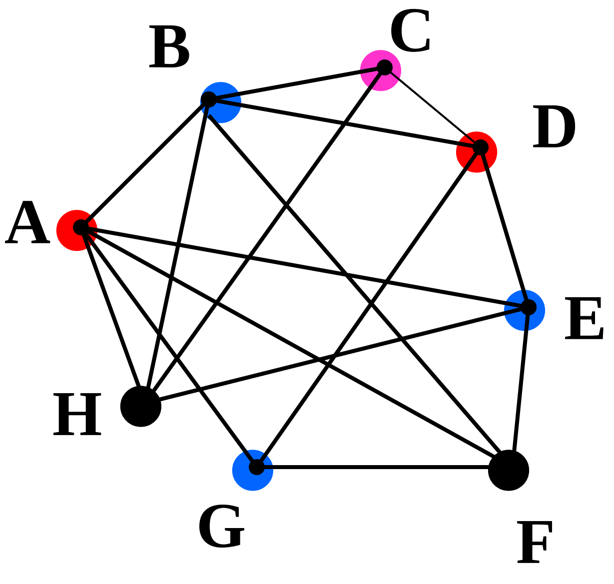
3.对G着色方法:(下面介绍韦尔奇.鲍威尔法)

韦尔奇.鲍威尔法(Welch.Powell):

(1)将G中的结点按照度数递减次序排序,(此排序可能不唯一,因为可能有些结点的度数相同)

(2)用第一种颜色对第一个结点着色,并按照排序,对与前面着色点不邻接的每一个点着上相同颜色.

(3)用另一种颜色对尚未着色的点, 重复执行(2)和(3),直到所有结点都着上颜色为止.



例如:结点排序:A,B,E,F,H,D,G,C

结点度数:5, 5, 4,4,4, 4, 3, 3

注意：在给A进行着色时，

与A不邻接的结点有D, C, 但是由于D 与C邻接，所以C不能与A着同一种颜色。

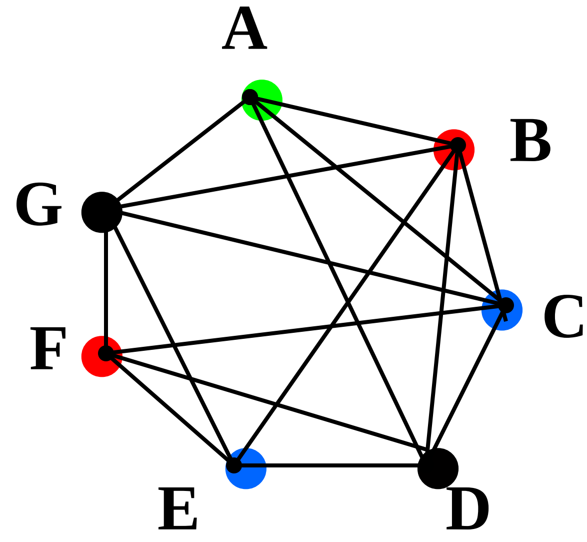
五. 应用举例

安排期末考试(学分制), 不能使一个学生在同一个时间参加两门课的考试.

设有七门课程,分别记作A,B,C,D,E,F,G. 如果两门课程有相同的学生选修, 就在两门课程之间连一直线.得到图:

结点度数递减排序:

B,C,D,G,A,E,F



对图正常着色后, 标有同一种颜色的课,可以同时考试.安排考试日程:

周一: A　　周二: B,F

周三:C,E　　周四: D,G

作业 P321 (1) (3) (7)a)b)

*六.色数的计算

1.图G只有一个孤立结点时， $\chi(G)=1$。

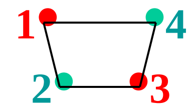
2. $\chi(Kn)=n$ 。

3.如果G是有n个结点的回路时，则

$$\chi(G)=\begin{cases} 2 & n\text{为偶数} \\ 3 & n\text{为奇数} \end{cases}$$

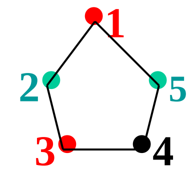
4.图G是结点数超过1的树时， $\chi(G)=2$。

5.图G是二部图时， $\chi(G)=2$。



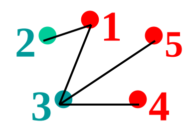
G

2色



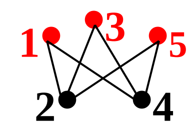
G

3色



G

2色



G

2色

树——连通无回路

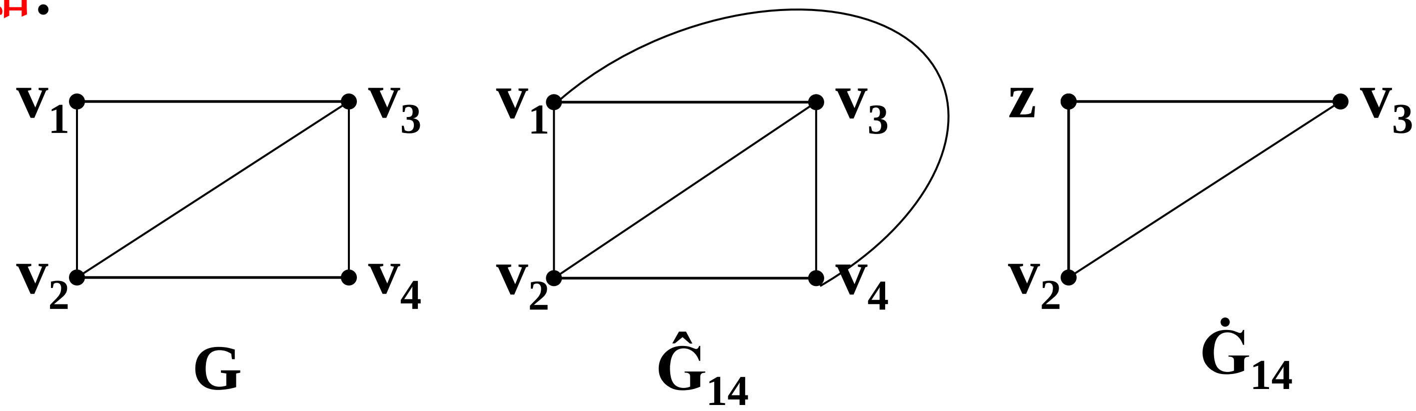
6.一般情况下色数的一种求法：

定义：设G=<V,E>， 设v_i、v_j不是邻接点，定义两个图

1) $\hat{G}_{ij}$表示在G中加上一条边(v_i,v_j)所得到的图。

2) $\dot{G}_{ij}$表示在G中把v_i与v_j缩为一点z所得到的图。即 G图中凡是与v_i、v_j关联的边都与z关联。

例如:



定理：　设v_i、v_j是图G中两个不邻接的点，则图G的色数$\chi(G)$为：　$\chi(G)=\min\{\chi(\hat{G}_{ij}), \chi(\dot{G}_{ij})\}$

证明：只要证明下面两个不等式分别成立即可。

$$\chi(G)\geq\min\{x(\hat{G}_{ij}), x(\dot{G}_{ij})\}$$

$$\chi(G)\leq\min\{x(\hat{G}_{ij}), x(\dot{G}_{ij})\}$$

先证明第一个不等式：$\chi(G) \geq \min\{x(\hat{G}_{ij}), x(\dot{G}_{ij})\}$

设$k=\chi(G)$,故可以用k种颜色使得G的相邻结点着不同颜色。

对v_i、v_j两个结点着色不外乎有两种情况：

同色或者异色。

如果v_i、v_j不同色：则 k种颜色足以使$\hat{G}_{ij}$的相邻两个结点有不同颜色。故

$\chi(G) \geq \chi(\hat{G}_{ij})$。

如果v_i、v_j同色：则 k种颜色足以使$\dot{G}_{ij}$的相邻两个结点有不同颜色。故

$\chi(G) \geq \chi(\dot{G}_{ij})$。

所以 $\chi(G) \geq \min\{\chi(\hat{G}_{ij}), \chi(\dot{G}_{ij})\}$

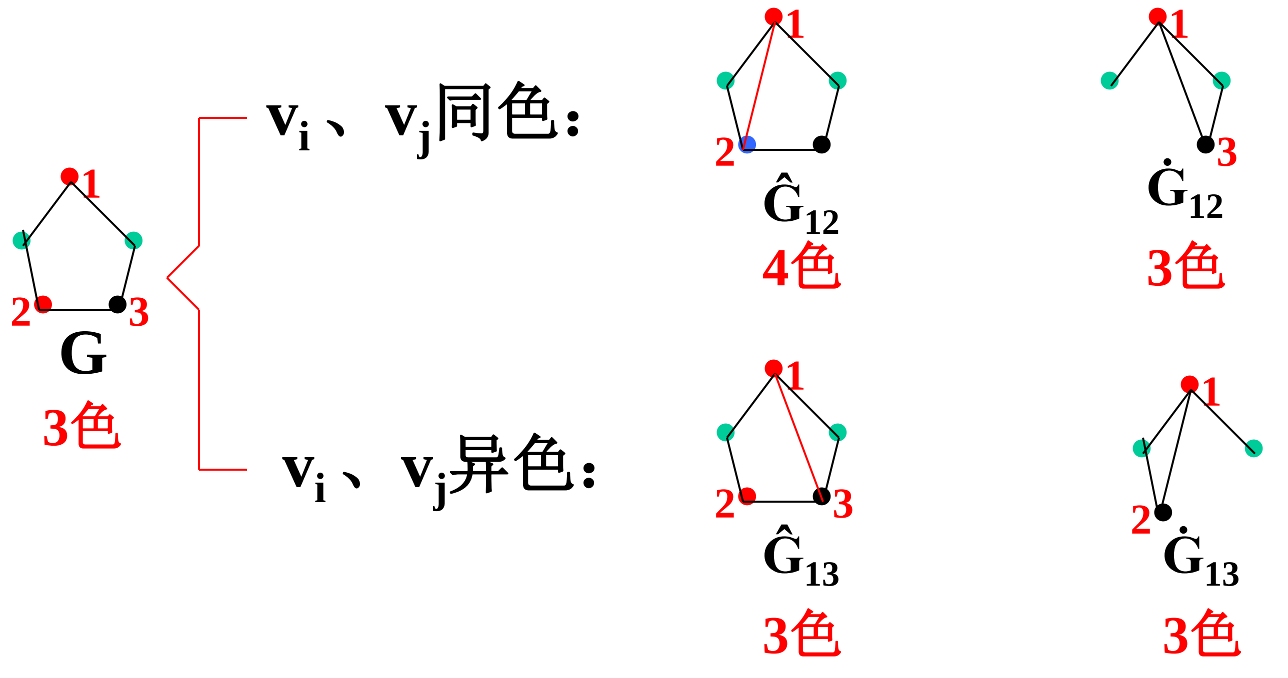
另一方面显然有

$$\chi(G)\le\chi(\hat{G}_{ij})$$

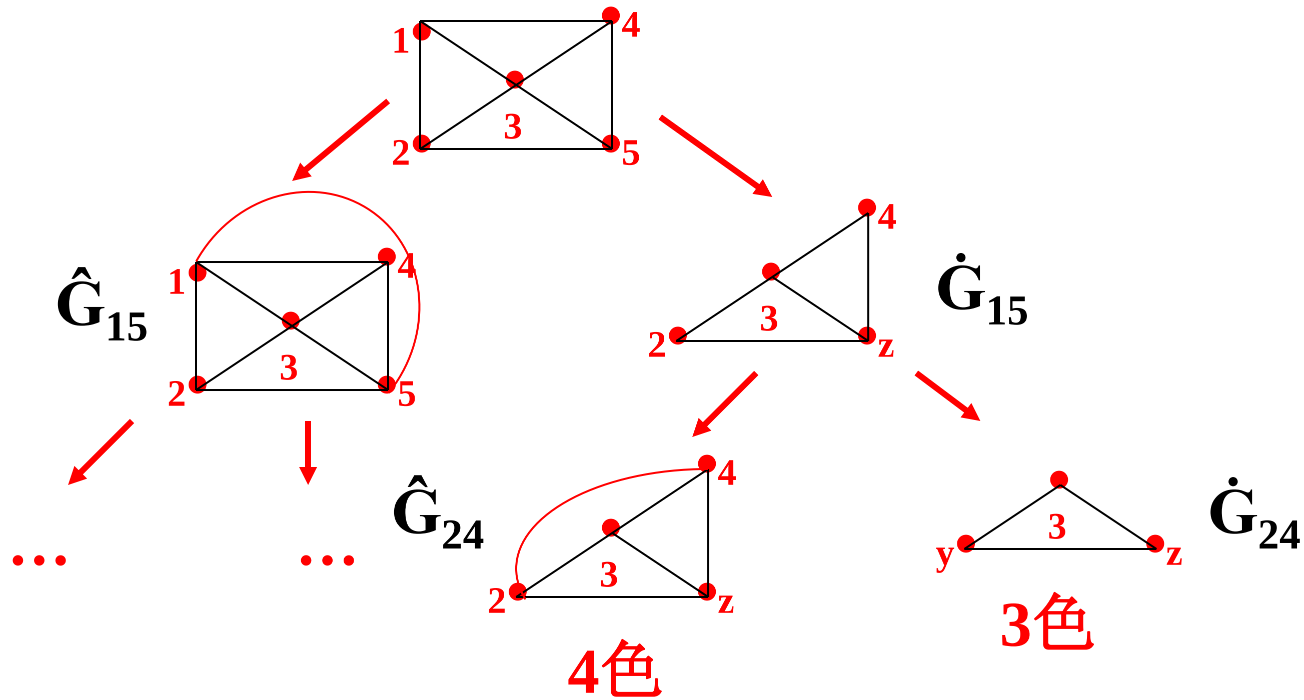
$$\chi(G)\le\chi(\dot{G}_{ij})$$

故 $\chi(G) \le \min\{\chi(\hat{G}_{ij}), \chi(\dot{G}_{ij})\}$

最后得 $\chi(G) = \min\{\chi(\hat{G}_{ij}), \chi(\dot{G}_{ij})\}$

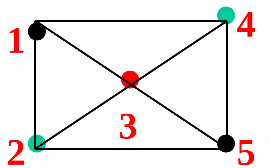


例如：



如此变化下去，直到都是完全图为止(无不相邻的结点)。

这些完全图中，边最少的图的色数是3，所以$\chi(G)=3$



*七.色数多项式

关于着色问题，伯克霍夫(Birkhoff)和刘易斯(Lewis)提出了另一个研究途经，引出了着色多项式的概念。

前面讨论的是，给定一个图的最少的着色数——$\chi(G)$。下面的问题是：给定一个有n个结点的图G，用$\lambda(\geq\chi(G))$种颜色着色，能有多少种不同的着色方式—色数多项式。

定义：设图G有n个结点，λ的多项式P(λ)的值给出颜色数目不超过λ时，图G的结点的着色不同方案数，称P(λ)为图G的着色多项式。

令m_i表示用i种颜色对G的结点进行着色的不同方案数，则用 $\lambda(\geq i)$种颜色对G进行着色，每次取i种颜色时，共有$m_i\binom{\lambda}{i}$种不同着色方案。其中$\binom{\lambda}{i}$为从λ种颜色中每次取i种颜色的不同组合数。于是有

$$P(\lambda) = m_1\binom{\lambda}{1} + m_2\binom{\lambda}{2} + \dots + m_n\binom{\lambda}{n}$$

$$= m_1\lambda + \frac{m_2}{2!}\lambda(\lambda-1) + \frac{m_3}{3!}\lambda(\lambda-1)(\lambda-2) + \dots\dots$$

$$+ \frac{m_n}{n!}\lambda(\lambda-1)(\lambda-2)\dots\dots(\lambda-n+1)$$

显然P(λ)为λ的多项式。

有两种特殊情况：

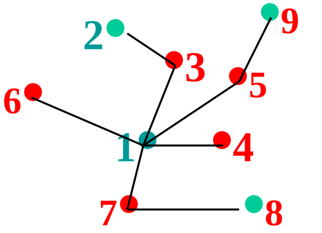
1.若G是个完全图Kn,则

$P(\lambda)=\lambda(\lambda-1)(\lambda-2)\dots(\lambda-n+1)$

因为第一个结点有λ种着色方式，第二个结点有(λ-1)种着色方式， …最后一个结点有 (λ-n+1)种着色方式。于是共有λ(λ-1)(λ-2)…(λ-n+1) 种着色方式。

2.若G是棵树(连通无回路的图)，则

$$P(\lambda)=\lambda(\lambda-1)^{n-1}$$



G 2色

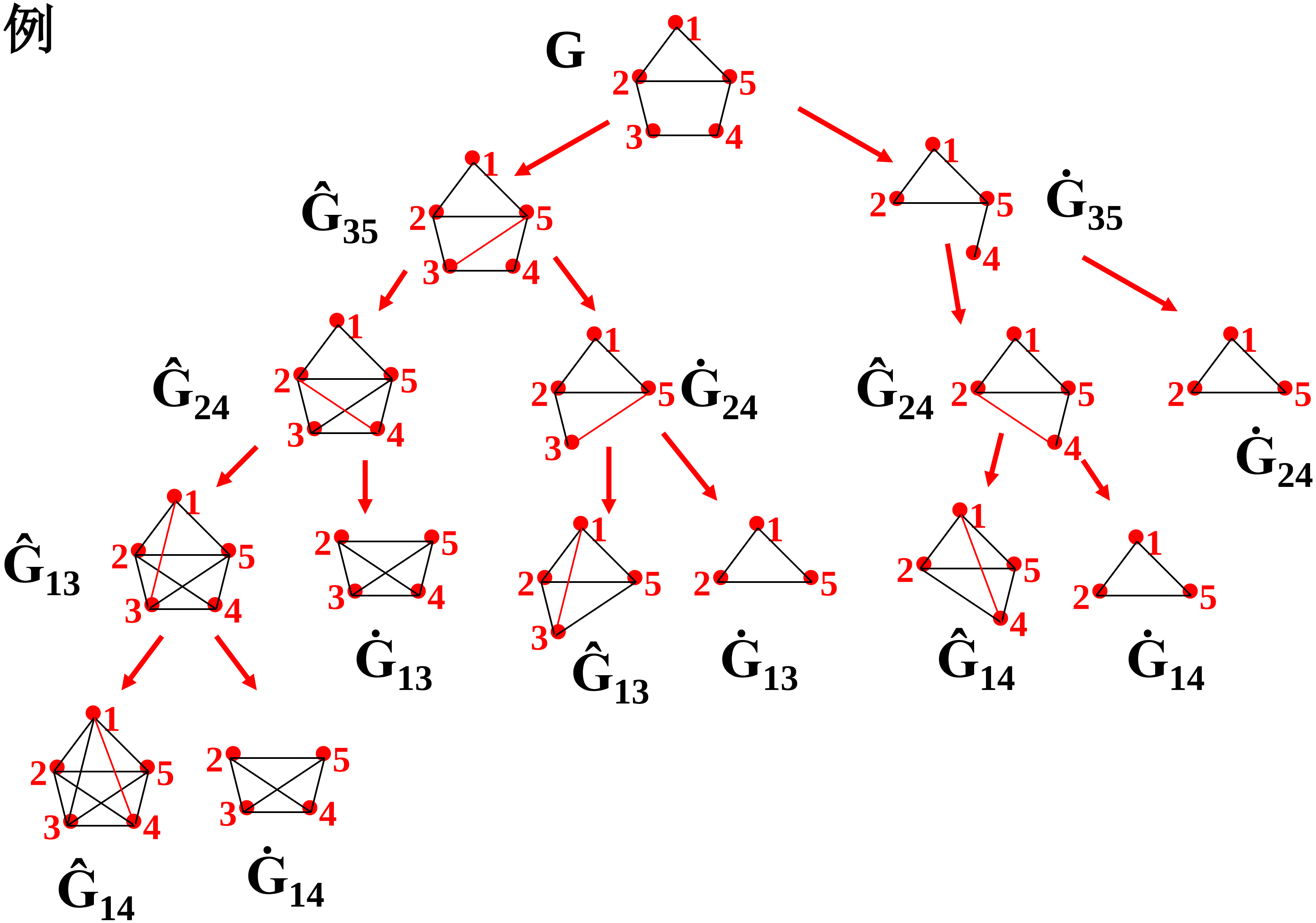
因为第一个结点有λ种着色方式，而其余n-1个结点都有λ-1种着色方式。

3.定理：设v_i、v_j是图G中不邻接的两个结点，设$P_1(\lambda)$是$\hat{G}_{ij}$的色数多项式，设$P_2(\lambda)$是$\dot{G}_{ij}$的色数多项式，则G的色数多项式为

$$P(\lambda) = P_1(\lambda) + P_2(\lambda)$$

证明：用$\lambda(\geq \chi(G))$种颜色对G的结点进行着色，其结果可以分成两类：一是v_i、v_j不同色，一是v_i、v_j同色。前一种有$P_1(\lambda)$种方式(因v_i与v_j之间连一条边后，就要着不同颜色，所以等同于$\hat{G}_{ij}$的着色方案数)，后一种有$P_2(\lambda)$方式(因这两结点同色，就相当于两点合并成一个结点后，$\dot{G}_{ij}$的着色数)。于是有 $P(\lambda) = P_1(\lambda) + P_2(\lambda)$。

例



从此图看出，最终有一个K_5，4个K_4，3个K_3。

最后得到一个K_5，4个K_4，3个K_3。

因完全图Kn的着色数 $P_n(\lambda)=\lambda(\lambda-1)(\lambda-2)\ldots(\lambda-n+1)$

$P(\lambda)= P_5(\lambda)+4P_4(\lambda)+3P_3(\lambda)$

$= (\lambda(\lambda-1)(\lambda-2)(\lambda-3)(\lambda-4)) +4(\lambda(\lambda-1)(\lambda-2)(\lambda-3)) +$

$3(\lambda(\lambda-1)(\lambda-2))$

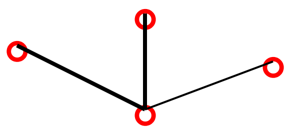
$= (\lambda(\lambda-1)(\lambda-2))((\lambda-3)(\lambda-4) +4(\lambda-3) +3)$

$= (\lambda(\lambda-1)(\lambda-2))((\lambda^2-7\lambda+12) +4\lambda-12 +3)$

$= (\lambda(\lambda-1)(\lambda-2))((\lambda^2-3\lambda+3)$

8-9 树与生成树

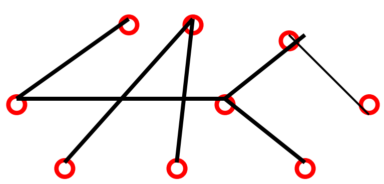
树是一种特殊的图, 它是图论中重要的概念之一, 它有着广泛的应用.在计算机科学中有如判定树、语法树、分类树、搜索树、目录树等等.



(a)

一.树 (Tree)

1.树的定义:一个连通无回路的无向图T,称之为树. 如(a)



(b)

2.叶结点:度数为1的结点, 称为叶结点.

3.分支结点(内结点):度数大于1的结点.

4.森林:一个无向图的每个连通分支都是树.如(b)

5.与树定义等价的几个命题

定理8-9.1给定图T, 以下关于树的定义是等价的。

(1)无回路的连通图。

(2)无回路且e=v-1 其中e是T的边数,v是T的结点数。

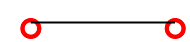
(3)连通的且e=v-1。

(4)无回路但添加一条新边则得到一条仅有的回路。

(5)连通的,但删去任一条边,T便不连通。

(6)每对结点之间有一条且仅有一条路。

证明:(1)⇒(2):已知T是连通无回路图,通过不断地增加T中的结点数,归纳证明。



当 v=2时, T如右图所示,e=1 显然e=v-1。

以后对T在保证连通又无回路的前提下每增加一个结点,也增加一条边. 设最后T有v个结点e条边, 所以 e=v-1。

(2)⇒(3): 已知T是无回路的,且$e=v-1$。(推出T是连通的)

假设T不是连通的,设T有k个连通分支, $T_1,T_2,\ldots,T_k,(k\geq 2)$

因为它的每个连通分支都是连通无回路的,所以都是树,

设T_i有结点数v_i,边数e_i, 所以边数 $e_i=v_i-1$

设T有v个结点,e条边. 所以

$$v= v_1+v_2+v_3+\ldots+v_k$$

$$e= e_1+e_2+e_3+\ldots+e_k=(v_1 -1)+(v_2 -1)+(v_3 -1)+\ldots+(v_k -1)$$

$$=(v_1+v_2+v_3+\ldots+v_k)-k=v-k$$

但是已知 $e=v-1$ 所以 $k=1$ 所以T是连通图。

(3)⇒(4):已知T是连通的且e=v-1(推出T无回路,且加一新边,得到仅有一条回路)假设T有回路C_1,从C_1中删去一条边以后T仍然连通性,如果还有回路C_2,再从C_2中删去一条边,如此下去.假设共删去k条边,就无回路了,得到树T',设T'有边e', ∴e'=v-1, ∴ e-k=v-1 又已知 e=v-1 ∴k=0 即T无回路

再证明:加一新边,得到仅有一条回路

假设加一条新边(u,v),如果u与v是邻接点,那么新边与原来(u,v)边构成平行边,自成一个回路. 如果u与v不邻接, 由于T是连通的,u间v必有一条路径P,加上新边(u,v)后,就与P构成回路, 且此回路必是唯一的.因为如果回路不唯一, 则删去(u,v)边后,还有回路, 与上面证出的T无回路矛盾.

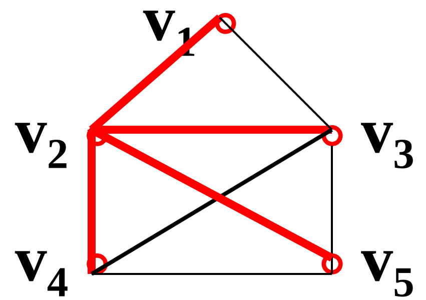
(4)⇒(5):已知T无回路,且加一条边得到仅有的一条回路.
(推出T是连通的,且删去一条边后T就不连通了)
假设T不是连通的,则存在两个结点v_i与v_j之间无路, 于是加上边(v_i,v_j)不会产生回路, 这与已知矛盾. 故T树连通的.
因T是连通无回路的, 故删去任何一条边后,T就不连通了.
(5)⇒(6):已知T是连通的,且删去一条边后T就不连通了.
(推出每对结点之间有且仅有一条路)
由T是连通图,则任何两个结点间都有一条路. 如果有两个结点间有多于一条的路, 那么T必有回路, 则删去回路中的一条边后,T仍然是连通的. 与已知矛盾.
(6)⇒(1):已知T每对结点间有且仅有一条路(推出T连通无回路)因为T 每对结点之间有一条路,所以T是连通的.若T有回路,则回路上任何两个结点间有两条路,与已知矛盾.

二. 生成树(支撑树)

在图论的应用中,找出一个连通图的所有不同的生成树,以及找出最小生成树是很有意义的.

1.定义: 如果图G的生成子图是树,则称此树为G的生成树.

2.弦: 图G中,不在其生成树里的边,称作弦. 所有弦的集合,称为该生成树的补.



定理8-9.2 连通图至少有一棵生成树.

证明: 如果G中无回路, 则G本身就是树.

如果G中有回路,可以通过反复删去回路中的边,使之既无回路,又连通.就得到生成树.

思考题: 设G是有n个结点,m条边的连通图, 问要删去多少条边,才得到一棵生成树?

*3. 连通图生成树的个数

这里讨论给定连通图G，它有多少个不同的生成树。先介绍 比内一柯西(Binet-Cauchy)定理。

1).定理(Binet-Cauchy定理)：给定两个矩阵

$A=(a_{ij})_{m\times n}$ $B=(b_{ij})_{n\times m}$

且$m\leq n$，则

$$\det(AB)=\sum_{(i)}A_iB_i$$

其中det(AB)是AB对应的行列式。则A_i、B_i是m阶行列式，A_i是从矩阵A中取第 j_1、j_2、…j_m列组成的行列式，而B_i正好是从矩阵B中取相应 的第j_1、j_2、…j_m行组成的行列式，$\sum\limits_{(i)}$ 是对所有排列j_1、j_2、…j_m 求和，即

$$\det(AB)=\sum_{1\le j1\prec j2\prec\ldots\prec jm\le n}\begin{vmatrix} a_{1j1} & a_{1j2} & \ldots & a_{1jm}\\ a_{2j1} & a_{2j2} & \ldots & a_{2jm}\\ \cdots & \cdots & \cdots & \cdots\\ a_{mj1} & a_{mj2} & \ldots & a_{mjm}\end{vmatrix}\begin{vmatrix} b_{j11} & b_{j12} & \ldots & b_{j1m}\\ b_{j21} & b_{j22} & \ldots & b_{j2m}\\ \cdots & \cdots & \cdots & \cdots\\ b_{jm1} & b_{jm2} & \ldots & b_{jmm}\end{vmatrix}$$

定理证明从略。

例如

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

$$AB=\begin{pmatrix}1 & 2 & 1\\ -1 & 3 & 1\end{pmatrix}\begin{pmatrix}2 & 1\\ 0 & 1\\ 1 & 1\end{pmatrix}=\begin{pmatrix}3 & 4\\ -1 & 3\end{pmatrix}$$

根据Binet-Cauchy定理得：

A:12列,B:12行　A:13列,B:13行　A:23列,B:23行

$$\det(AB)=\begin{vmatrix}3 & 4\\ -1 & 3\end{vmatrix}=\begin{vmatrix}1 & 2\\ -1 & 3\end{vmatrix}\begin{vmatrix}2 & 1\\ 0 & 1\end{vmatrix}+\begin{vmatrix}1 & 1\\ -1 & 1\end{vmatrix}\begin{vmatrix}2 & 1\\ 1 & 1\end{vmatrix}+\begin{vmatrix}2 & 1\\ 3 & 1\end{vmatrix}\begin{vmatrix}0 & 1\\ 1 & 1\end{vmatrix}$$

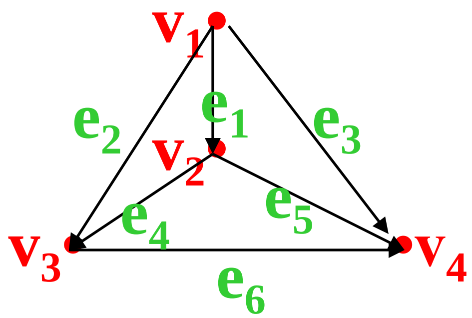
$$=5\bullet 2+2\bullet 1+(-1)\bullet(-1)=13$$

2).用Binet-Cauchy定理求生成树的个数

(1)基本关联矩阵

具有n个结点的连通图G，它的关联矩阵B的秩是n-1。

例如，给定图G如图所示。它的关联矩阵为：



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

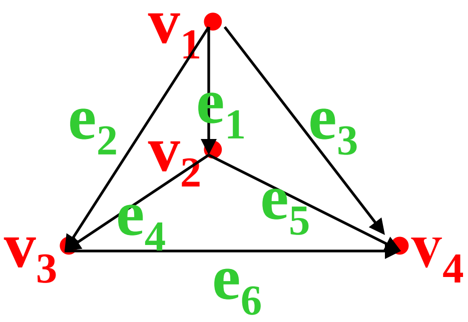
显然矩阵B的秩为3。

定义：从具有n个结点的连通图G的关联矩阵B中去掉结点v_k对应的一行得到的(n-1)×m矩阵B_k,称之为对应于结点v_k的基本关联矩阵。

定理：B_k是连通图G的基本关联矩阵，则G的生成树的个数为$\det(B_k B_k^T)$。

证明从略。

应用此定理，求前面图的生成树个数，由关联矩阵得基本关联矩阵B_4及B_4^T如下：



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

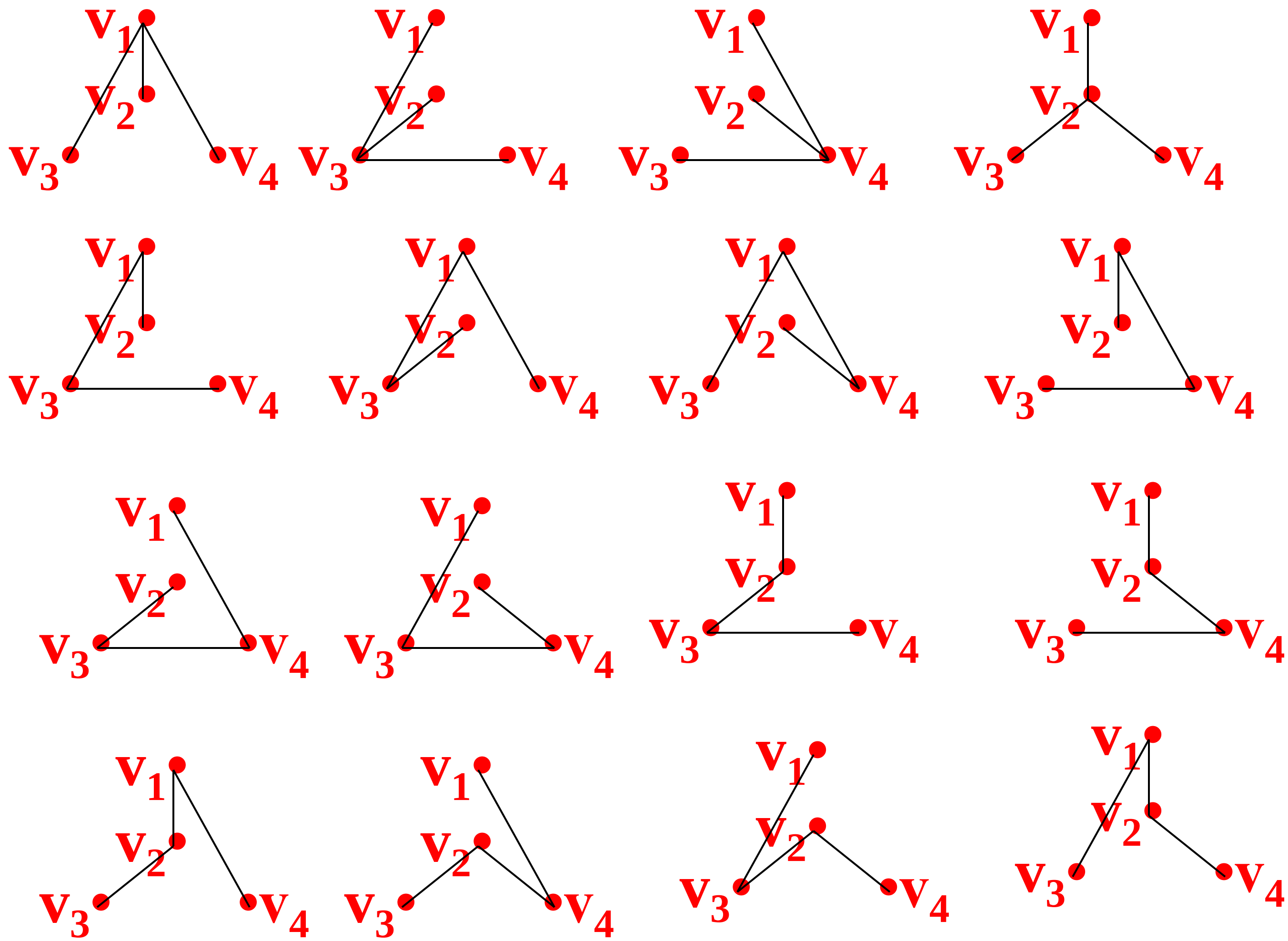
$$B_4 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \end{bmatrix} \qquad B_4^T = \begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{B_4 B_4^T=}\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & -1 & 0 \\ 1 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}=\begin{bmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{bmatrix}$$

于是G有生成树的个数为：

$$\det(B_4 B_4^T)=\begin{vmatrix} 3 & -1 & -1 \\ -1 & 3 & -1 \\ -1 & -1 & 3 \end{vmatrix} \mathbf{=27-1-1-3-3-3=16}$$

这16棵树如后面所示：



实际上只有两种不同构的生成树。(图中边的方向没画)，若给定图是无向图，可随便给各边加个方向，变成有向图后，再按照上述方法处理。

3)求连通图G生成树个数的另一个方法:

Cayley公式：

定义：$e\in E(G)$，将e从G中去掉后的图记作G-e。而将e的两个结点重合在一起，称作收缩边e，这样得到的图记作G.e 。

定义：图G的生成树个数记作$\tau(G)$。

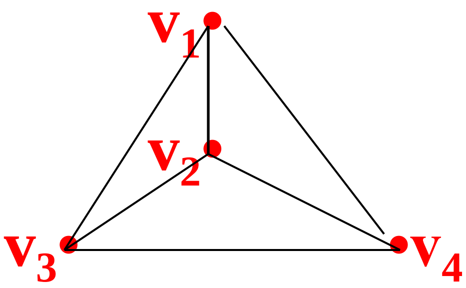
定理(Cayley公式)：G是个连通图， $e\in E(G)$，则

$$\tau(G)=\tau(G-e)+\tau(G.e)$$

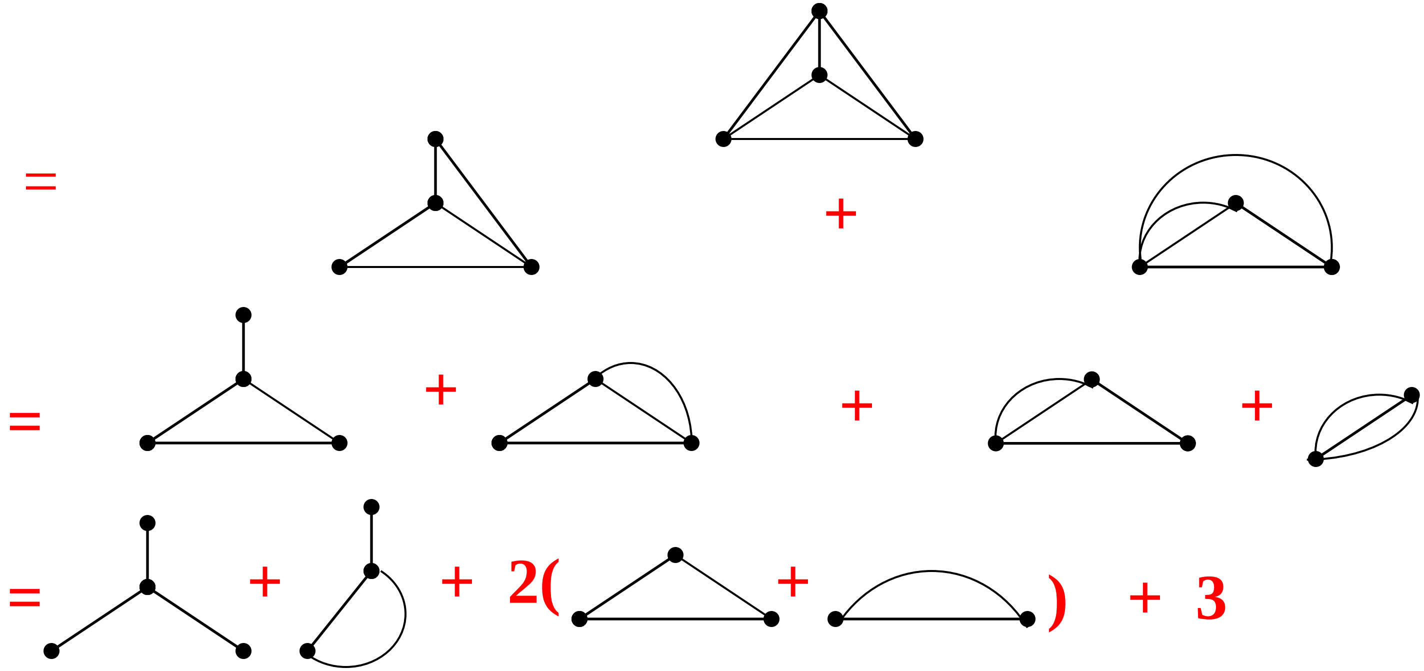
证明从略。

$\tau(G-e)$：是不含边e的生成树个数。

$\tau(G.e)$：是含有边e的生成树个数。



下面用此公式，计算$\tau(K_4)$



$$= \quad 1 \quad + \quad 2 \quad + \quad 2\,(\quad 3 \quad + \quad 2 \quad) \quad + \quad 3 = 16$$

其结果与前面相同。

定理： 完全图K_n的生成树数目$\tau(K_n)=n^{n-2}$

证明：将K_n的每个边任给定一个方向，得到有向图G，设G中结点v_k对应的基本关联矩阵B_K,于是得到

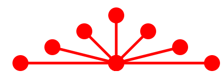
$$\det(B_k B_k^T)=\det\begin{bmatrix} n-1 & -1. & \ldots\ldots & -1 \\ -1 & n-1.. & \ldots & -1 \\ \ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots & \ldots\ldots & \\ -1 & -1 & \ldots\ldots & n-1 \end{bmatrix}=n^{n-2}$$

从前面的例子知道K_4的生成树数目是16。

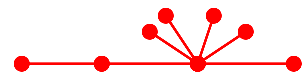
应用此公式可以验证$4^{4-2}=16$。

例. 画出8个结点所有不同构的树：v=8 e=7 度数总和14，按照结点度序列有：

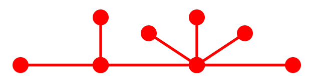
1) (1 1 1 1 1 1 1 7) 1



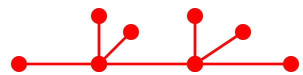
2) (1 1 1 1 1 1 2 6) 1



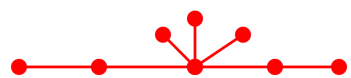
3) (1 1 1 1 1 1 3 5) 1

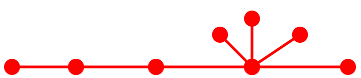


4) (1 1 1 1 1 1 4 4) 1

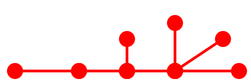


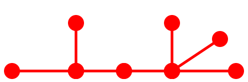
5) (1 1 1 1 1 2 2 5) 2

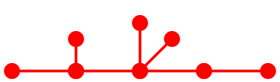




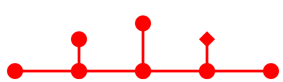
6) (1 1 1 1 1 2 3 4) 3



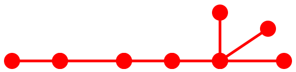


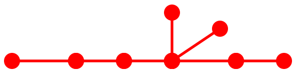


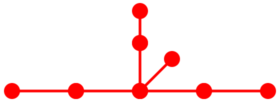
7) (1 1 1 1 1 3 3 3) 1



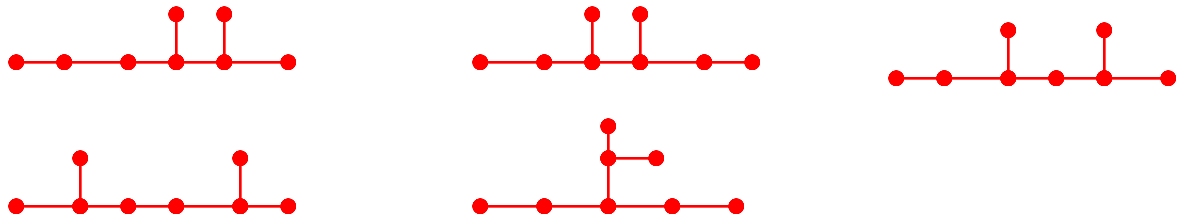
8) (1 1 1 1 2 2 2 4) 3



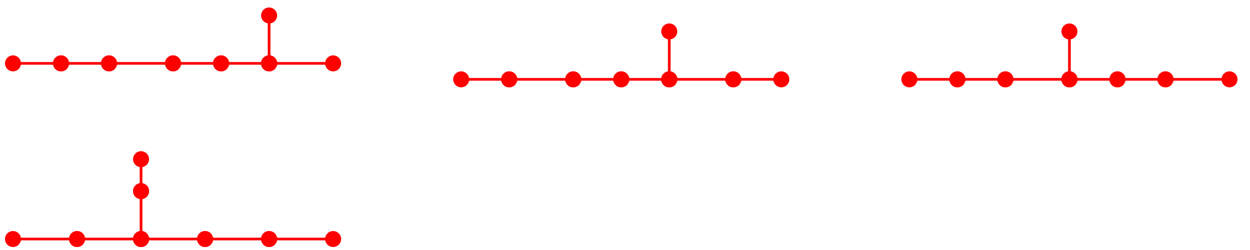




9) (1 1 1 1 2 2 3 3) **5**



10) (1 1 1 2 2 2 2 3) **4**



11) (1 1 2 2 2 2 2 2) **1**

共有23棵不同构的树

三.赋权图的最小生成树

1.定义:一棵生成树中的所有边的权之和称为该**生成树的权**. 具有最小权的生成树,称为**最小生成树**.

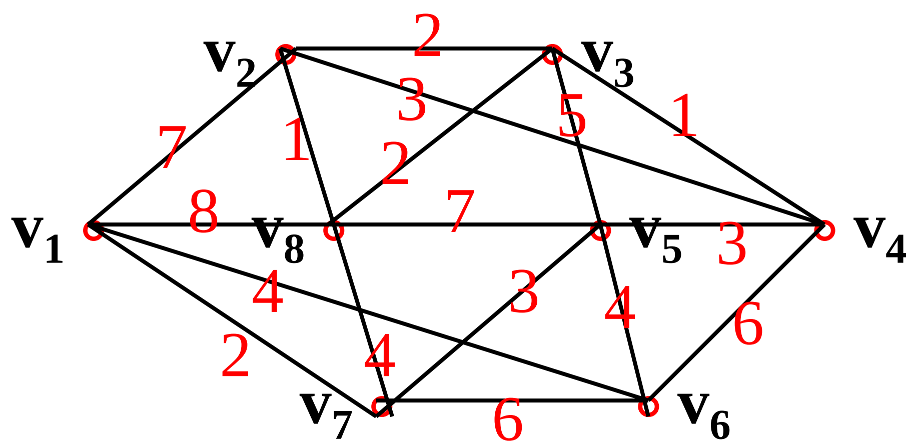
最小生成树很有实际应用价值.例如结点是城市名,边的权表示两个城市间的距离, 从一个城市出发走遍各个城市,如何选择最优的旅行路线.又如城市间的通信网络问题,如何布线,使得总的线路长度最短.

例如:右图所示

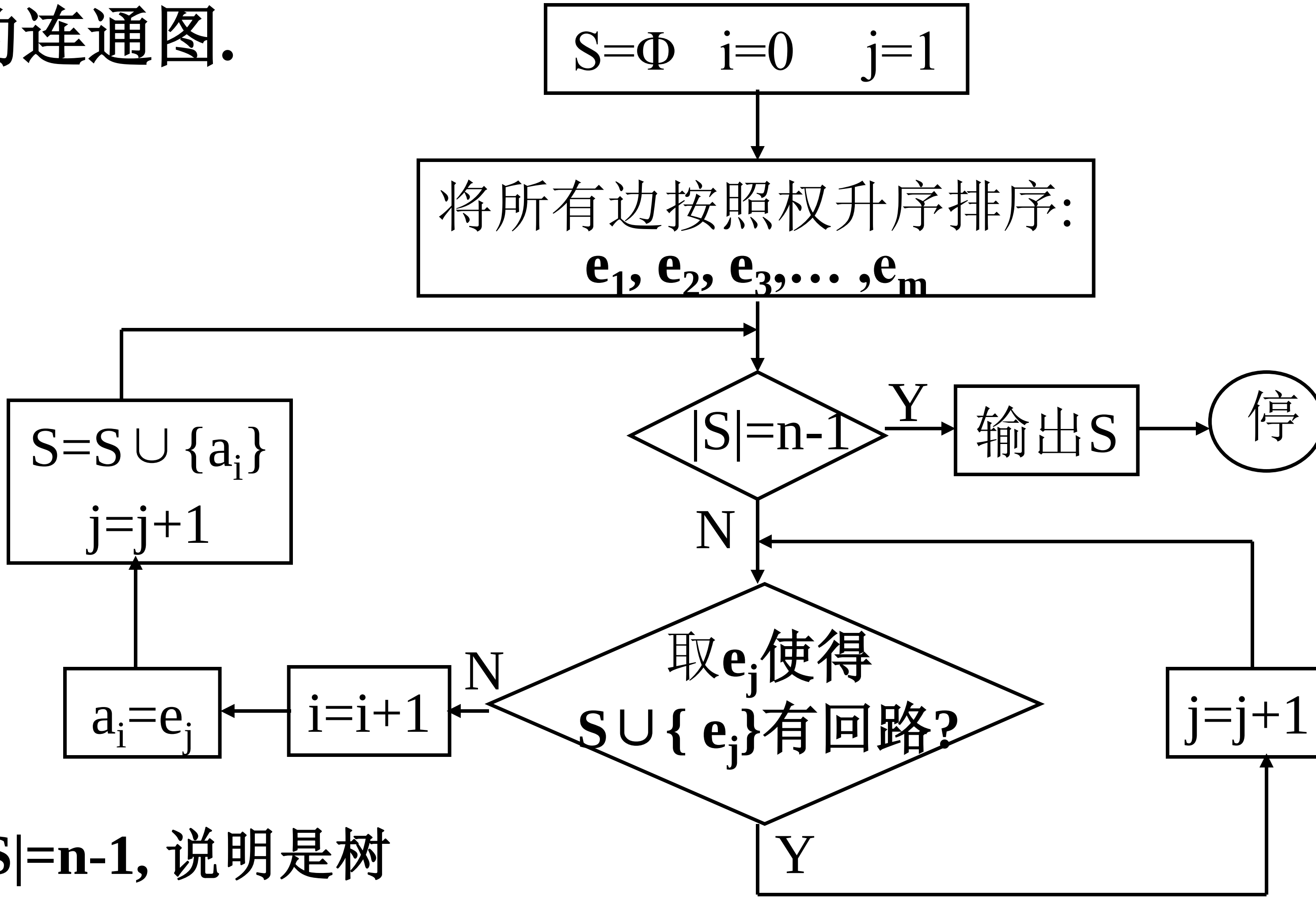
2.求最小生成树算法

---Kruskal算法:

(避圈法)



Kruskal算法: 设G是有n个结点,m条边($m\geq n-1$)的连通图.

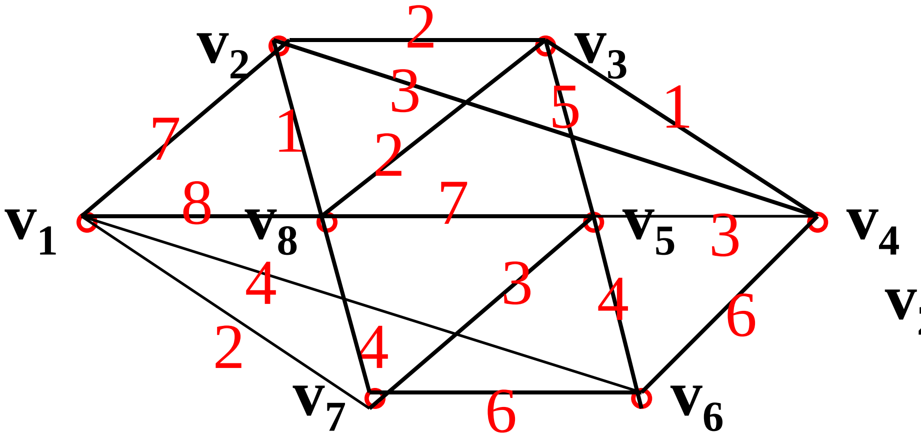


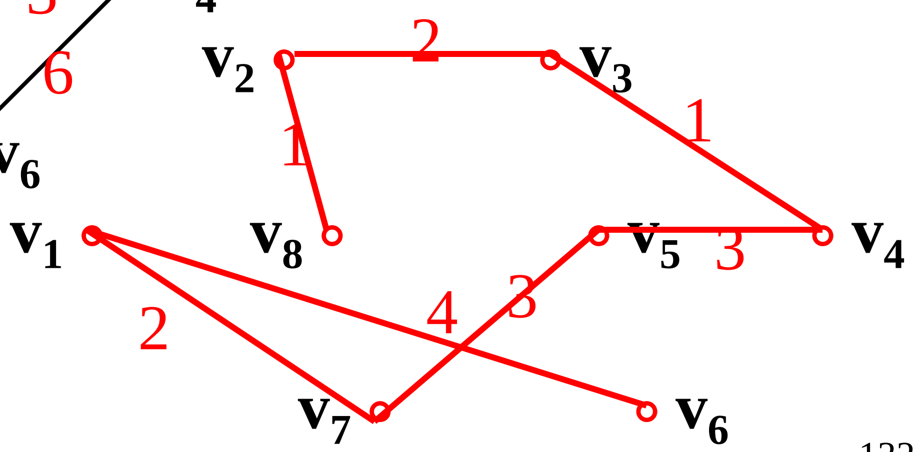
$|S|=n-1$, 说明是树

最后$S=\{a_1, a_2, a_3,\dots,a_{n-1}\}$

边按升序排序:边(v_i, v_j)记成e_{ij}

边	e_{28}	e_{34}	e_{23}	e_{38}	e_{17}	e_{24}	e_{45}	e_{57}	e_{16}
权	1	1	2	2	2	3	3	3	4
边	e_{78}	e_{56}	e_{35}	e_{46}	e_{67}	e_{58}	e_{12}	e_{18}	
权	4	4	5	6	6	7	7	8	





本节要掌握

树的6个定义,

会画生成树和最小生成树.

作业 p327 (2)(3)(6)

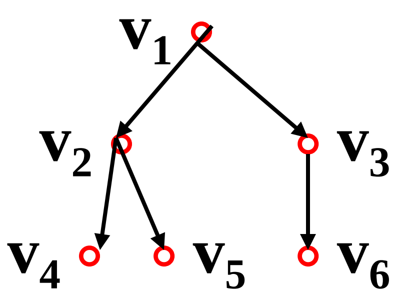
8-10. 根树及其应用

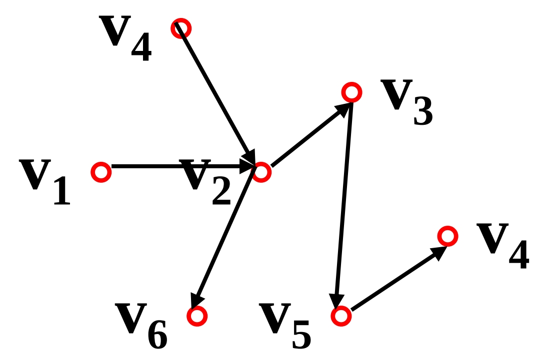
下面讨论有向树,它的应用很广泛.在计算机科学中有如判定树、语法树、分类树、搜索树、目录树等等.

一.有向树

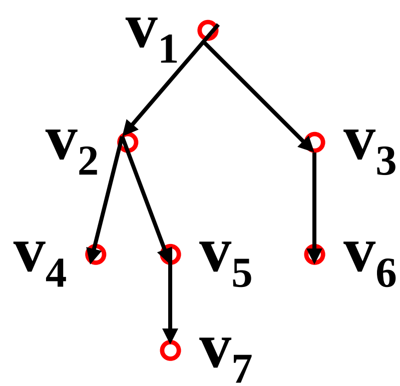
1.定义:如果G是个有向图,且在不考虑边的方向时(即看成无向图时),是一棵树,则称G是有向树.

例如:





二.根树:如果一棵有向树,恰有一个结点的入度为0,其余所有结点的入度均为1,则称此树为根树.



1.树根:入度为0的结点.

2.叶:出度为0的结点.

3.分支结点(内结点):出度不为0的结点.

4.父结点与子结点:如果 $<v_i,v_j>$ 是根树中的一条边,则称 v_i 是 v_j 的父结点, v_j 是 v_i 的子结点.

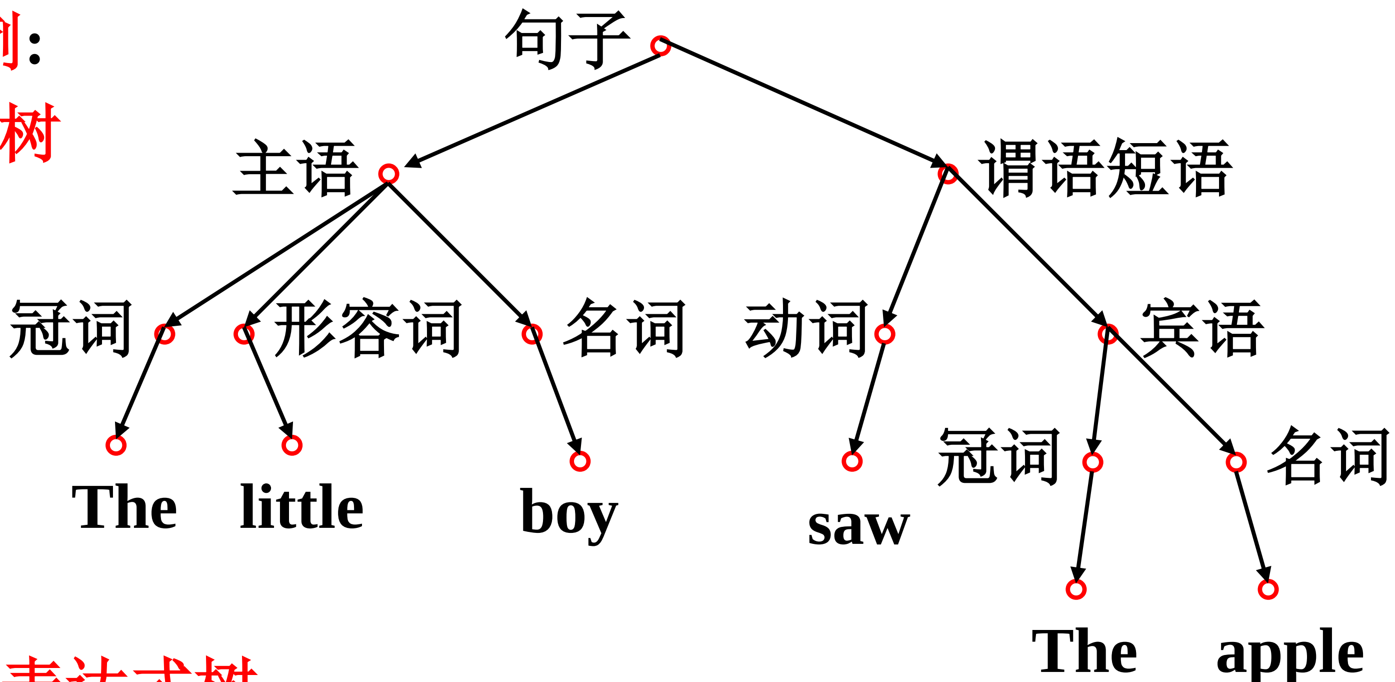
5.祖先结点与后裔结点: 在根树中,如果从 v_i 到 v_j 有路,则称 v_i 是 v_j 的祖先结点, v_j 是 v_i 的后裔结点.

6.根树结点的层次:从根结点到某个结点的路径的长度,称为该结点的层次. 同一层次的结点称为兄弟结点.

7.树高:从树根到各个叶结点的路径中, 最长路径的长度,称为该树的高度(树高).

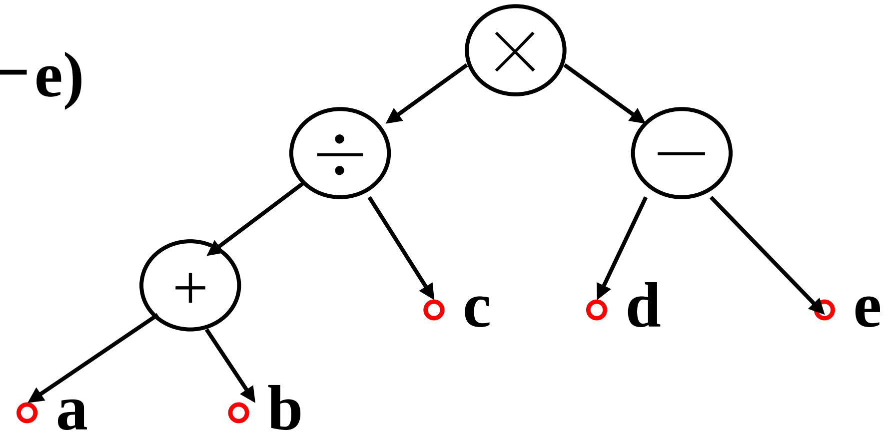
三.举例:

a)语法树

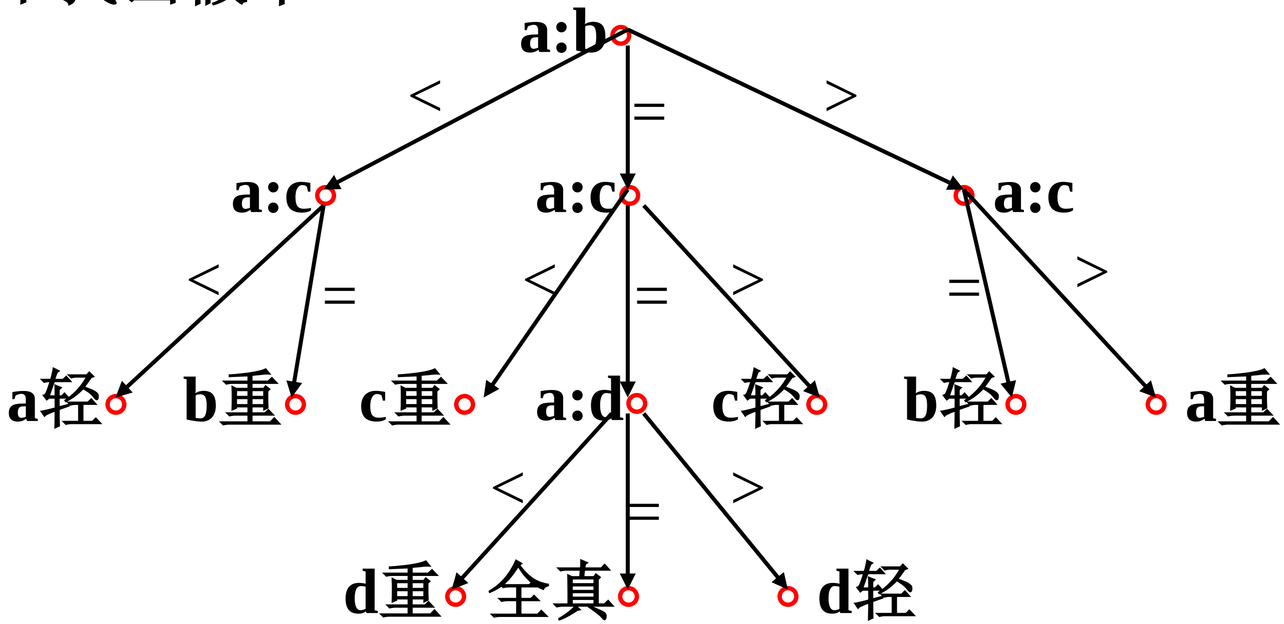


b)算术表达式树

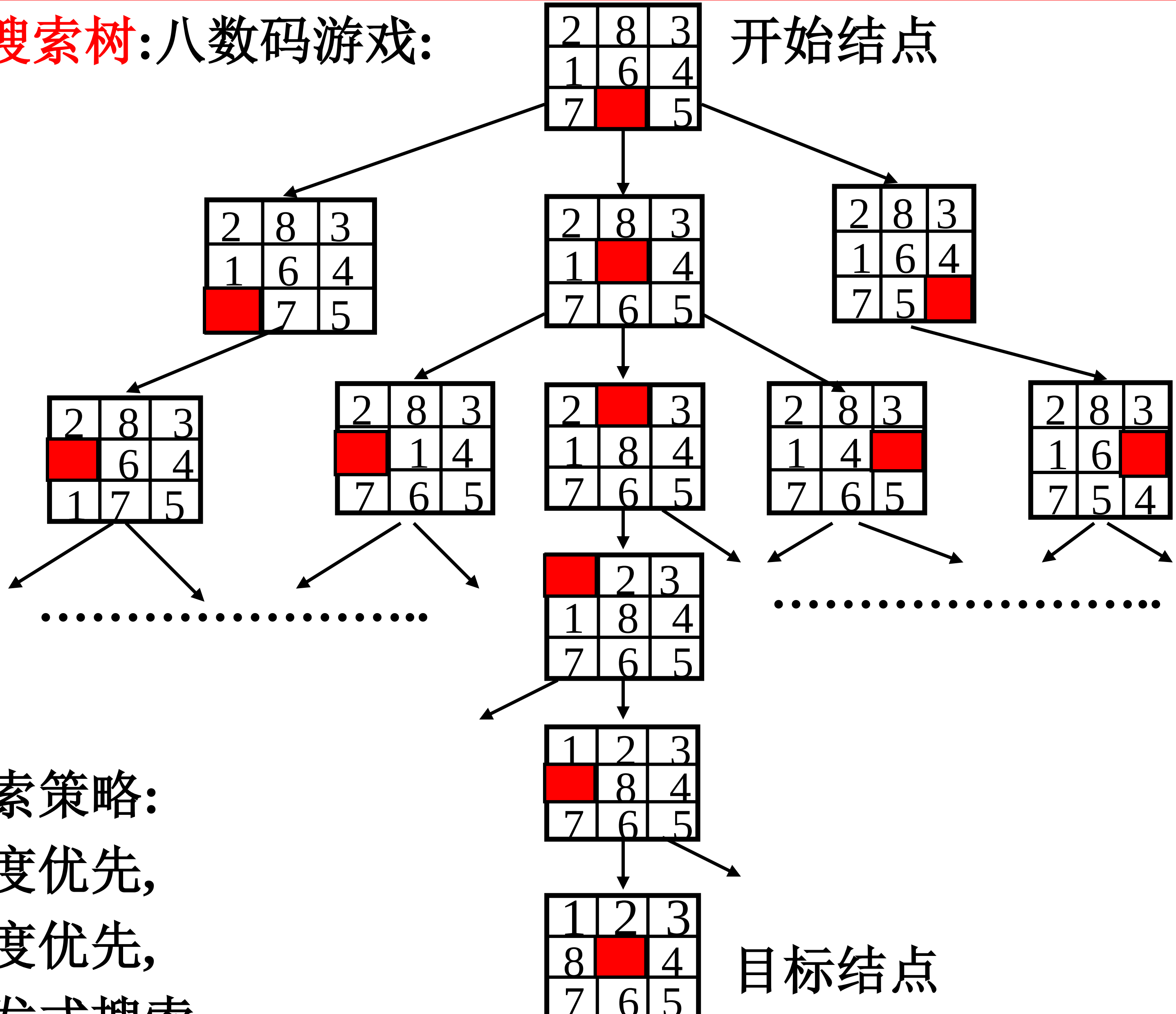
$((a+b)\div c)\times(d-e)$



c)判定树:有四枚金币a,b,c,d,已知道三个是真的,最多一个是假的,它们的外表完全相同,只是重量有点差别.给你一架天平找出假币.



d)搜索树:八数码游戏:



搜索策略:

宽度优先,

深度优先,

启发式搜索,….

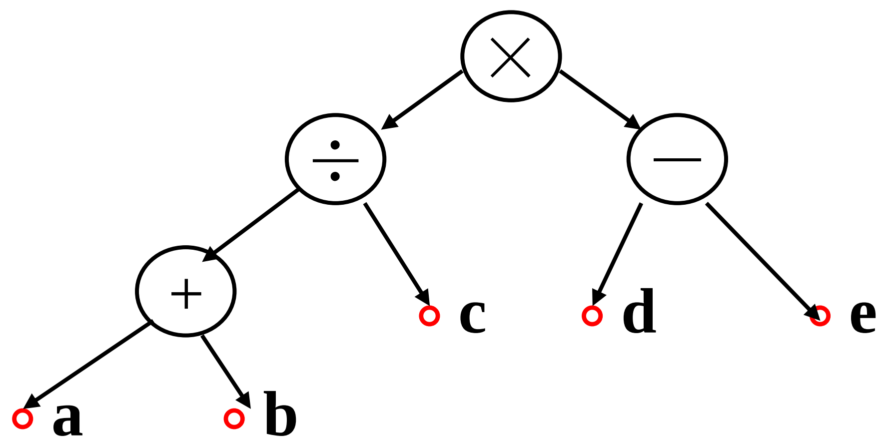
四.有序树

如前面的算术表达式树, 家谱树,都是有序树, 即同一层的结点是有次序的, 如家谱树,最左边是老大,其次是老二,依此类推.

定义:在有向树中,如果规定了每一层上的结点的次序,称之为有序树.

算术表达式树:

((a+b)÷c)×(d−e)

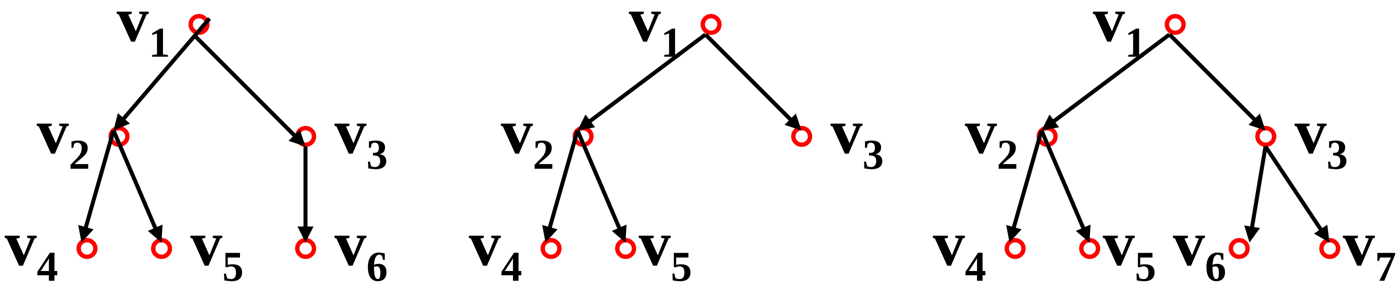


五.m叉树与完全m叉树

1.m叉树:在根树中,如果每个结点的出度最大是m, 则称此树是m叉树.

2.完全m叉树:在根树中,如果每个结点的出度都是m或者等于0, 则称此树是完全m叉树.

3. 正则m叉树:在完全m叉树中,如果所有树叶的层次相同,则称之为正则m叉树.



定理8-10.1 T是棵完全m叉树, 有t个叶结点, i个分支结点,

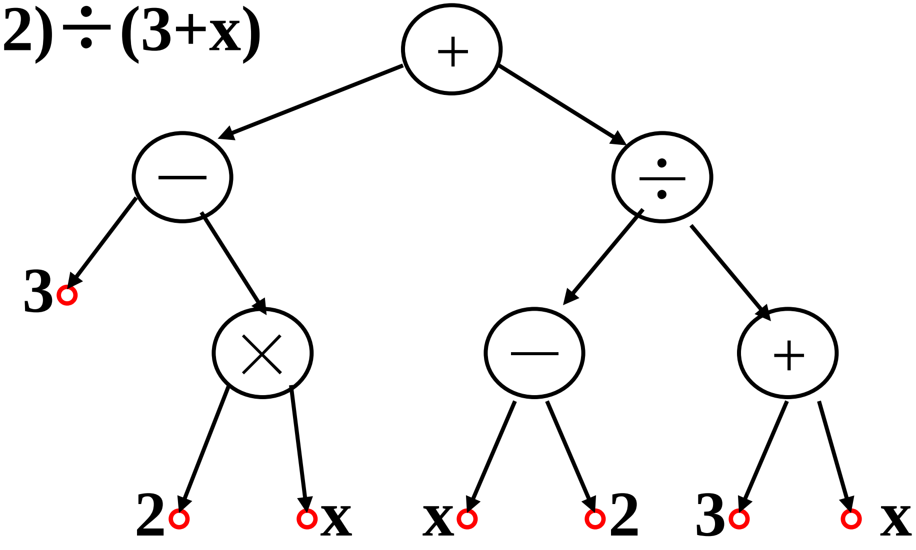
则$(m-1)i=t-1$.

证明:T的所有结点的出度总和为 mi. 入度总和$(i-1)+t$. 故 $mi=i-1+t$ 所以$(m-1)i=t-1$

六. 二叉树的存贮

二叉树便于在计算机内存贮, 设有算术表达式;

(3－(2×x))+((x-2)÷(3+x))



存贮时,每个结点含有三个信息:

left	data	right

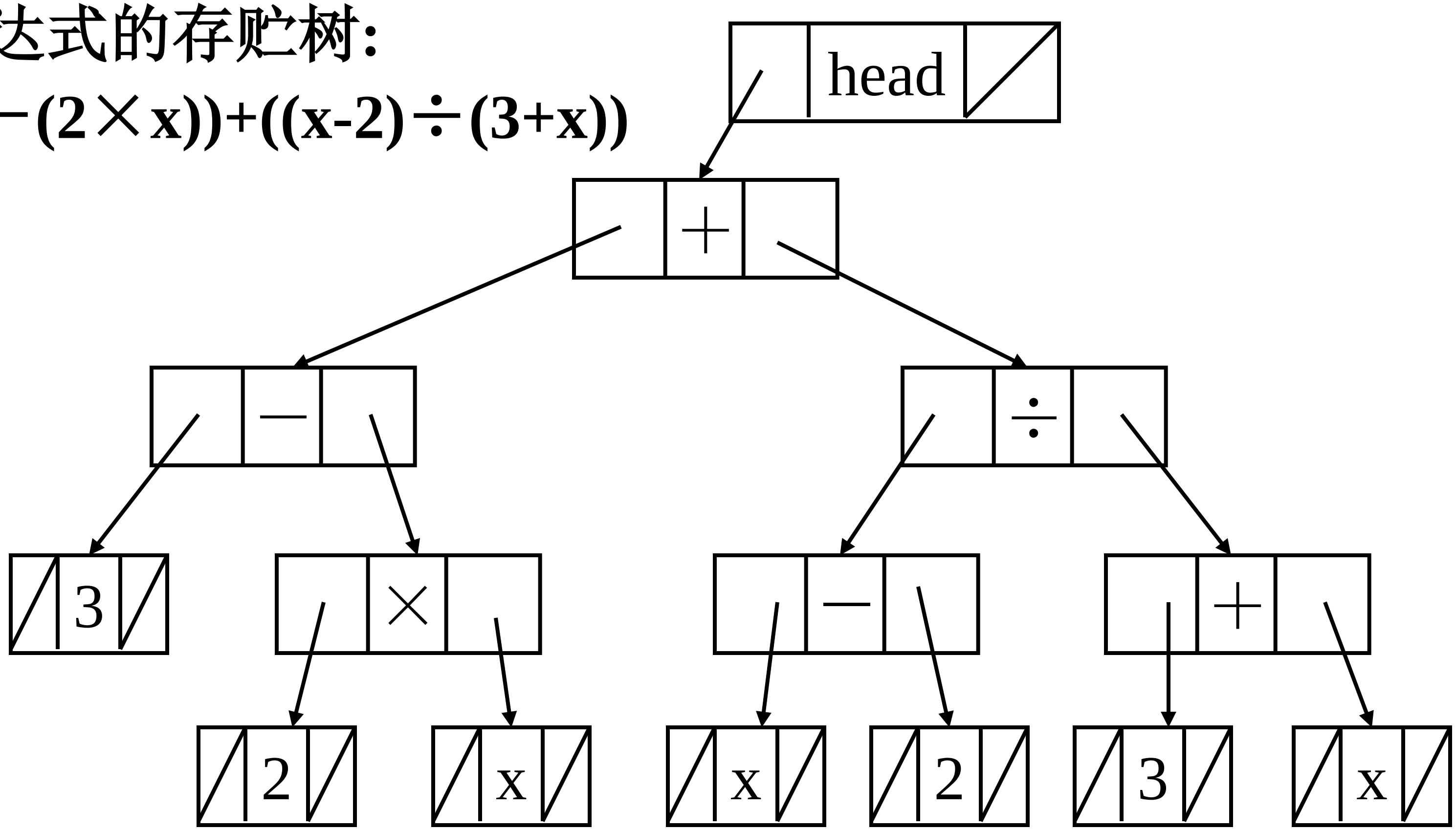
left-----是左指针,指向左子结点.

data----数据

right---右指针, 指向右子结点.

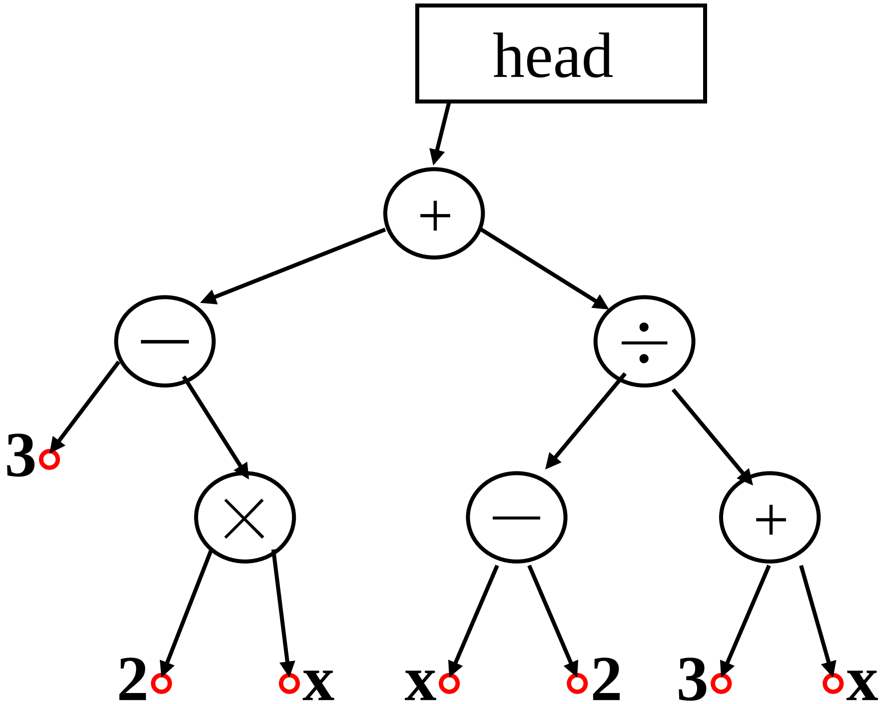
表达式的存贮树:

(3－(2×x))+((x-2)÷(3+x))



如果使用矩阵表示此树,需要13×13的矩阵, 需要169单元存贮空间,而且矩阵中有很多0. 显然冗余太多.

我们用三个一维数组构成的序列表示这棵树:



index i	left(i)	data(i)	right(i)
1	2	head	0
2	3	+	8
3	4	—	5
4	0	3	0
5	6	×	7
6	0	2	0
7	0	x	0
8	9	÷	12
9	10	—	11
10	0	x	0
11	0	2	0
12	13	+	14
13	0	3	0
14	0	x	0

0 表示无左(右)子结点

只用了42个存贮单元,

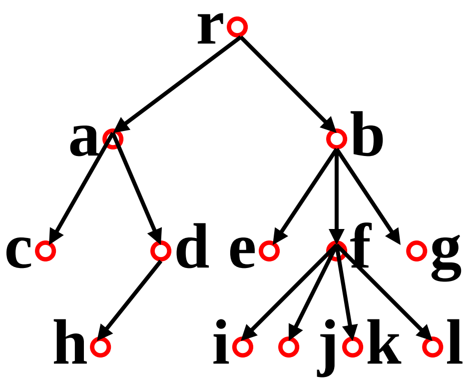
可见节省内存.

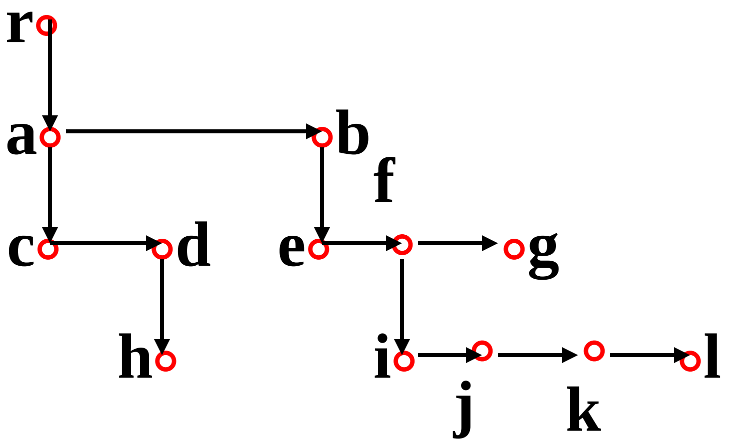
七. m叉有序树转化成二叉树

因为二叉树便于存贮, 也便于处理, 所以通常可以将多叉树化成二叉树。方法是:

1.每个结点保留左儿子结点, 剪掉右边其分支，被剪掉的结点如下处理(重新嫁接)。

2.同一个层次的结点, 从左到右依次画出(被剪掉的结点嫁接到它的哥哥结点上)。





八.遍历二叉树

在二叉树的一些应用中, 常常要在树中查找具有某些特征的结点,或者对所有结点逐一进行某种处理, 这就提出了遍历二叉树问题. 即按照一定规律巡访树中每个结点一次.

由于二叉树是一个非线性结构, 每个结点都可能在左右两棵子树上, 为此要寻找一种规律, 以便使二叉树上结点的信息排成一个线性队列上, 从而便于遍历.

有三种遍历方式

1.先序遍历

2.中序遍历

3.后序遍历

1.先序遍历

(1)访问根结点.

(2)先序遍历左子树

(3)先序遍历右子树

结果:＋－3×2－x2÷x＋3x

2.中序遍历

(1) 中序遍历左子树

(2)访问根结点.

(3)中序遍历右子树

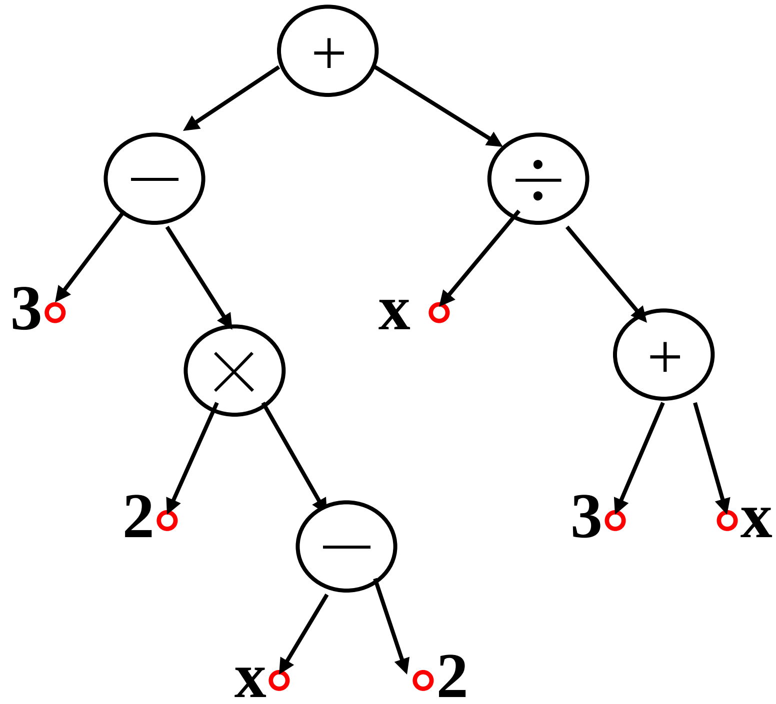
结果:3－2×x－2＋x÷3＋x

3.后序遍历

(1)后序遍历左子树

(2)后序遍历右子树

(3)访问根结点.

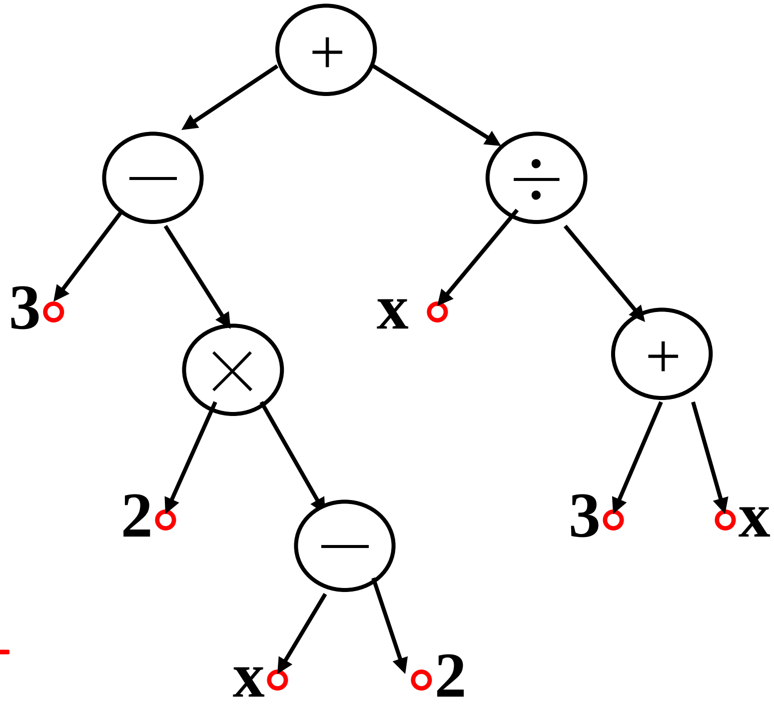


后序遍历:32x2－×－x3x＋÷＋

后序遍历的应用：

源程序经过编译之后，算术表达式都变成后序形式(逆波兰表达式)，在计算时，不必考虑运算的优先权，扫描表达式时，遇到运算符号，就将其前面的两个运算数做此运算即可。例如，前面的例子

(3－(2×(x－2)))+(x÷(3+x))



后序遍历:

32x2－×－x3x＋÷＋

32(x－2)×－x3x＋÷＋

3(2×(x－2)) －x3x＋÷＋

(3－(2×(x－2)))x3x＋÷＋

(3－(2×(x－2)))x(3＋x)÷＋

(3－(2×(x－2)))(x÷(3＋x))＋

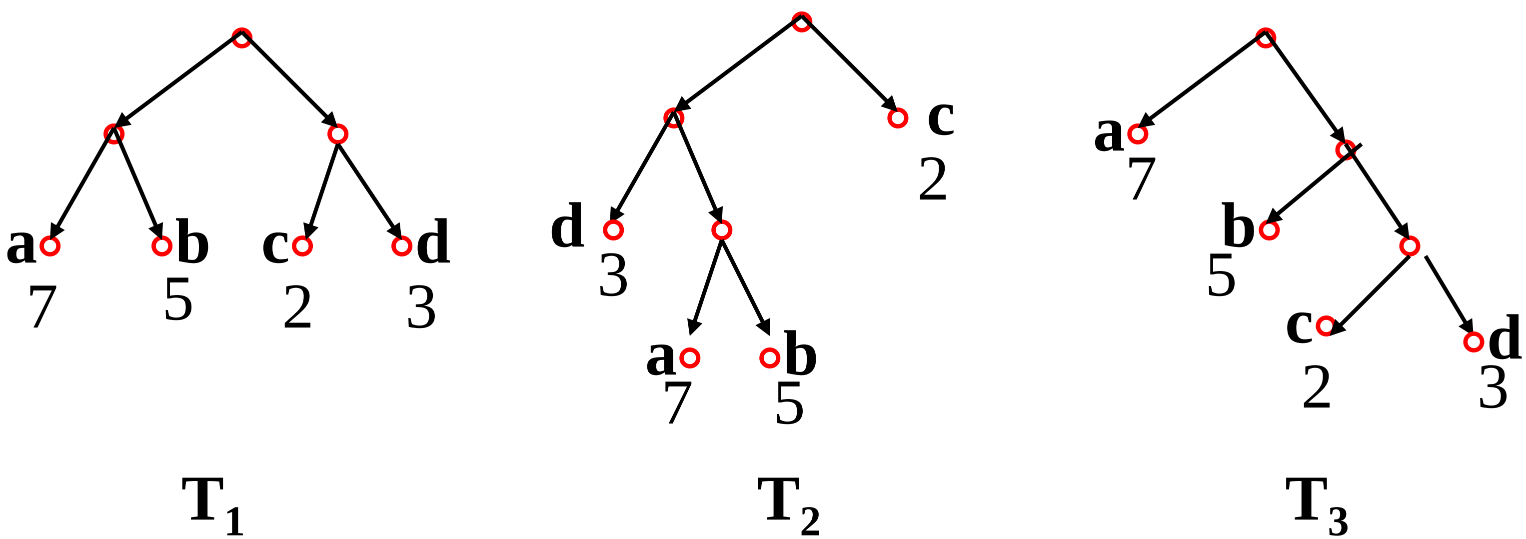
(3－(2×(x－2)))＋(x÷(3＋x))

九. 最优树(哈夫曼树 Huffman)

二叉树的一个重要应用就是最优树.

1.带权二叉树的定义:设有一组权值:$w_1, w_2, w_3, \ldots, w_m$,不仿设$w_1 \le w_2 \le w_3 \le \ldots \le w_m$, 设有一棵二叉树有m片叶子,分别带有权值$w_1, w_2, w_3, \ldots, w_m$,称此树为带权二叉树.

例如:下边是有叶结点a,b,c,d, 分别带有权7,5,2,3的二叉树:



2. 带权树T的权W(T):

$$W(T)=\sum_{i=1}^{m} w_i L(w_i)$$

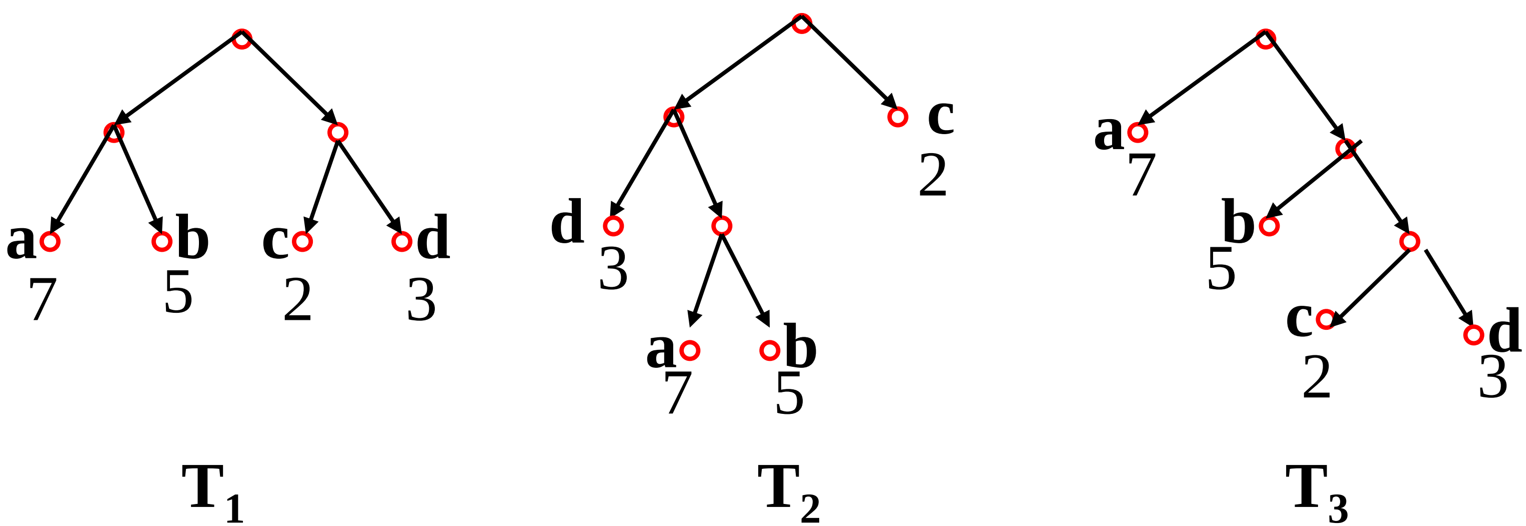
其中$L(w_i)$是标有权w_i的叶结点的从根到该叶结点的路长.

上例中: $W(T_1)=7\times2+5\times2+2\times2+3\times2=34$

$W(T_2)=3\times2+7\times3+5\times3+2\times1=44$

$W(T_3)=7\times1+5\times2+2\times3+3\times3=32$

由此看出$W(T_3)$是比较小的.



3.最优树: 带权树中,权数最少的二叉树.

4.画最优树的算法---哈夫曼算法:

(1)先将权按照升序排序,设为$w_1 \le w_2 \le w_3 \le \ldots \le w_m$.

(2)以w_1和w_2为儿子结点, 构造它们的父结点,且其权为 w_1+w_2，并从权的序列中去掉w_1和w_2。

(3) w_1+w_2再与其余权一起排序, 再从此队列中取出前面两个权值为儿子结点, 同(2)的方法构造它们的父结点.

依此类推, 直至最后. 即得到最优树.

例如,给定一组权:2,3,5,7,11,13,17,19,23. 构造一棵最优树.

100

42,58

24,34,42

19,23,24,34

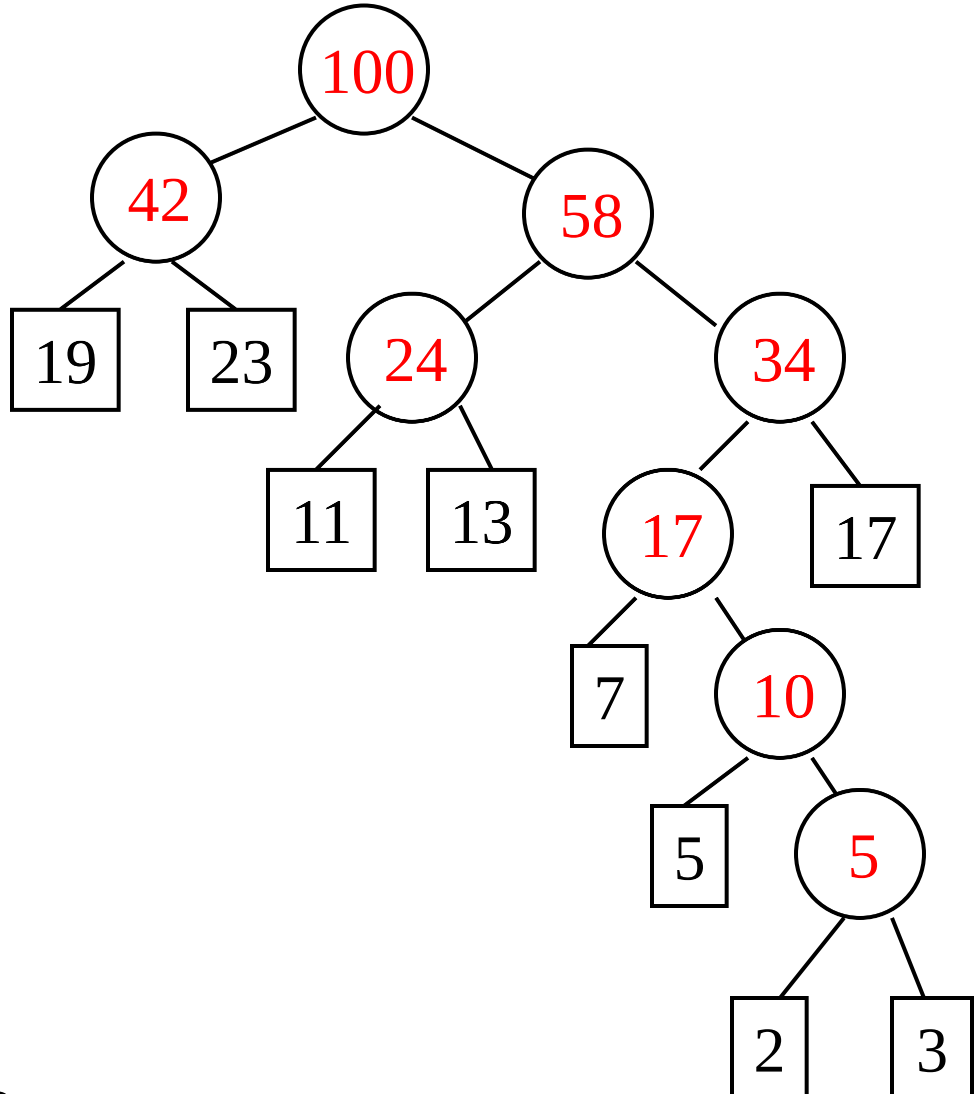
17,17,19,23,24

11,13,17,17,19,23

7,10,11,13,17,19,23

5,5,7,11,13,17,19,23

2,3,5,7,11,13,17,19,23

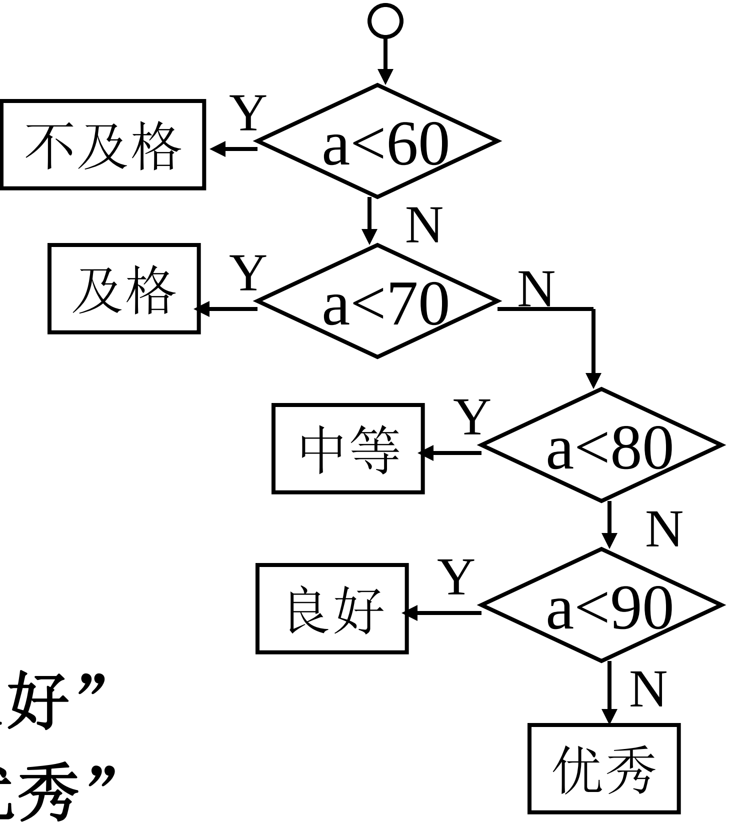


5. 最优树的应用举例

(1)用于程序设计

例如编写一个将百计分a转换成五计分的程序,如果这样:

```
if a<60
    then  b="不及格"
    else  if a<70
        then  b="及格"
        else   if  a<80
            then  b="中等"
            else  if  a<90
                then b="良好"
                else   b="优秀"
```



上述程序是正确的,但不是最优的,.

衡量一个程序是否优化:

a)空间复杂性:一个是看它在运行时需要使用的存贮空间的大小,

b)时间复杂性:还要看它运行时间的长短.

显然在分数正态分布情况下,上述程序运行的时间,不是最优的. 设分数的分布如下:

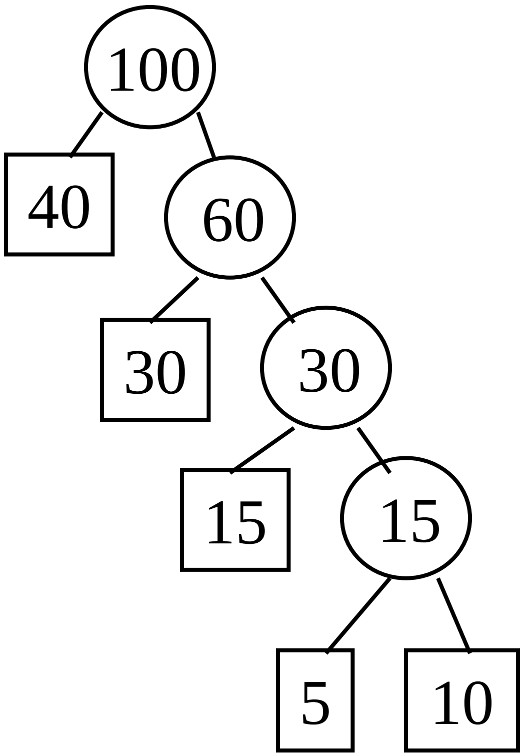
分数:	0-59	60-69	70-79	80-89	90-100
比例(%):	5	15	40	30	10

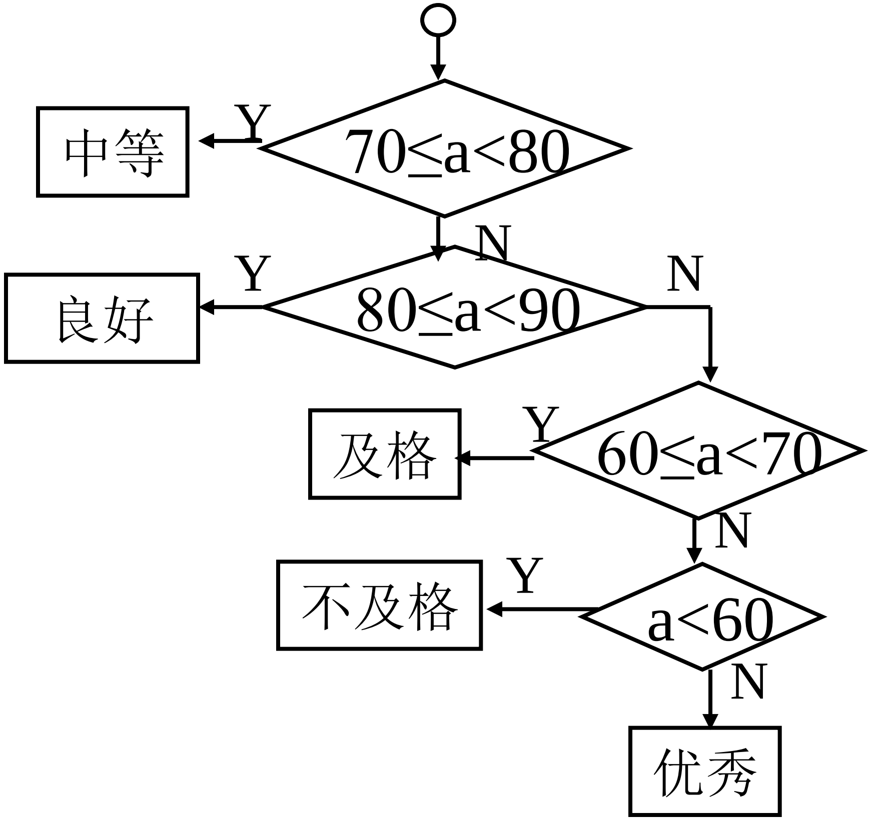
可见在分数正态分布的情况下,上述程序中,有80%的分数,至少要比较3次(因为 80%的分数>70分)才得出结果.

那么如何设计这个程序才合理呢?就是按最优树来设计.

分数:	0-59	60-69	70-79	80-89	90-100
比例(%):	5	15	40	30	10

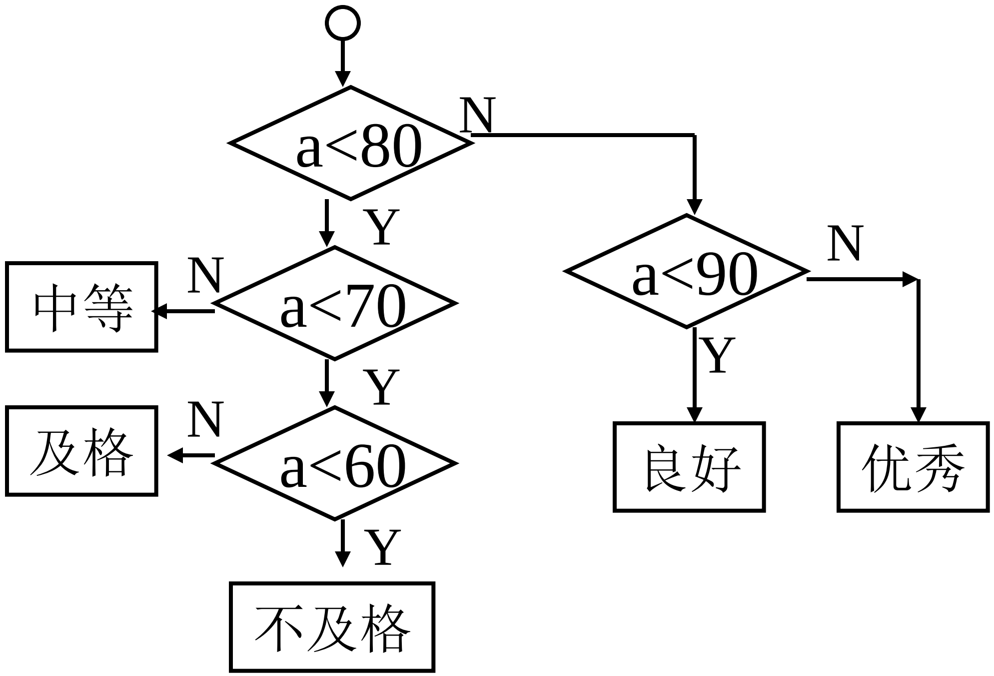
设权序列为: 5,10,15,30,40　　构造最优树:





为了使得判断框内只比较一次,流程图可以改成下面框图:

在分数正态分布下,按照这个流程图,编制程序,比较合理.



(2)前缀码(哈夫曼编码)

a) 问题的提出: 数据通讯时,需要将信息编码, 即用二进制符号串表示信息.

例如要传送的报文为:“**ABACCDA**” , 只有**4**个基本符号,

只要二位二进制符号就可以分辨. 设A,B,C,D的编码分别是“00,01,10,11”. 这样上述报文“ABACCDA”可翻译成“00010010101100”, 译文含有14个符号.

这种编码各个符号的编码是等长等. 当然这样的编码在报文的接收端容易译码.

但是在发送报文时,总是希望报文最短, 节约开支. 所以等长的编码不是最优的, 因为在报文中各个符号出现的频率是不同的. 所以考虑用不等长的编码, 应该使得在报文中出现频率最高的符号编码最短.

比如A,B,C,D的编码分别为: 0,00,1,01. 这样此报文的译文为:“000011010”, 译文的长度是短了,只有9个符号, 但是在报文的接收端如何翻译成原文呢?比如译文中的“0000”是“AAAA”还是“BB”,还是“”ABA”, 无法翻译.

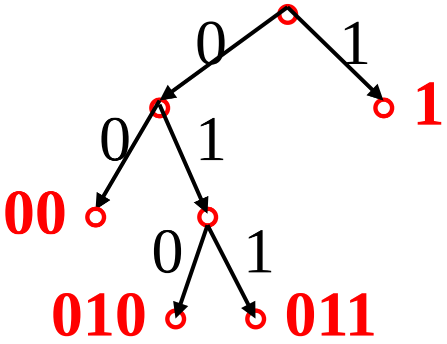
产生这个问题的原因是:有的符号的编码是另一个符号编码的前缀. 比如A的编码“0”,是B编码“00”的前缀.这样就无法翻译报文.

直接促使我们设计前缀码.

b)前缀码的定义:一个符号的编码不是另一个符号编码的前缀.

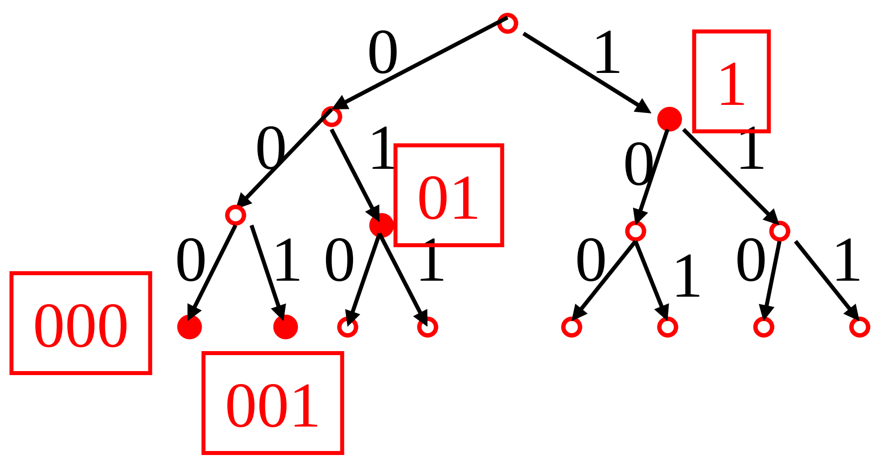
c)前缀码的设计:

每棵二叉树对应一组前缀码. 在二叉树的边上,将每个结点下面的两条边分别标上0和1, 然后从根到叶,把这个路径的边上所标的0-1符号串写下来,就是这个叶结点对应的前缀码.



反之,任何一组前缀码也对应一棵二叉树.

例如有一组前缀码{1,01,001,000},因为其中最长的编码为000,有3位,可以画一棵高度为3的二叉树,如图:



现在再回到前面的问题,对报文“ABACCDA”设计编码.

先计算各个符号在报文中出现的频率:A,B,C,D的频率分别是:43%,14%,29%,14% . 分别表示成权(43,14,29,14)

对权排序:**14,14,29,43**

画最优树:

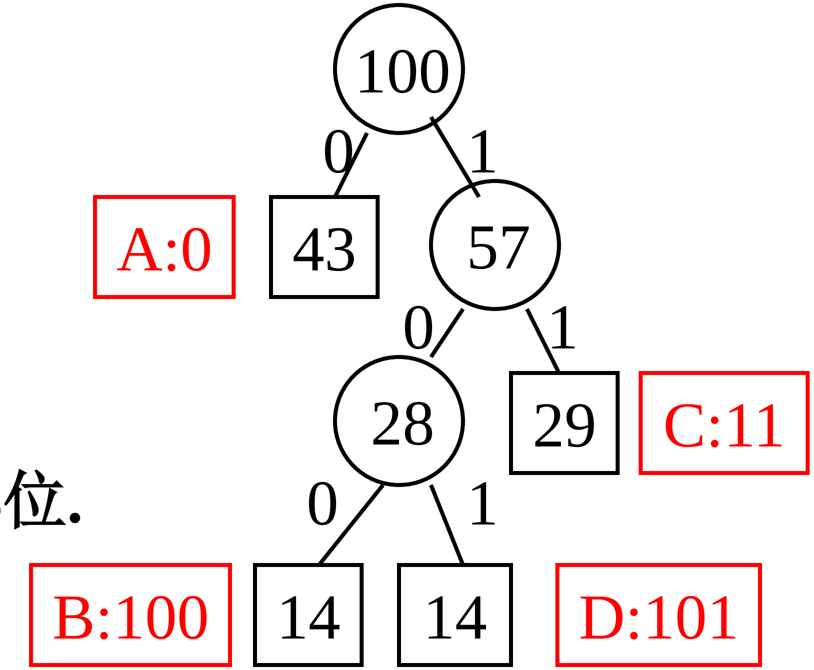
编码是: **A, B, C, D**

0, 100, 11, 101.

原报文 "**ABACCDA**"译文:

"**0 1 0 0 0 1 1 1 1 1 0 1 0**"有**13**位.

A **B** A C C D A



那么在接收端如何译码呢?

就是将译文中符号从头读,

顺着这棵最优树,反复从根到叶找前缀码.

比如有如下报文 "**01110110111100111100100**"

原文是 "**AC D D C B C CAAB** "

作业 P337 (1), (2), (3), (5)a) b) , (6), (8)

图　论　小　结

本章内容特点:

概念很多, 而且有些概念很相近,容易弄错, 要特别仔细. 另外本章内容容易理解, 也不那么抽象.

下面再将内容归纳一下:

一.图的概念

图的定义, 有向边,无向边,平行边,环

邻接点,邻接边, 孤立结点

有向图, 无向图,简单图,混合图,零图,平凡图,多重图,

完全图,子图, 生成子图,补图,

结点的度, 结点的出度, 结点的入度,

图的最大度$\Delta(G)$,最小度$\delta(G)$,

深入了解图所有结点度数总和与边的关系, 出度和与入度和关系

掌握图的同构的概念，并会判定同构的图。

二.路与回路

路,回路,迹,闭迹,通路,圈

无向图的连通性:连通图,连通分支,连通分支数$W(G)$,

点割集,割点,点连通度$k(G)$,

边割集,割边(桥),边连通度$\lambda(G)$

结点间的距离, 图的直径

有向图的连通性:可达性,

强连通,单侧连通,弱连通, 强分图,单侧分图,弱分图.(会求这些分图)

三.图的矩阵

邻接矩阵A:结点与结点之间的邻接关系矩阵.

根据邻接矩阵判断:各结点的度, 有向图结点出,入度.

由A^k可以求一个结点到另一个结点长度为k的路条数.

可达矩阵P:结点u到结点v的可达性的矩阵.

用P可以判定:有向图的强分图.

关联矩阵M:是结点与边的关联关系矩阵.

用M判定:各结点的度

四.欧拉图与汉密尔顿图(会判定)

欧拉路,欧拉回路,欧拉图.

判定:有欧拉路的充要条件:无或有两个奇数度的结点.

有欧拉回路的充要条件:所有结点度数均为偶数.

汉密尔顿路,汉密尔顿回路,汉密尔顿图

汉密尔顿图的判定:

必要条件:V的任何非空子集S,有$W(G-S)\leq|S|$

充分条件:每对结点的度数和$\geq|V| =n$

*五. 二部图 作为一般了解，但要掌握K3,3 。

六. 平面图

平面图的定义，平面的边界,

欧拉公式: $v-e+r=2$

判定:

必要条件: $e\leq 3v-6$

充要条件:G不含与K5或K3,3在2度结点内同构子图.

七. 对偶图与着色

会画对偶图, 会对图正常着色。

八.树与生成树

树的定义:6个定义,其中最主要的是连通无回路, e=v-1

分支结点, 叶结点

会求最小生成树

九. 根树

m叉树,完全 m叉树, (m-1)i=t-1

m叉树变成二叉树

会画最优树,

会设计前缀码