离 散 数 学

电子科技大学

计算机科学与工程(网络空间安全)学院

2024年5月23日星期四

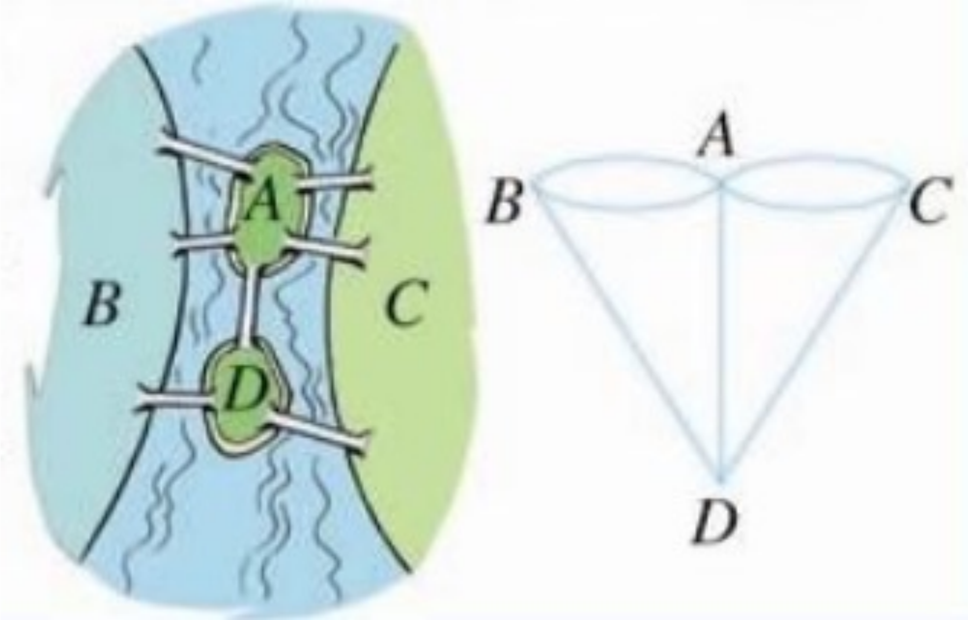
第四篇 图论

图论是一门很有实用价值的学科，它在自然科学、社会科学等各领域均有很多应用。自上世纪中叶以来，它受计算机科学蓬勃发展的刺激，发展极其迅速，应用范围不断拓广，已渗透到诸如语言学、逻辑学、物理学、化学、电讯工程、计算机科学以及数学的其它分支中。特别在计算机科学中，如形式语言、数据结构、分布式系统、操作系统等方面均扮演着重要的角色。

引言

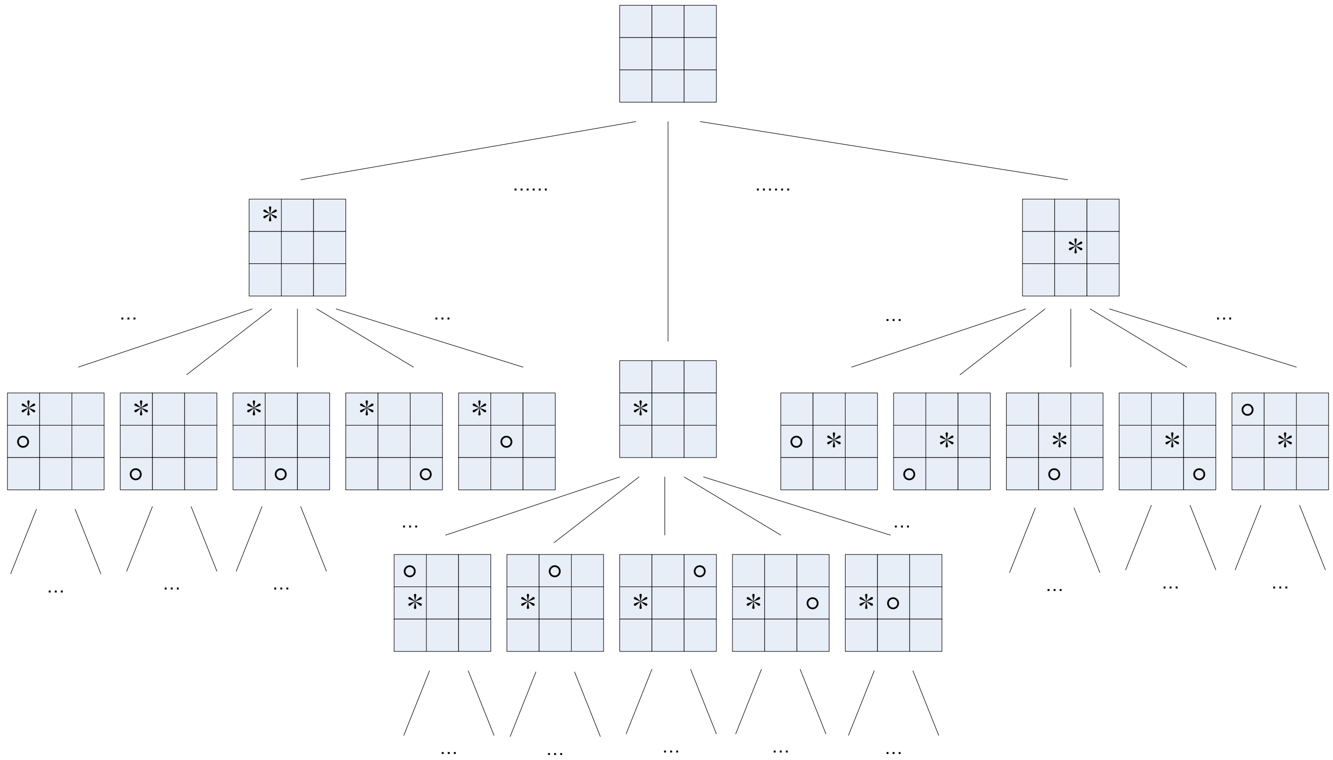
1. 七桥问题 欧拉
2. 游戏、博弈问题
3. 克希荷夫定律
4. 树 凯莱
5. 四色猜想

哥尼斯堡七桥问题和欧拉



博弈树问题

假设甲乙双方在进行二人棋盘游戏，从初始局面开始，甲乙两方交替走棋，局面的变化可以表示成一个树形结构，这就是博弈树（game-tree）。一种井字棋的博弈树，如图所示。



另一种博弈问题-囚徒的困境

- 1950年，由就职于兰德公司的梅里尔·弗勒德（Merrill Flood）和梅尔文·德雷希尔（Melvin Dresher）拟定出相关困境的理论，后来由顾问艾伯特·塔克（Albert Tucker）以囚徒方式阐述，并命名为“囚徒困境”。
- 警方逮捕甲、乙两名嫌疑犯，但没有足够证据指控二人入罪。于是警方分开囚禁嫌疑犯，分别和二人见面，并向双方提供以下相同的选择：

 * 若一人认罪并作证检控对方，而对方保持沉默，此人将即时获释，沉默者将判监10年。

 * 若二人都保持沉默，则二人同样判监半年。

 * 若二人都互相检举，则二人同样判监2年。

电子科技大学

四色猜想

俄 罗 斯
U S S I A
哈萨克斯坦
KAZAKHSTAN
蒙 古
MONGOLIA
中华人民共和国
PEOPLE'S REPUBLIC
OF CHINA
朝鲜
D.P.R.KOREA
韩国
R.O.KOREA
日本
JAPAN
萨哈林岛
(库页岛)
千岛群岛
阿富汗
AFGHANISTAN
巴基斯坦
PAKISTAN
印 度
INDIA
缅甸
MYANMAR
泰国
THAILAND
鄂霍次克海
黄海
东海
南海
北马里亚纳
关岛(美)

教学目标

图是一类具有广泛实际问题背景的数学模型，有着极其丰富的内容，是数据结构等课程的先修内容。学习时应掌握好图论的基本概念、基本方法和基本算法，善于把实际问题抽象为图论的问题，然后用图论的方法去解决。

图论作为一个数学分支，有一套完整的体系和广泛的内容，本篇仅介绍图论的初步知识，其目的在于今后对计算机有关学科的学习和研究时，可以以图论的基本知识作为工具。

第9章 图

我们所讨论的图(Graph)与人们通常所熟悉的图，例如圆、椭圆、函数图表等是很不相同的。图论中所谓的图是指某类具体离散事物集合和该集合中的每对事物间以某种方式相联系的数学模型。如果我们用点表示具体事物，用连线表示一对具体事物之间的联系。那么，一个图就是由一个表示具体事物的点的集合和表示事物之间联系的一些线的集合所构成，至于点的位置和连线的长短曲直是无关紧要的。

9.0 内容提要

1. 图的基本概念
2. 图的表示、分类
3. 图的性质
4. 通路与回路
5. 图的连通性
6. 图的应用

9.1 本章学习要求

重点掌握 一般掌握 了解

1

1. 图的概念
2. 特殊图
3. 图论的基本定理
4. 通路与回路
5. 图的连通性

2

1. 图的同构
2. 图的构成与证明

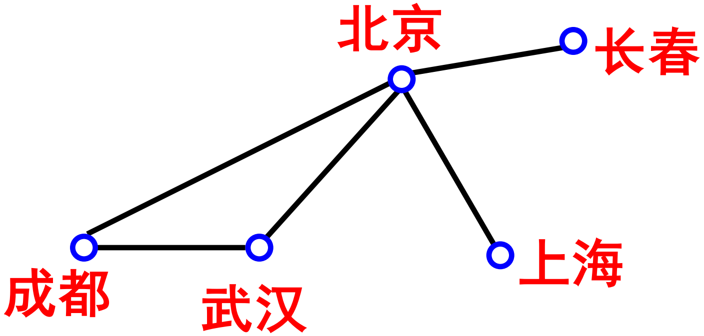
3

图论中的应用

9.2 图的基本概念

9.2.1 图的定义

例9.2.1（1）考虑一张航线地图，图中用点表示城市，当两个城市间有直达航班时，就用一条线将相应的点连接起来。这种航线地图的一部分如下图所示；



例9.2.1（2）


板凳
楼主主题帖，或回复主题帖
回复
占位来的..
[S002]Lydie-金融-
看看大家的认真回复，你好意思在这挂个占位么
回复，所有的回复都是平行的
楼中楼也是平行关系
[S347]树先生-建...
很优秀
[S3]Chiang-CM-佛山... 回复 [S002]Lydie-金融-... (2016-01-06 11:54:31)
[S3]Chiang-CM-佛山... 回复 [S3]Chiang-CM-佛山... (2016-01-06 11:54:49)
【竟然只能240字....】
[S002]Lydie-金融-... 回复 [S3]Chiang-CM-佛山... (2016-01-06 20:46:41)
上面那段最后一句是在翻译SWOT么哈哈
收起回复

例9.2.1（3）

```
▶<div id="post_6">…</div>
▼<div id="post_9">
  ▼<table id="pid9" summary="pid9" cellspacing="0"
  cellpadding="0">
    ▼<tbody>
      ▼<tr>
        ▶<td class="pls">…</td>
        ▼<td id="plc_9" class="plc postContent">
          ▼<div class="post_top">
            ▶<div class="topt fright">…</div>
            ▼<p class="postinfo">
                <span class="postnum">板凳</span>
                <span class="gn rn" title="[S3]Chiang-CM-佛山、">
                [S3]Chiang-CM-佛山、</span> == $0
                <span class="posttime" title="2016-01-01 01:40:
                46">2016-01-01 01:40:46</span>
              </p>
            </div>
          ▼<div class="pct">
            ▶<div class="pctmessage mbm xs2" style>…</div>
              <!--楼中楼-->
            ▼<div id="replyFloorList_9" class="replaylist">
              ▶<div id="floorPre_9" style="display: none;">
              …</div>
              ▼<div id="floorList_9" style>
                ▼<div class="replayitem cl">
                  ▶<span>…</span>
                  ▼<div class="rpbox">
... div div #postlist div #post_9 #pid9 tbody tr #plc_9 div p span.gn.rn
```

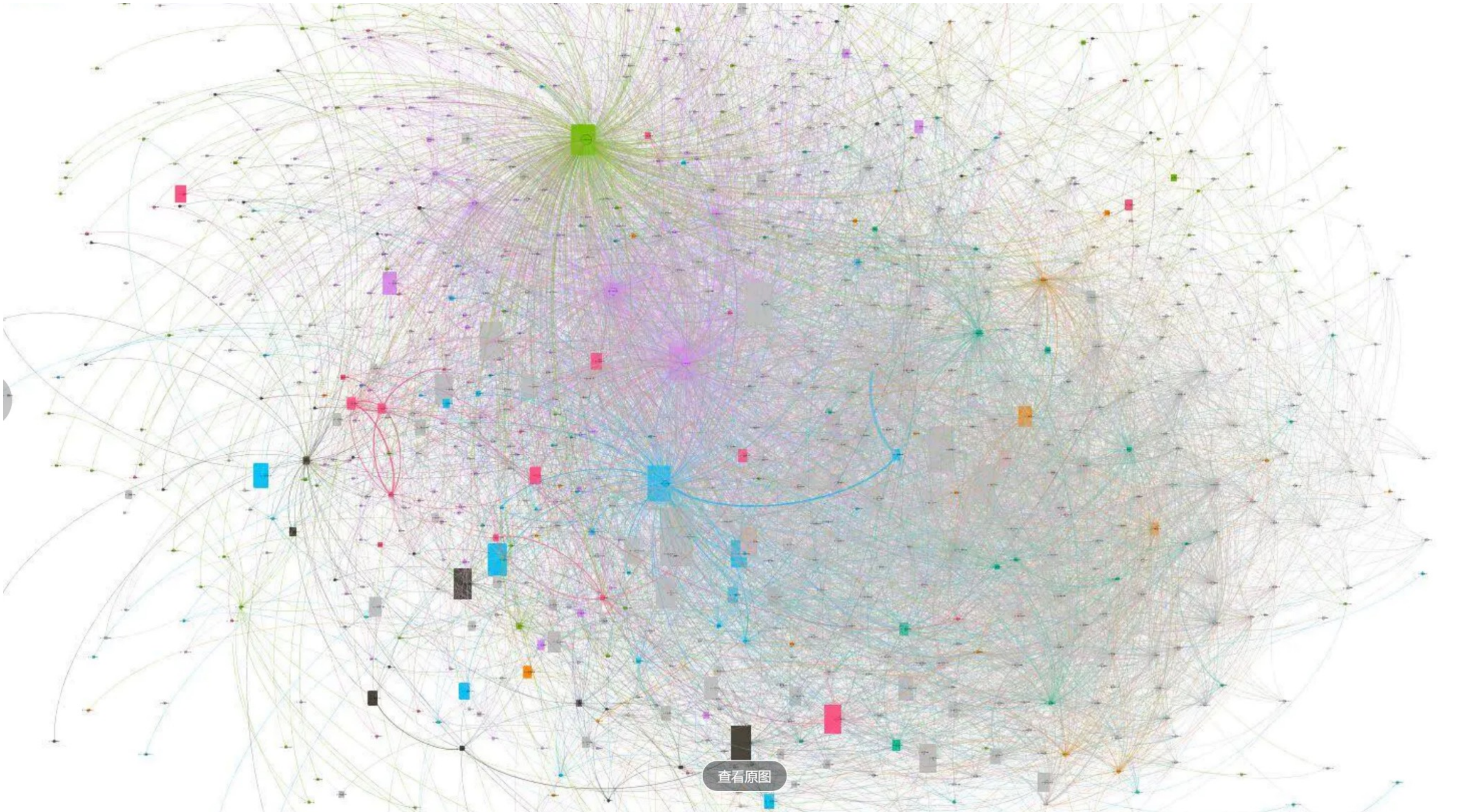
他人主题回复

```python
item = LinkItem()

# 他人的发贴
links = selector.xpath('//span[@class="gn "]/@title').extract()
for link in links:
    item['bname']=link
    gid = re.findall(r'\d+', str(link))[0]
    item['aid'] = sid
    item['bid'] = gid

    yield item

# 他人的发贴
links = selector.xpath('//span[@class="gn rn"]/@title').extract()
for link in links:
    item['bname'] = link
    gid = re.findall(r'\d+', str(link))[0]
    item['aid'] = sid
    item['bid'] = gid

    yield item

# 他人的回复
links = selector.xpath('//div[@class="postinfo mbn ptn"]/b/@title').extract

for link in links:
    item['bname'] = link
```

```
<span class="gn rn" title="[S3]Chiang-CM-佛山、
[S3]Chiang-CM-佛山、</span>
<span class="posttime" title="2016-01-01 01:40
46">2016-01-01 01:40:46</span>
</p>
</div>
▼<div class="pct">
  ▶<div class="pctmessage mbm xs2" style>…</div>
  <!--楼中楼-->
  ▼<div id="replyFloorList_9" class="replaylist">
    ▶<div id="floorPre_9" style="display: none;">
    …</div>
    ▼<div id="floorList 9" stvle>
```

```sql
1  select aid as source,bid as target,count(bid) as weight  from bbs.link  group by aid,bi
```

	A	B	C
8328	[S801]mai-通信-北京	[S756]ammy-医疗卫生-	1
8329	[S801]mai-通信-北京	[S801]mai-通信-北京	55
8330	[S801]mai-通信-北京	[S814] babyer-工管-石	1
8331	[S801]mai-通信-北京	[S816]玲珑剔透-学生-	2
8332	[S803]Emily-外贸-深圳	[S756]ammy-医疗卫生-	1
8333	[S803]Emily-外贸-深圳	[S803]Emily-外贸-深圳	1
8334	[S804] 追梦-英语翻译	[S756]ammy-医疗卫生-	1
8335	[S804] 追梦-英语翻译	[S804] 追梦-英语翻译	2
8336	[S805]Cathy-教育-深圳	[704]Summers-留学/葡	1
8337	[S805]Cathy-教育-深圳	[S756]ammy-医疗卫生-	1
8338	[S805]Cathy-教育-深圳	[S805]Cathy-教育-深圳	1
8339	[S806]乐默笙箫-教师-D	[S756]ammy-医疗卫生-	1
8340	[S806]乐默笙箫-教师-D	[S806]乐默笙箫-教师-D	6
8341	[S807]我是胖猴子-教育	[S300]-贝姑娘-签证-广	1
8342	[S807]我是胖猴子-教育	[S756]ammy-医疗卫生-	1
8343	[S807]我是胖猴子-教育	[S807]我是胖猴子-教育	1
8344	[S808]黄小样-服装辅料	[S756]ammy-医疗卫生-	1


查看原图

基本思想

用图形表示一组对象，其中有些对象对是有联系的。当然，这几个图形也可以表示其它的含义。 对于这种图形，我们感兴趣的只是有多少个点和哪些结点之间有线连接，至于连线的长短曲直和结点的位置却无关紧要，只要求每一条线都起始于一个点，而终止于另一个点。

定义9.2.1

一个图(Graph)是一个序偶<V, E>，记为G = <V, E>，其中：

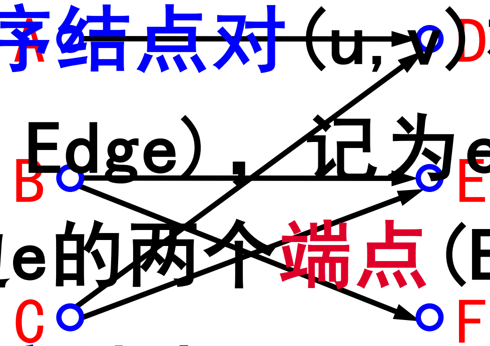
（1）V = $\{v_1, v_2, \cdots, v_n\}$是有限非空集合，$v_i$称为结点(Nodal Point)，简称点(Point)，V称为结点集(Nodal Set)。

（2）E是有限集合，称为边集(Frontier Set)。E中的每个元素都有V中的结点对与之对应，称之为边(Edge)。

与边相关的几个概念

定义9.2.1中的结点对即可以是无序的，也可以是有序的。

若边e与无序结点对(u,v)相对应，则称e为无向边(Undirected Edge)，记为e = (u, v) = (v, u)，这时称u、v是边e的两个端点(End point)。



若边e与有序结点对<u, v>相对应，则称e为有向边(Directed Point)(或弧)，记为e = <u, v>，这时称u为e的始点(Initial Point)(或弧尾)，v为e的终点(terminal Point)(或弧头)，统称为e的端点。

9.2.2 图的表示

对于一个图G，如果将其记为G = <V, E>，并写出V和E的集合表示，这称为图的集合表示。

而为了描述简便起见，在一般情况下，往往只画出它的图形：用小圆圈表示V中的结点，用由u指向v的有向线段或曲线表示有向边<u, v>，无向线段或曲线表示无向边(u, v)，这称为图的图形表示。

例9.2.2

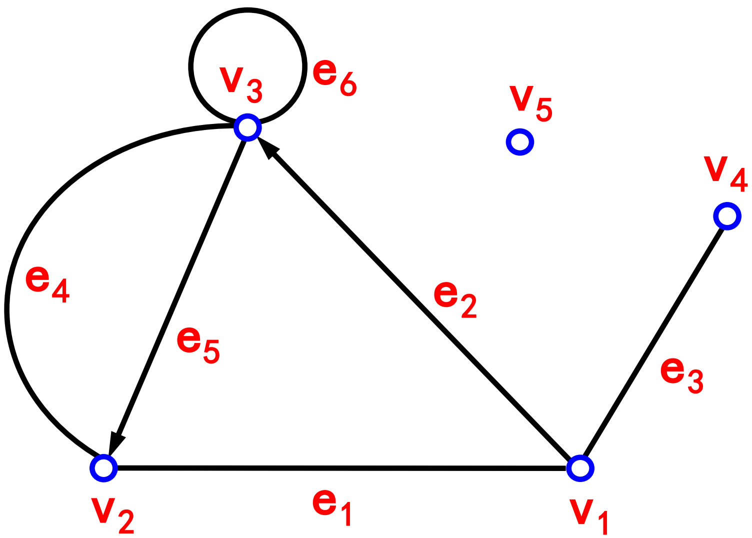
设图G = <V, E>，这里$V = \{v_1, v_2, v_3, v_4, v_5\}$，$E = \{e_1, e_2, e_3, e_4, e_5, e_6\}$，其中$e_1 = (v_1, v_2)$，$e_2 = \langle v_1, v_3\rangle$，$e_3 = (v_1, v_4)$，$e_4 = (v_2, v_3)$，$e_5 = \langle v_3, v_2\rangle$，$e_6 = (v_3, v_3)$。试画出图G的图形，并指出哪些是有向边，哪些是无向边？

例9.2.2 分析

分析　由于V中有5个结点，因此要用5个小圆圈分别表示这5个结点，点的具体摆放位置可随意放。而对E中的6条边，圆括号括起的结点对表示无向边，直接用直线或曲线连接两个端点，尖括号括起的结点对表示有向边，前一个是始点，后一个是终点，用从始点指向终点的又向直线或曲线连接。

例9.2.2 解

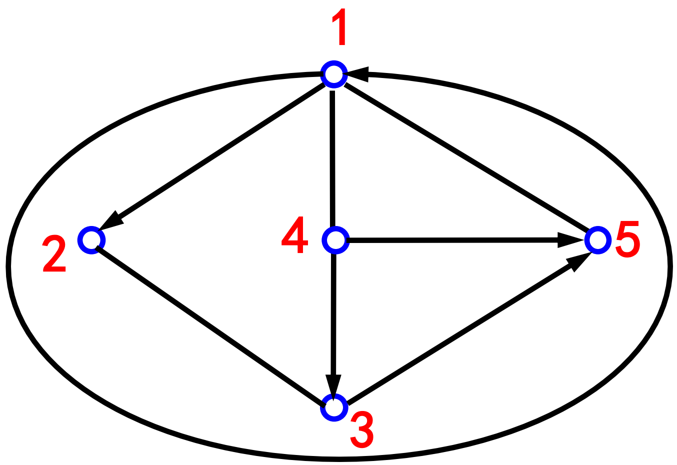
G的图形如下图所示。



G中的e_1、e_3、e_4、e_6是无向边，e_2、e_5是有向边。

例9.2.3

设图G = <V, E>的图形如下图所示，试写出G的集合表示。



图的集合表示为G=<V,E>，V={1, 2, 3, 4, 5}，E={<1,1>, <1,2>, (1,4), (1,5), (2,3), <3,5>, <4,3>, <4,5>}。

解析 图中的小圆圈构成结点集合，将连接结点对的直线或曲线用圆括号括起该结点对表示无向边，将连接结点对的有向直线或曲线用尖括号括起该结点对表示有向边，这里箭头指向的结点放在后面。

两种描述方法的优缺点

- 用集合描述图的优点是精确，但抽象不易理解；
- 用图形表示图的优点是形象直观，但当图中的结点和边的数目较大时，使用这种方法是很不方便的，甚至是不可能的。

图的矩阵表示

我们在学习中常常需要分析图并在图上执行各种过程和算法，也许必须用计算机来执行这些算法，因此必须把图的结点和边传输给计算机，由于集合与图形都不适合计算机处理，所以要找到一种新的表示图的方法，这就是图的矩阵表示。

由于矩阵的行和列有固定的次序，因此在用矩阵表示图时，先要将图的结点进行排序，若不具体说明排序，则默认为书写集合V时结点的顺序。

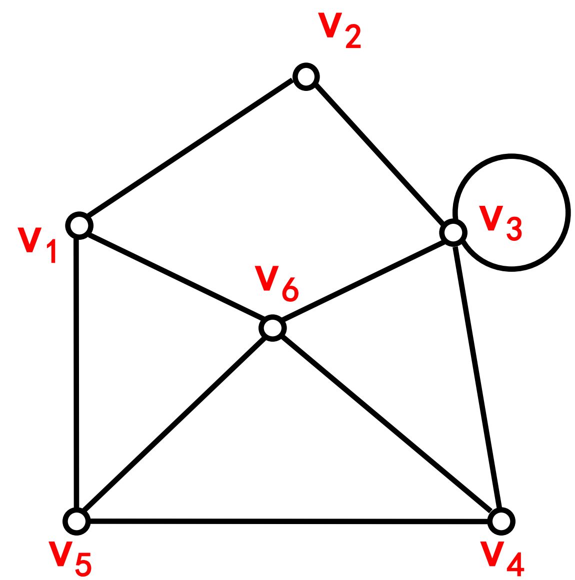
定义9.2.2

设图G = <V, E>，其中V = $\{v_1, v_2, \cdots, v_n\}$，并假定结点已经有了从$v_1$到$v_n$的次序，则n阶方阵$A_G$ = $(a_{ij})_{n\times n}$称为G的邻接矩阵(Adjacency Matrix)，其中

$$a_{ij}=\begin{cases}1, & 若(v_i,v_j)\in E或<v_i,v_j>\in E\\ 0, & 否则\end{cases}\qquad i,j=1,2,3,\cdots,n$$

例9.2.4

试写出下图所示图G的邻接矩阵。



解析 首先将图中的6个结点排序，然后利用定义9.2.2写出其邻接矩阵。

初学时可先在矩阵的行与列前分别按结点排序标上结点，若第i行前的结点到第j列前的结点有边相连，则在邻接矩阵的第i行第j列元素为1，否则为0。若结点排序为$v_1v_2v_3v_4v_5v_6$，则可标记如下：

$$
\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array}
\begin{array}{c}
\begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\
\begin{pmatrix}
0 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 1 & 1 & 0
\end{pmatrix}
\end{array}
$$

说明

由定义9.2.2可看出，图G = <V, E>的邻接矩阵依赖于V中元素的次序。对于V中各元素不同的排序，可得到同一图G的不同邻接矩阵。但是，G的任何一个邻接矩阵可以从G的另一邻接矩阵中通过交换某些行和相应的列而得到，其交换过程与将一个排序中的结点交换位置变为另一个排序是一致的。如果我们略去由结点排序不同而引起的邻接矩阵的不同，则图与邻接矩阵之间是一一对应的。因此，我们略去这种由于V中元素的次序而引起的邻接矩阵的任意性，只选V中元素的任一种次序所得出的邻接矩阵，作为图G的邻接矩阵。

例

图中的结点重排次序为 $v_5v_2v_1v_3v_6v_4$，得另一个邻接矩阵

$$A_{1G}=\begin{pmatrix}0&0&1&0&1&1\\0&0&1&1&0&0\\1&1&0&0&1&0\\0&1&0&1&1&1\\1&0&1&1&0&1\\1&0&0&1&1&0\end{pmatrix}$$

在邻接矩阵A_{1G}中，如果先交换第1、3行，而后交换第1、3列；接着交换第3、4行，再交换第3、4列；接着交换第5、6行，再交换第5、6列；接着交换第4、5行，再交换第4、5列。那么就能由邻接矩阵A_{1G}得到邻接矩阵A_G。

9.2.3 图的操作

定义9.2.3 设图G = <V, E>。

1. 设e∈E，用G−e表示从G中去掉边e得到的图，称为**删除边e**。又设E′⊆E，用G−E′表示从G中删除E′中所有边得到的图，称为**删除E′**。
2. 设v∈V，用G−v表示从G中去掉结点v及v关联的所有边得到的图，称为**删除结点v**。又设V′⊂V，用G−V′ 表示从G中删除V′中所有结点及关联的所有边得到的图，称为**删除V′**。

定义9.2.3（续）

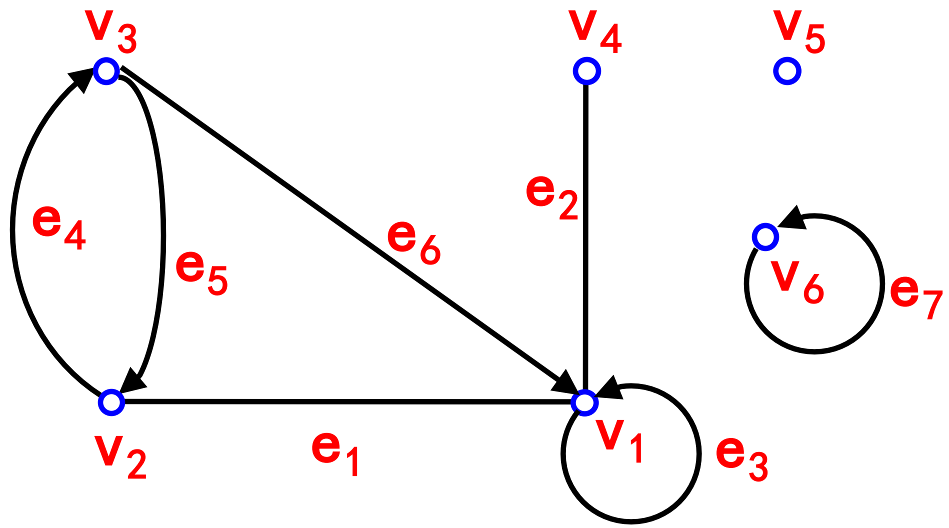
3. 设e =（u, v）∈E，用G\e表示从G中删除e，将e的两个端点u，v用一个新的结点w代替，使w关联除e外的u和v关联的一切边，称为边e的收缩。一个图G可以收缩为图H，是指H可以从G经过若干次边的收缩而得到。
4. 设u，v∈V(u，v可能相邻，也可能不相邻)，用G∪(u，v)表示在u，v之间加一条边(u，v)，称为加新边。

9.2.4 邻接点与邻接边

定义9.2.4 在图G = <V, E>中，若两个结点v_i和v_j是边e的端点，则称v_i与v_j互为邻接点(Adjacent Point)，否则v_i与v_j称为不邻接的；具有公共结点的两条边称为邻接边(Adjacent Edge)；两个端点相同的边称为环(Ring)或自回路(Self-Loop)；图中不与任何结点相邻接的结点称为孤立结点(Isolated Point)；仅由孤立结点组成的图称为零图(Null Graph)仅含一个结点的零图称为平凡图(Trivial Graph)；含有n个结点，m条边的图，称为(n, m)图。

例9.2.5

试写出下图所示图G的所有结点的邻接点、所有边的邻接边，并指出所有的孤立结点和环。



例9.2.5 分析

根据定义9.2.4，如果两个结点间有边相连，那么它们互为邻接点；如果两条边有公共结点，那么它们互为邻接边。需要注意的是，只要当一个结点处有环时，它才是自己的邻接点；由于一条边有两个端点，在计算邻接边时要把这两个端点都算上，例如e_2和e_4都是e_1的邻接边。所有边都是自己的邻接边。

例9.2.5 解

图G所有结点的邻接点和孤立结点，所有边的邻接边和环如下表所示图G既不是平凡图，也不是零图，而是一个(6， 7)图。

结点	邻接点	是否孤立结点
v_1	V_1, v_2, v_3, v_4	否
v_2	v_1, v_3	否
v_3	v_1, v_2	否
v_4	v_1	否
v_5		是
v_6	v_6	否

边	邻接边	是否环
e_1	$E_1, e_2, e_3, e_4, e_5, e_6$	否
e_2	e_1, e_2, e_3, e_6	否
e_3	e_1, e_2, e_3, e_6	是
e_4	e_1, e_4, e_5, e_6	否
e_5	e_1, e_4, e_5, e_6	否
e_6	$e_1, e_2, e_3, e_4, e_5, e_6$	否
e_7	e_7	是

9.2.5 图的分类

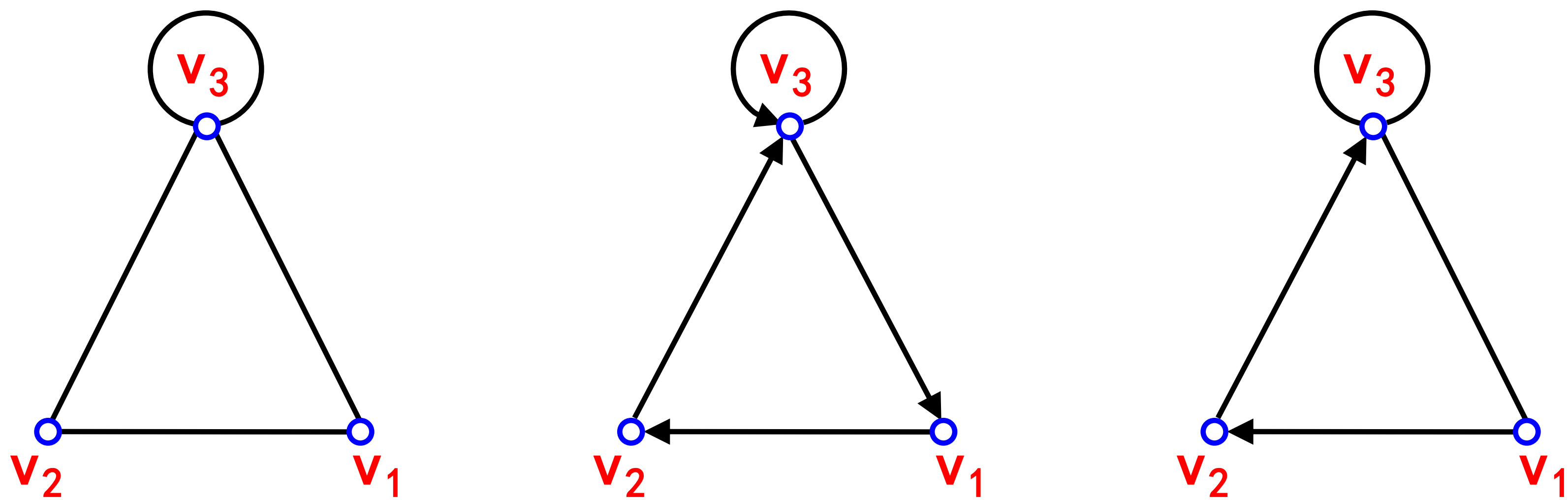
1. 按边有无方向分类

定义9.2.5　每条边都是无向边的图称为无向图(Undirected Graph)；每条边都是有向边的图称为有向图(Directed Graph)；有些边是无向边，而另一些边是有向边的图称为混合图(Mixed Graph)。

第6章的关系图都是有向图，这时邻接矩阵就是关系矩阵。

例9.2.6

试判断下图所示的三个图是无向图、有向图，还是混合图？



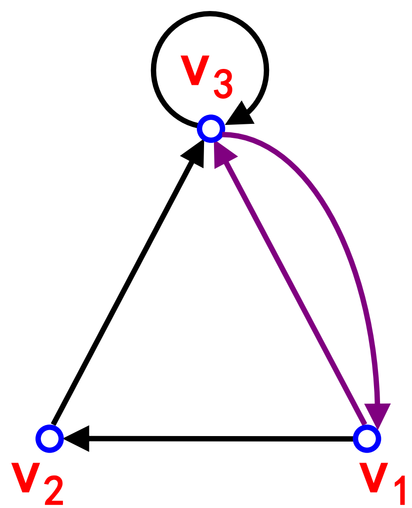
G_1 G_2 G_3

分析 判断无向图、有向图和混合图，仅仅看边有无方向就行了。

解 G_1为无向图，G_2为有向图，G_3为混合图。

说明

我们仅讨论无向图和有向图，至于混合图，我们可将其中的无向边看成方向相反的两条有向边，从而转化为有向图来研究。例如可将图G_3转化为下图。



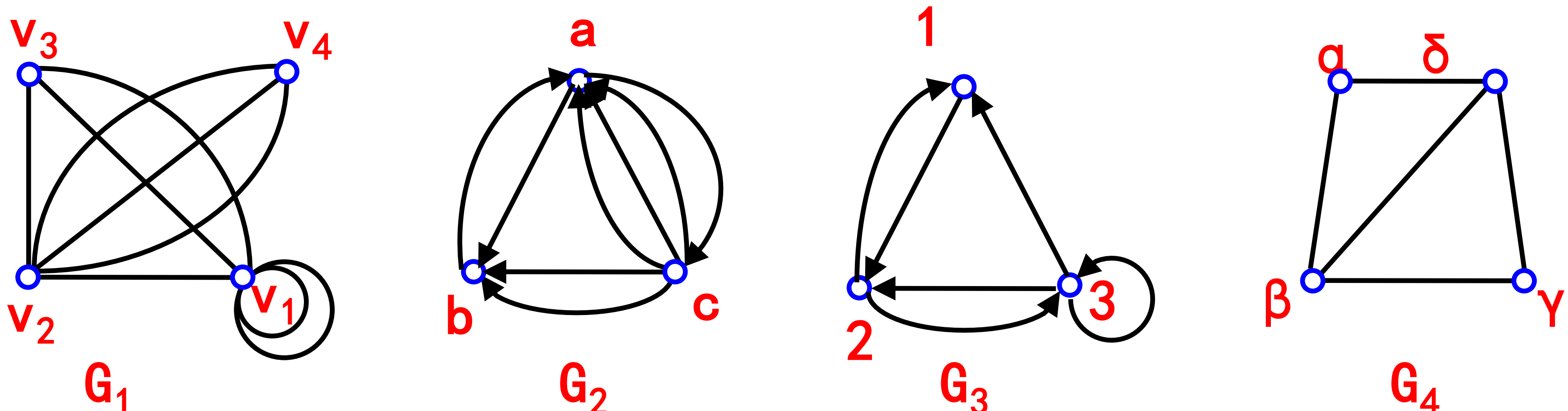
G_3

2. 按有无平行边分类

定义9.2.6 在有向图中，两结点间(包括结点自身间)若有同始点和同终点的几条边，则这几条边称为平行边(Parallel Edge)；在无向图中，两结点间(包括结点自身间)若有几条边，则这几条边称为平行边。两结点a、b间相互平行的边的条数称为边(a, b)或<a, b>的重数(Repeated Number)。含有平行边的图称为多重图(Multigraph)；非多重图称为线图(Line Graph)；无环的线图称为简单图(Simple Graph)。

例9.2.7

试判断下图所示的4个图是多重图、线图，还是简单图？并指出多重图中所有平行边的重数。



解　G_1、G_2是多重图，G_3是线图，G_4是简单图。G_1中平行边(v_1, v_3)的重数为2，(v_2, v_4)的重数为3；G_2中平行边<c，a>的重数为3，<c，b>的重数为2。

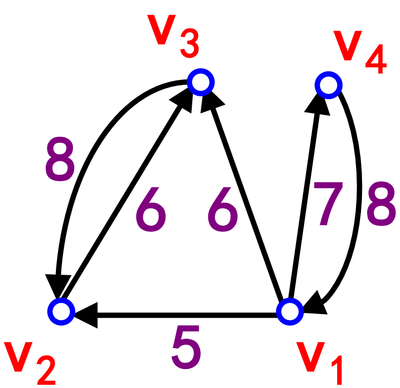
分析　多重图和线图，仅仅看有无平行边；简单图是一种特殊的线图，仅仅无环而已。两个端点都相同的无向边是平行边，但两个端点都相同的有向边不一定是平行边，例如G_2中的<c，a>和<a，c>就不是平行边，因此<c，a>的重数是3，而不是4。

3. 按边或结点是否含权分类

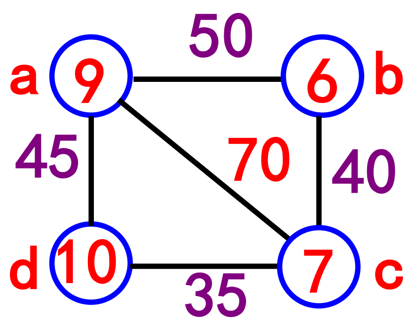
定义9.2.7 赋权图(Weight Graph)G是一个三重组<V, E, g>或四重组<V, E, f, g>，其中V是结点集合，E是边的集合，f是从V到非负实数集合的函数，g是从E到非负实数集合的函数。

例9.2.8

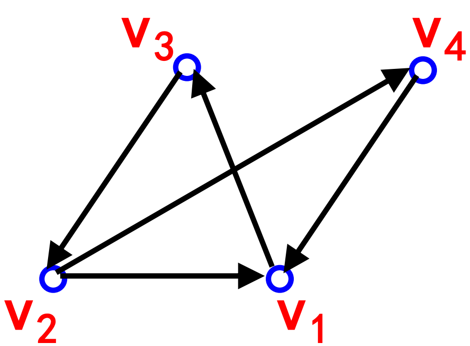
下图所示的图哪个是赋权图，哪个是无权图？是赋权图的请写出相应的函数。



G_1



G_2



G_3

例9.2.8 解

分析 对每条边都赋予非负实数值，或者对每条边和每个结点都赋予非负实数值的图就是赋权图。图G_1的每条边都赋予了非负实数值，因此图G_1是赋权图。图G_2的每条边和每个结点都赋予了非负实数值，因此图G_2是赋权图。而图G_3的边没有赋予非负实数值，因此图G_3不是赋权图。

在图中，G_1、G_2是赋权图，G_3不是赋权图。记图$G_1=\langle V_1, E_1, g_1\rangle$和$G_2=\langle V_2, E_2, f_2, g_2\rangle$，其中：

$g_1(\langle v_1, v_2\rangle)=5$，$g_1(\langle v_1, v_3\rangle)=6$，

$g_1(\langle v_1, v_4\rangle)=7$，$g_1(\langle v_2, v_3\rangle)=6$，

$g_1(\langle v_3, v_2\rangle)=8$，$g_1(\langle v_4, v_1\rangle)=8$。

$f_2(a)=9$，$f_2(b)=6$，$f_2(c)=7$，$f_2(d)=10$；

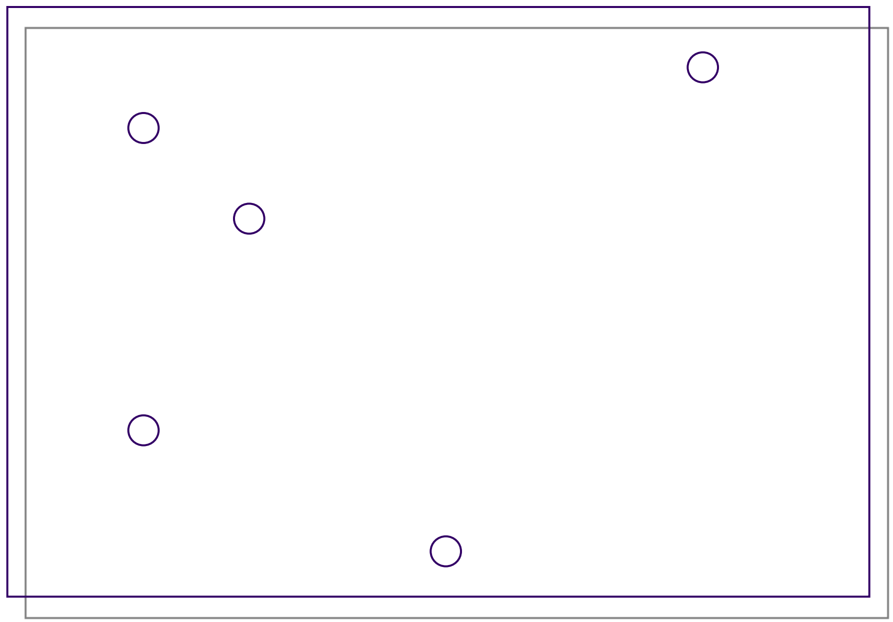
$g_2((a, b))=50$，$g_2((a, c))=70$，

$g_2((a, d))=45$，$g_2((b, d))=40$，

$g_2((c, d))=35$，

赋权图的一个应用

制造加工业中的钻孔路径



带螺孔钉的金属板

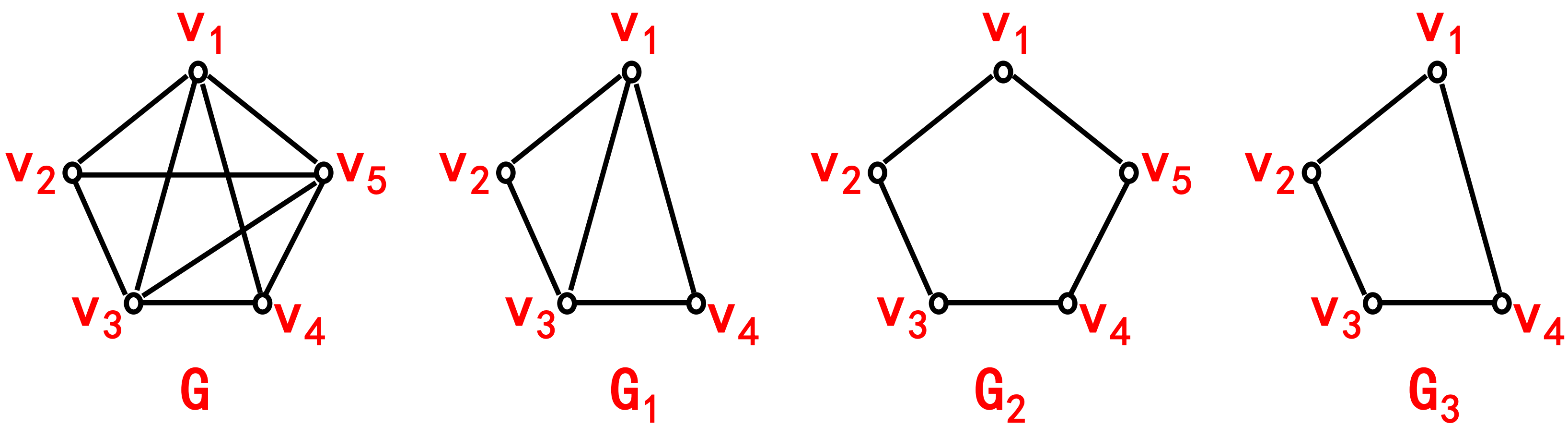
9.2.6 子图与补图

定义9.2.8 设有图$G = \langle V, E\rangle$和图$G_1 = \langle V_1, E_1\rangle$。

1. 若$V_1 \subseteq V$，$E_1 \subseteq E$，则称G_1是G的**子图**(Subgraph)，记为$G_1 \subseteq G$。
2. 若$G_1 \subseteq G$，且$G_1 \neq G$(即$V_1 \subset V$或$E_1 \subset E$)，则称G_1是G的**真子图**(Proper Subgraph)，记为$G_1 \subset G$。
3. 若$V_1 = V$，$E_1 \subseteq E$，则称G_1是G的**生成子图**(Spanning Subgraph)。
4. 设$V_2 V$且$V_2 \neq \Phi$，以V_2为结点集，以两个端点均在V_2中的边的全体为边集的G的子图，称为V_2导出的G的子图，简称V_2的**导出子图**(Induced Subgraph)。

例9.2.9

判断下图中，G_1、G_2和G_3是否是图G的子图、真子图、生成子图、导出子图？



分析 由于子图、真子图和生成子图只需要判断点集和边集是否是图G的结点集和边集，因此容易得出G_1、G_2和G_3都是图G的子图、真子图，只有G_3是图G的生成子图。由于导出子图要求G中两个端点都在V_2中的边都在E_2中，而v_1v_3都不在G_2和G_3中，因此仅有G_1是G的导出子图。

解 G_1、G_2和G_3都是图G的子图、真子图；G_3是图G的生成子图；G_1是图G的$\{v_1, v_2, v_3, v_4\}$的导出子图。

注 每个图都是它自身的子图、生成子图和导出子图。

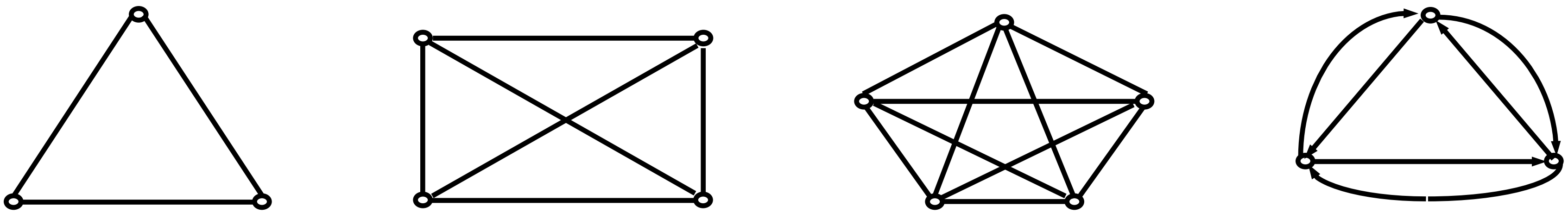
定义9.2.9

设G = <V, E>为一个具有n个结点的无向简单图，如果G中任意两个结点间都有边相连，则称G为无向完全图(Undirected Complete Graph)，简称G为完全图(Complete Graph)，记为K_n。

设G = <V, E>为一个具有n个结点的有向简单图，如果G中任意两个结点间都有两条方向相反的有向边相连，则称G为有向完全图(directed Complete Graph)，在不发生误解的情况下，也记为K_n。

对于完全图来说，其邻接矩阵除主对角元为0外，其它元素均为1。

例



无向完全图K_n的边数为 $C(n,2)=\frac{1}{2}n(n-1)$

有向完全图K_n的边数为 $P(n,2)=n(n-1)$

定义9.2.10

设G = <V, E>为简单图，G’ = <V, E_1>为完全图，则称G_1 = <V, E_1-E>为G的**补图**(Complement of Graph)，记为 $\overline{G}$。

注 在定义9.2.10中，当G为有向图时，则G’为有向完全图；当G为无向图时，则G’为无向完全图。

G的补图也可理解为从结点集V的完全图中删除G中的边剩下的图，即G与其补图的结点集是相同的，边集是相对于完全图的边集为全集的补集。

显然，若G_1= $\overline{G}$ ，则G= $\overline{G_1}$ ，即它们互为补图。

K_n的补图为n个结点的零图。

9.2.7 结点的度数与握手定理

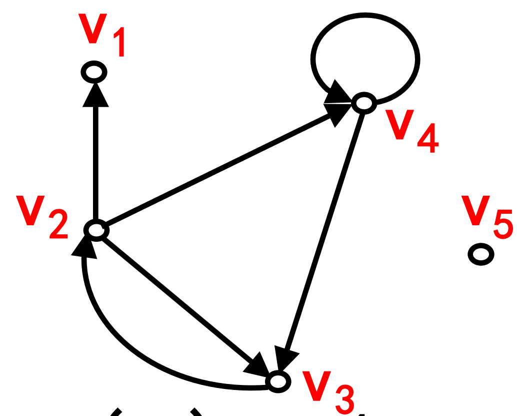
定义9.2.11 （1）图G = <V, E>中以结点v∈V为端点的边数(有环时计算两次)称为结点v的度数(Degree)，简称度，记为deg(v)。

（2）有向图G = <V, E>中以结点v为始点的边数称为v的出度(Out-Degree)，记为$deg^+(v)$；以结点v为终点的边数称为v的入度(In-Degree)，记为$deg^-(v)$。显然，$deg(v) = deg^+(v)+deg^-(v)$。

（3）对于图G = <V, E>，度数为1的结点称为悬挂结点(Hanging Point)，以悬挂结点为端点的边称为悬挂边(Hanging Edge)。

例9.2.12

求右图中所有结点的度数、出度和入度，指出悬挂结点和为悬挂边。



解 $\deg(v_1)=1$，$\deg^+(v_1)=0$，$\deg^-(v_1)=1$

$\deg(v_2)=4$，$\deg^+(v_2)=3$，$\deg^-(v_2)=1$

$\deg(v_3)=3$，$\deg^+(v_3)=1$，$\deg^-(v_3)=2$

$\deg(v_4)=4$，$\deg^+(v_4)=2$，$\deg^-(v_4)=2$

$\deg(v_5)=0$，$\deg^+(v_5)=0$，$\deg^-(v_5)=0$

v_1为悬挂结点，$\langle v_2, v_1\rangle$为悬挂边。

分析　求结点的度数非常简单，只需要数一下以该结点为端点的边数，出度只需要数一下以其为始点的边数，入度只需要数一下以其为终点的边数，无向环算2度，又有向环出度和入度各算1度。只有度数为1的才是悬挂结点，以悬挂结点为端点的边才是悬挂边。

利用邻接矩阵描述

设图G = <V, E>, V = $\{v_1, v_2, \cdots, v_n\}$的邻接矩阵为

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

若G是无向图，则A中第i行元素是由结点v_i所关联的边所决定，其中为1的元素数目等于v_i的度数，即，

$$\deg(v_i) = \sum_{k=1}^{n} a_{ik} + a_{ii} \qquad \deg(v_i) = \sum_{k=1}^{n} a_{ki} + a_{ii}$$

利用邻接矩阵描述

若G是有向图，则A中第i行元素是由结点v_i为始点的边所决定，其中为1的元素数目等于v_i的出度，即

$$\deg^{+}(v_i)=\sum_{k=1}^{n}a_{ik}$$

A中第i列元素是由结点v_i为终点的边所决定，其中为1的元素数目等于v_i的入度，即。

$$\deg^{-}(v_i)=\sum_{k=1}^{n}a_{ki}$$

定理9.2.1(握手定理)

图中结点度数的总和等于边数的二倍，即设图G = <V, E>，则有

$$\sum_{v \in V} \deg(v) = 2|E|$$

[illegible] 1736年 [illegible] 如果许多人在见面时握了手，两只手握在一起，被握过手的总次数为偶数。故此定理称为图论的基本定理或握手定理。

推论9.2.1

图中度数为奇数的结点个数为偶数。

今后常称度数为奇数的结点为奇度数结点(Odd Degree Point)，度数为偶数的结点为偶度数结点(Even Degree Point)。

证明 设$V_1=\{v \mid v\in V$且$\deg(v)$为奇数$\}$，$V_2=\{v \mid v\in V$且$\deg(v)$为偶数$\}$。显然，$V_1\cap V_2=\varnothing$，且$V_1\cup V_2=V$，于是

$$\sum_{v\in V}\deg(v)=\sum_{v\in V_1}\deg(v)+\sum_{v\in V_2}\deg(v)=2|E|,$$

式中$2|E|$和$\sum_{v\in V_2}\deg(v)$（偶数之和为偶数）均为偶数，因而$\sum_{v\in V_1}\deg(v)$也为偶数。于是$|V_1|$为偶数，即度数为奇数的结点个数为偶数。

定理9.2.2

有向图中各结点的出度之和等于各结点的入度之和，等于边数，即设有向图G = <V, E>，则有

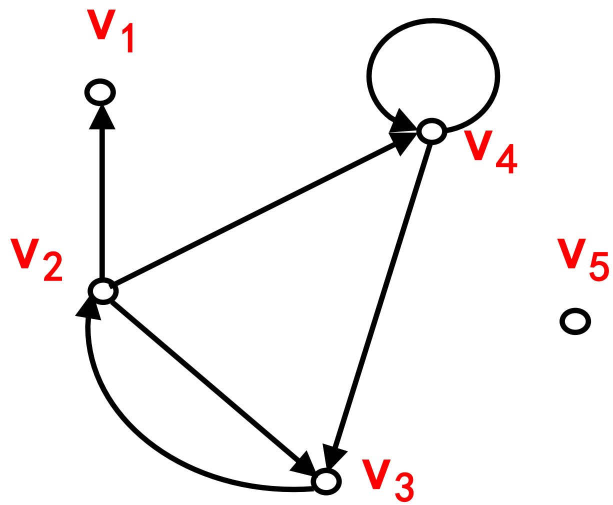
$$\sum_{v \in V} \deg^{+}(v) = \sum_{v \in V} \deg^{-}(v) = |E|$$

分析 利用握手定理，考虑一条边贡献一个出度和一个入度即可。

证明 因为每条有向边具有一个始点和一个终点(环的始点和终点是同一个结点)，因此，每条有向边对应一个出度和一个入度。图G中有|E|条有向边，则G中必产生|E|个出度，这|E|个出度即为各结点的出度之和，G中也必产生|E|个入度，这|E|个入度即为各结点的入度之和。因而，在有向图中，各结点的出度之和等于各结点的入度之和，都等于边数|E|。

定义9.2.12

设 $V = \{v_1, v_2, \cdots, v_n\}$ 为图G的结点集，称 $(deg(v_1), deg(v_2), \cdots, deg(v_n))$ 为G的**度数序列**(Degree Sequence)。



上图的度数序列为(1，4，3，4，0)。

例9.2.14

（1）(3，5，1，4)，(1，2，3，4，5)能成为图的度数序列吗？为什么？

（2）已知图G中有15条边，2个度数为4的结点，4个度数为3的结点，其余结点的度数均小于等于2，问G中至少有多少个结点？为什么？

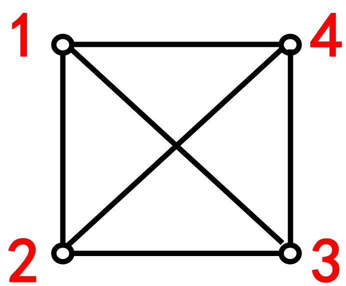
分析 这是一种利用握手定理及其推论的例子。度数序列中的每个元素都是结点的度数，利用推论9.2.1，一定有偶数个奇数。

解 （1）由于这两个序列中，奇数度数的个数均为奇数，由握手定理的推论9.2.1知，它们都不能成为图的度数序列。

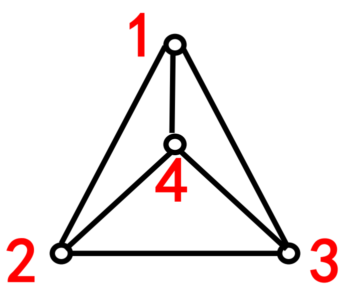
(2)图中边数为15，由握手定理知，G中所有结点的度数之和为30，2个度数为4的结点，4个度数为3的结点占去20度，还剩下10度。若其余全是度数为2的结点，还需要5个结点来占用这10度，所以G至少有11个结点。

9.2.8 图的同构

图是表达事物之间关系的工具，因此，图的最本质的内容是结点和边的关联关系。而在实际画图时，由于结点的位置不同，边的长短曲直不同，同一事物间的关系可能画出不同形状的图来。例如下图中的两个图G_1和G_2实际上是同一个图K_4。



G_1



G_2

定义9.2.13

设两个图G = <V, E>和G' = <V', E'>，如果存在**双射**函数g：V→V'，使得对于任意的e = (v_i, v_j)(或者<v_i, v_j>)∈E当且仅当e' = (g(v_i), g(v_j))(或者<g(v_i), g(v_j)>)∈E'，并且e与e'的**重数相同**，则称G与G'**同构**(Isomor- phism)，记为G≌G'。

对于同构，形象地说，若图的结点可以任意挪动位置，而边是完全弹性的，只要在不拉断的条件下，一个图可以变形为另一个图，那么这两个图是同构的。

两个图同构的必要条件

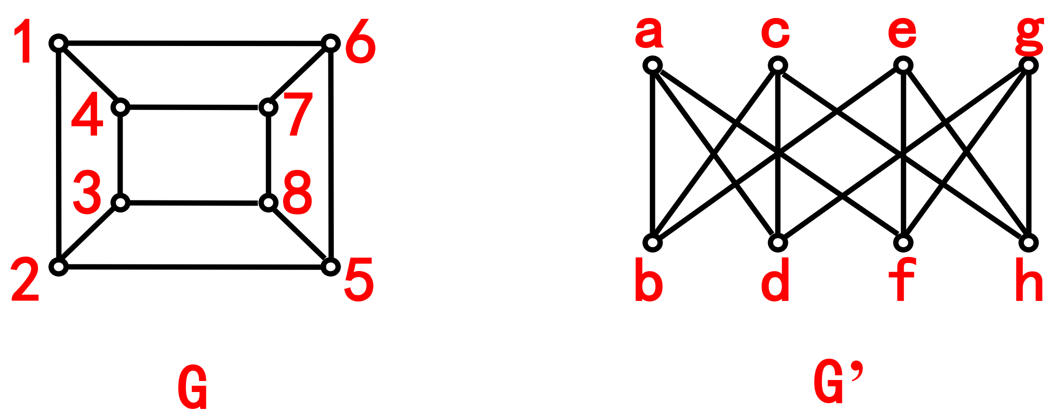
（1）结点数目相同；

（2）边数相同；

（3）度数相同的结点数相同。

例9.2.14

试证明下图中，G≌G’。



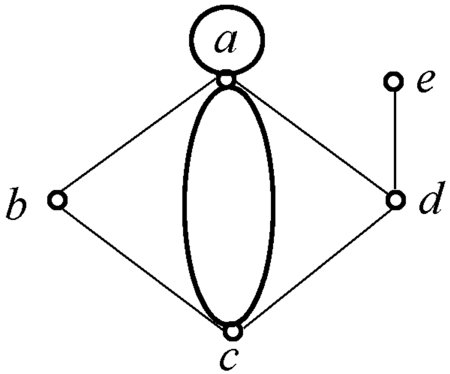
分析 证明两个图同构，关键是找到满足要求的结点集之间的双射函数，现在还没有好的办法，只有凭经验去试。

证明 构造结点之间的双射函数f如下：f(1)=a，f(2)=b，f(3)=c，f(4)=d，f(5)=e，f(6)=f，f(7)=g，f(8)=h

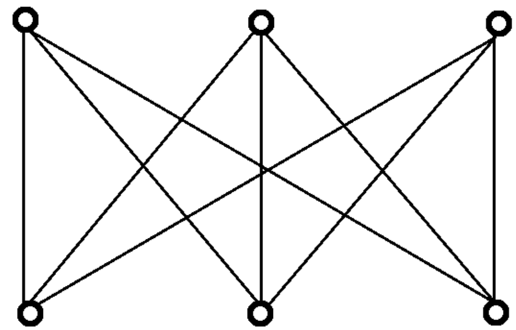
容易验证，f满足定义9.2.13，所以G≌G’。

另一组判断同构的例子

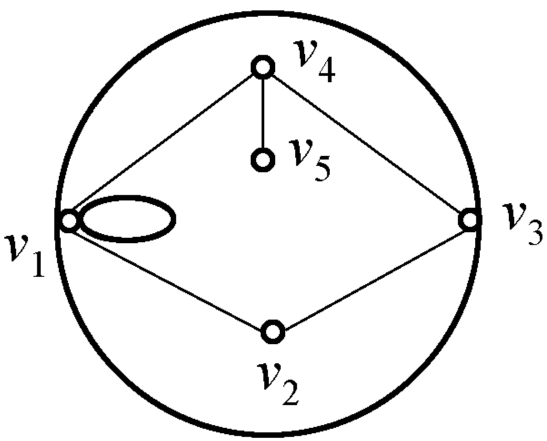
下面两组图形是否同构，如果同构，指出双射函数



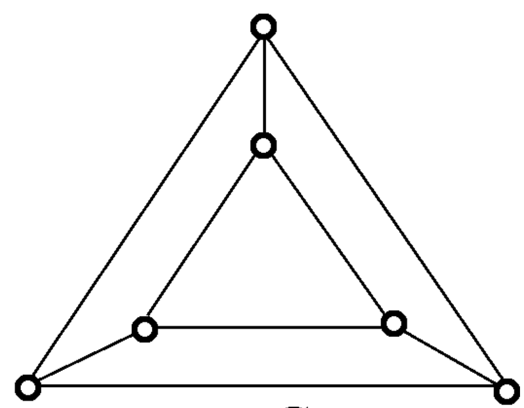
G_1



G_3



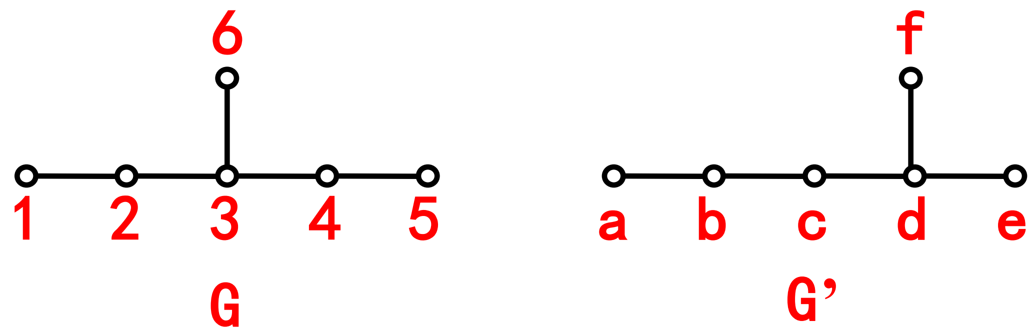
G_2



G_4

例9.2.15

证明下图中，G与G’不同构



分析 证明两个图不同构，通常用反证法。

证明 假设G≌G’，双射函数为f。由定义9.2.13，v与f(v)的度数一定相同，因此有f(3)=d。G中3与一个度数为1的结点6邻接，而G’中d与两个度数为1的结点e、f邻接，矛盾。

注意 图同构的三个必要条件不是充分条件。在上图的G与G’两个图，虽然满足以上三个条件，但不同构。

寻找一种简单而有效的方法来判断图的同构是图论中一个重要而未解决的问题。

9.2.9 图的难点

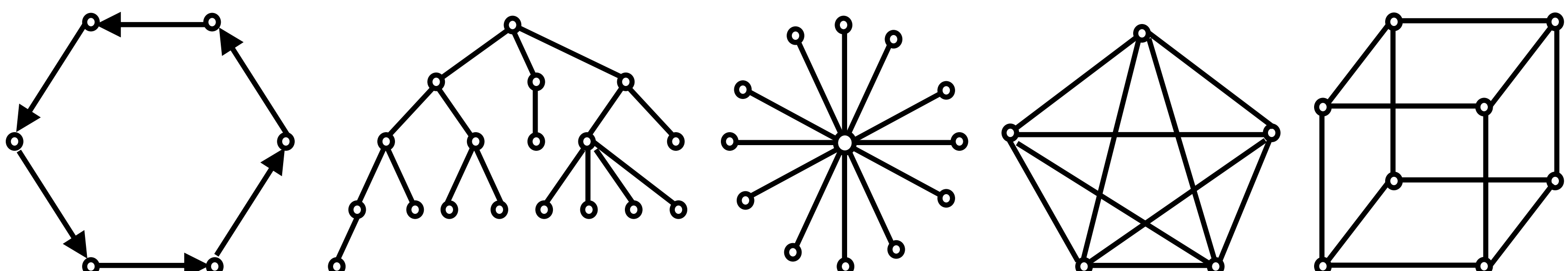
- 图是由两个集合构成的，可以利用集合的有关知识来研究它，如子图、完全图、补图等；
- 图的计算机表示就是它的邻接矩阵，实际中的图都是很大的，可能有成千上万的结点和边，用手工处理是很难想象的；
- 判断图的同构，是图论中一个重要而未解决的问题。现在还没有好的办法，只有凭经验按定义去试；
- 握手定理是图论的基本定理，很多理论都是以它为基础的，必须熟练掌握，并能灵活运用。

9.2.10 图的应用

自从克希荷夫运用图论从事电路网络的拓扑分析以来，尤其是近几十年来，网络理论的研究和应用十分引人注目，电路网络、运输网络、信息网络等与工程和应用紧密相关的课题受到了高度的重视，其中多数问题都与优化有关，涉及到问题的费用、容量、可靠性和其它性能指标，有重要的应用价值。网络应用的一个重要方面就是通讯网络。如电话网络、计算机网络、管理信息系统、医疗数据网络、银行数据网络、开关网络等等。这些网络的基本要求是网络中各个用户能够快速安全地传递信息，不产生差错和故障，同时使建造和维护网络所需费用低。因此通讯网络涉及的因素很多，我们就不详细介绍，仅说明一些基本知识。

通讯网络

通讯网络中最重要的整体问题之一是网络的结构形式。通讯网络是一个强连通的有向图，根据用途和各种性能指标有着不同的结构形式，下图给出了一些典型的结构。



(a) (b) (c) (d) (e)

环（Ring）型网络

树（Tree）型网络

星

分布式（Dist

网

立方体（Cube）型网络

9.3.1 通路与回路

通路与回路是图论中两个重要的基本概念。本小节所述定义一般来说既适合有向图，也适合无向图，否则，将加以说明或分开定义。

定义9.3.1（续）

给定图G=<V,E>中结点和边相继交错出现的序列 $\Gamma=v_0e_1v_1e_2v_2\cdots e_kv_k$。

1. 若Γ中边e_i的两端点是v_{i-1}和v_i（G是有向图时要求v_{i-1}与v_i分别是e_i的始点和终点），i=1,2,…,k，则称Γ为结点v_0到结点v_k的通路(Entry)。v_0和v_k分别称为此通路的始点和终点，统称为通路的端点。通路中边的数目k称为此通路的长度(Length)。当$v_0=v_n$时，此通路称为回路(Circuit)。

定义9.3.1

2. 若通路中的所有边互不相同，则称此通路为简单通路(Simple Entry)或一条迹；若回路中的所有边互不相同，则称此回路为简单回路(Simple Circuit)或一条闭迹。

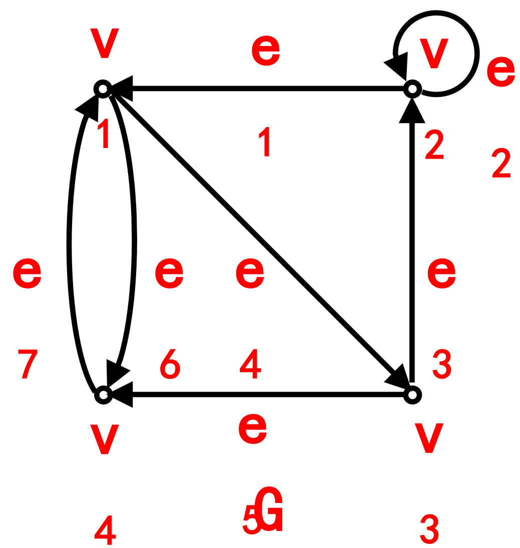
3. 若通路中的所有结点互不相同（从而所有边互不相同），则称此通路为基本通路(Basic Entry)或者初级通路、路径；若回路中除$v_0=v_k$外的所有结点互不相同（从而所有边互不相同），则称此回路为基本回路(Basic Circuit)或者初级回路、圈。

说明

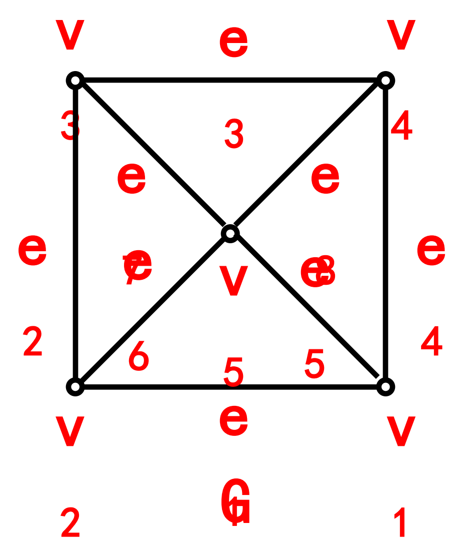
1. 回路是通路的特殊情况。因而，我们说某条通路，它可能是回路。但当我们说一基本通路时，一般是指它不是基本回路的情况。
2. 基本通路（回路）一定是简单通路（回路），但反之不真。因为没有重复的结点肯定没有重复的边，但没有重复的边不能保证一定没有重复的结点。
3. 在不会引起误解的情况下，一条通路 $v_0e_1v_1e_2v_2\cdots e_nv_n$ 也可以用边的序列 $e_1e_2\cdots e_n$ 来表示，这种表示方法对于有向图来说较为方便。在线图中，一条通路 $v_0e_1v_1e_2v_2\cdots e_nv_n$ 也可以用结点的序列 $v_0v_1v_2\cdots v_n$ 来表示。

例9.3.1

判断下图G1中的回路$v_3e_5v_4e_7v_1e_4v_3e_3v_2e_1v_1e_4v_3$、$v_3e_3v_2e_2v_2e_1v_1e_4v_3$、$v_3e_3v_2e_1v_1e_4v_3$是否是简单回路、基本回路？图$G_2$ 中的通路$v_1e_1v_2e_6v_5e_7v_3e_2v_2e_6v_5e_8v_4$、$v_1e_5v_5e_7v_3e_2v_2e_6v_5e_8v_4$、$v_1e_1v_2e_6v_5e_7v_3e_3v_4$是否是简单通路、基本通路？并求其长度。



G_1



G_2

例9.3.1 分析

判断一条通（回）路是否是简单通（回）路、基本通（回）路，主要是看它有无重复的边、结点。

在图G_1中，$v_3e_5v_4e_7v_1e_4v_3e_3v_2e_1v_1e_4v_3$中有重复的边$e_4$，因此它不是简单回路，也不是基本回路；

$v_3e_3v_2e_2v_2e_1v_1e_4v_3$虽然没有重复的边，但有重复的结点$v_2$，因此只能是简单回路，而不是基本回路；

而$v_3e_3v_2e_1v_1e_4v_3$中既没有重复的边，也没有重复的结点，因此既是基本回路，也是简单回路；

例9.3.1 分析

在图G_2中，$v_1e_1v_2e_6v_5e_7v_3e_2v_2e_6v_5e_8v_4$中有重复的边$e_6$，因此它不是它既不是简单通路，也不是基本通路；

$v_1e_5v_5e_7v_3e_2v_2e_6v_5e_8v_4$虽然没有重复的边，但有重复的结点$v_5$，因此只能简单通路，但不是基本通路；

$v_1e_1v_2e_6v_5e_7v_3e_3v_4$中既没有重复的边，也没有重复的结点，因此既是基本通路，也是简单通路。

至于通（回）路的长度就是其包含的边的数目，这只需要数一数就行了。

例9.3.1 解

在图G_1中，$v_3e_5v_4e_7v_1e_4v_3e_3v_2e_1v_1e_4v_3$是一条长度为6的回路，但既不是简单回路，也不是基本回路；

$v_3e_3v_2e_2v_2e_1v_1e_4v_3$是一条长度为4的简单回路，但不是基本回路；

$v_3e_3v_2e_1v_1e_4v_3$是一条长度为3的基本回路，也是简单回路；

例9.3.1 解（续）

在图G_2中，$v_1e_1v_2e_6v_5e_7v_3e_2v_2e_6v_5e_8v_4$是一条长度为6的通路，但既不是简单通路，也不是基本通路；

$v_1e_5v_5e_7v_3e_2v_2e_6v_5e_8v_4$是一条长度为5的简单通路，但不是基本通路；

$v_1e_1v_2e_6v_5e_7v_3e_3v_4$是一条长度为4的基本通路，也是简单通路。

说明

在图G_1中，简单回路$v_3e_3v_2e_2v_2e_1v_1e_4v_3$既可以用边的序列$e_3e_2e_1e_4$来表示，也可以用结点的序列$v_3v_2v_2v_1v_3$来表示；

在图G_2中，简单通路$v_1e_5v_5e_7v_3e_2v_2e_6v_5e_8v_4$既可以用边的序列$e_5e_7e_2e_6e_8$来表示，也可以用结点的序列$v_1v_5v_3v_2v_5v_4$来表示。

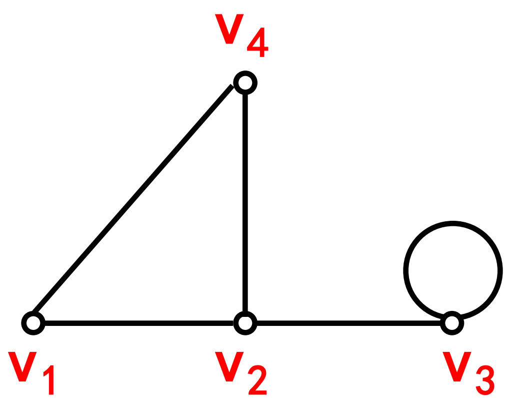
定理9.3.1

设G = <V, E>为线图，V = {v_1, v_2, …, v_n}，A = $(a_{ij})_{n\times n}$为G的邻接矩阵，$A_m=(a_{ij}^{(m)})_{n\times n}$。则

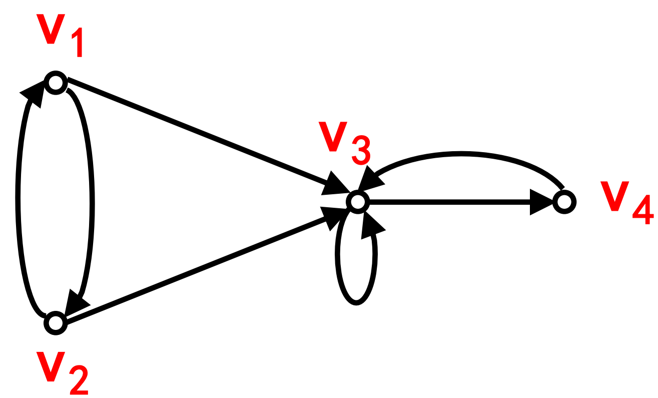
- $a_{ij}^{(m)}$ 为从结点v_i到结点v_j长度为m的通路数目；
- $a_{ii}^{(m)}$ 为结点v_i到自身的长度为k的回路数目；
- $\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}^{(m)}$为G中长度为m的通路（含回路）总数。

例9.3.2

求下图中图G_1和G_2的从结点v_1到结点v_3长度为2和3的通路数目及所有长度为2和3的通路数目。



G_1



G_2

分析 利用定理9.3.3，求图中长度为m的通路数目，只需要先写出图的**邻接矩阵**，然后计算邻接矩阵的m次方即可。

例9.3.2 解

在图中，G_1是无向线图，G_2是有向线图，它们的邻接矩阵分别为：

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

例9.3.2 解(续)

下面计算邻接矩阵的幂，

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

$$a_{13}^{(2)}=1 \qquad \sum_{i=1}^{4}\sum_{j=1}^{4}a_{ij}^{(2)}=21 \qquad \sum_{i=1}^{4}a_{ii}^{(2)}=9$$

因而G_1中从结点v_1到结点v_3长度为2通路数目为1，长度为2的通路（含回路）总数为21，其中9条为回路。

例9.3.2 解(续)

$$(A(G_2))^2=\begin{pmatrix}0&1&1&0\\1&0&1&0\\0&0&1&1\\0&0&1&0\end{pmatrix}\times\begin{pmatrix}0&1&1&0\\1&0&1&0\\0&0&1&1\\0&0&1&0\end{pmatrix}=\begin{pmatrix}1&0&2&1\\0&1&2&1\\0&0&2&1\\0&0&1&1\end{pmatrix}$$

$$a_{13}^{(2)}=2 \qquad \sum_{i=1}^{4}\sum_{j=1}^{4}a_{ij}^{(2)}=13 \qquad \sum_{i=1}^{4}a_{ii}^{(2)}=4$$

G_2中从结点v_1到结点v_3长度为2通路数目为2，长度为2的通路（含回路）总数为13，其中5条为回路。

例9.3.2 解(续)

$$(A(G_1))^3 = A(G_1)\cdot(A(G_1))^2 = \begin{pmatrix} 2 & 4 & 2 & 3 \\ 4 & 3 & 4 & 4 \\ 2 & 4 & 3 & 2 \\ 3 & 4 & 2 & 2 \end{pmatrix}$$

$$a_{13}^{(3)} = 2 \qquad \sum_{i=1}^{4}\sum_{j=1}^{4} a_{ij}^{(3)} = 48 \qquad \sum_{i=1}^{4} a_{ii}^{(3)} = 10$$

因而G_1中从结点v_1到结点v_3长度为3的通路数目为2，长度为3的通路（含回路）总数为48，其中10条为回路。

例9.3.2 解(续)

$$(A(G_2))^3 = A(G_2)\cdot(A(G_2))^2 = \begin{pmatrix} 0 & 1 & 4 & 2 \\ 1 & 0 & 4 & 2 \\ 0 & 0 & 3 & 2 \\ 0 & 0 & 2 & 1 \end{pmatrix}$$

$$a_{13}^{(3)} = 4 \qquad \sum_{i=1}^{4}\sum_{j=1}^{4} a_{ij}^{(3)} = 22 \qquad \sum_{i=1}^{4} a_{ii}^{(3)} = 4$$

G_2中从结点v_1到结点v_3长度为3的通路数目为4，长度为3的通路（含回路）总数为22，其中4条为回路。

定义9.3.2

在图$G = \langle V, E\rangle$中，v_i，$v_j \in V$。

（1）如果从v_i到v_j存在通路，则称v_i到v_j是可达的，否则称v_i到v_j不可达。规定：任何结点到自己都是可达的。

（2）如果v_i到v_j可达，则称长度最短的通路为从v_i到v_j的短程线(Geodesic)，从v_i到v_j的短程线的长度称为从v_i到v_j的距离(Distance)，记为$d(v_i, v_j)$。如果v_i到v_j不可达，则通常记为$d(v_i, v_j) = \infty$。

定理9.3.2

在一个具有n个结点的图中，如果从结点v_i到结点v_j（$v_i \neq v_j$）存在一条通路，则从v_i到v_j存在一条长度不大于n-1的通路。

分析　通路的长度为序列中的结点数减1，如果结点不重复，最多n个，因此通路长度最多n-1；如果结点有重复，则在重复的结点间构成一条回路，删除这条回路，剩下的仍然是从结点v_i到结点v_j的通路。一直删下去，直到无重复结点为止，这样定理就得证了。

几个结论

推论9.3.1 在一个具有n个结点的图中，如果从结点v_i到结点v_j（$v_i \neq v_j$）存在一条通路，则从v_i到v_j存在一条长度不大于n-1的基本通路。

定理9.3.3 在一个具有n个结点的图中，如果存在经过结点v_i回路，则存在一条经过v_i的长度不大于n的回路。

推论9.3.2 在一个具有n个结点的图中，如果存在经过结点v_i回路，则存在一条经过v_i的长度不大于n的基本回路。

利用邻接矩阵判断可达

利用定理9.3.2和定理9.3.3，我们可以通过计算图的邻接矩阵及其幂的方法来判断从v_i到v_j是否可达，以及从v_i到v_j的距离。

设矩阵 $B_n = A+A^2+A^3+\cdots+A^n$

则B^n中的元素

$$b_{ij}^{(n)} = a_{ij}^{(1)} + a_{ij}^{(2)} + \cdots + a_{ij}^{(n)} = \sum_{m=1}^{n} a_{ij}^{(m)}$$

表示图G中从结点v_i到结点v_j的长度小于等于n的通路总数，若$i = j$，$b_{ii}^{(n)}$为G中结点v_i到自身的长度小于等于n的回路总数。

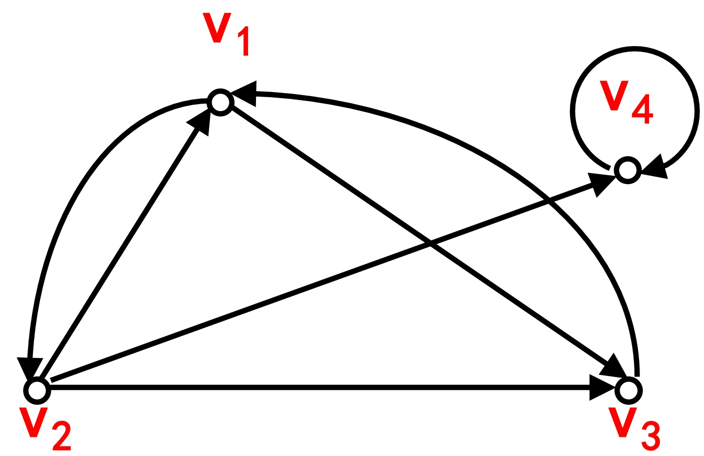
定理9.3.4

设G = <V, E>为线图，V = {v_1, v_2, …, v_n}，A = $(a_{ij})_{n\times n}$为G的邻接矩阵，A^m = $(a_{ij}^{(m)})$nxn，m=1, 2, …, n；B^n = $(b_{ij}^{(n)})$nxn = $A+A^2+A^3+\cdots+A^n$。则有：如果 $b_{ij}^{(n)}>0$，那么从v_i到v_j可达，否则不可达；并且

$$d(v_i,v_j)=\begin{cases}\infty, & \text{如果所有}a_{ij}^{(1)},a_{ij}^{(2)},\cdots,a_{ij}^{(n)}\text{均为}0\\ k, & \text{否则，}k=\min\{m \mid a_{ij}^{(m)}\neq 0, m=1,2,\cdots,n\}\end{cases}$$

例9.3.3

判断右图中图G中结点之间的可达关系，并求任两结点间的距离。



分析　利用定理9.3.4，先写出图的邻接矩阵A，然后计算A的幂即可。

例9.3.3 解

在图中，G的邻接矩阵及其2、3、4次幂分别为：

$$A=\begin{pmatrix}0&1&1&0\\1&0&1&1\\1&0&0&0\\0&0&0&1\end{pmatrix}\qquad A^2=\begin{pmatrix}2&0&1&1\\1&1&1&1\\0&1&1&0\\0&0&0&1\end{pmatrix}$$

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

例9.3.3 解(续)

从而有

$$B_4 = A + A^2 + A^3 + A^4 = \begin{pmatrix} 7 & 4 & 7 & 5 \\ 7 & 4 & 7 & 7 \\ 4 & 3 & 4 & 2 \\ 0 & 0 & 0 & 4 \end{pmatrix}$$

故从 v_1 到 v_1, v_2, v_3, v_4 都是可达的；从 v_2 到 v_1, v_2, v_3, v4都是可达的；从 v_3 到 v_1, v_2, v_3, v_4 都是可达的；从 v_4 到 v_4 都是可达的，从 v_4 到 v_1, v_2, v_3 都是不可达的。并且有

$d(v_1, v_2)=d(v_1, v_3)=d(v_2, v_1)=d(v_2, v_3)=d(v_2, v_4)=d(v_3, v_1)=1$, $d(v_1, v_4)=d(v_3, v_2)=2$，$d(v_3, v_4)=3$，$d(v_4, v_1)=d(v_4, v_2)=d(v_4, v_3)=\infty$。

定义9.3.3

设G = <V, E>是一个线图，其中V = $\{v_1, v_2, ..., v_n\}$，并假定结点已经有了从v_1到v_n的次序，称n阶方阵P = $(p_{ij})_{n\times n}$为图G的可达性矩阵(Accessibility Matrix)，其中

$$p_{ij}=\begin{cases}1, & \text{当}v_i\text{到}v_j\text{至少存在一条非零长度的通路}\\ 0, & \text{否则}\end{cases}$$

$i,j=1,2,3,\cdots,n$

定理9.3.5

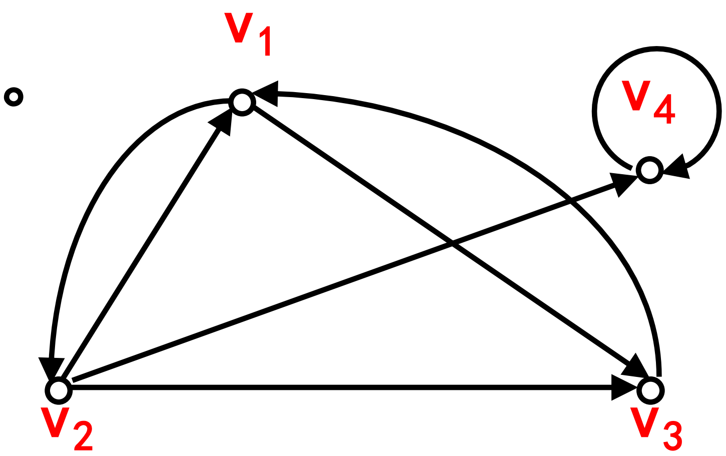
设G = <V, E>为线图，A、P分别是G的邻接矩阵和可达性矩阵，则有

$$P = A \vee A^{(2)} \vee A^{(3)} \vee \cdots \vee A^{(n)} = \bigvee_{i=1}^{n} A^{(i)}$$

这里，$A^{(i)}$表示做矩阵布尔乘法的i次幂。

例9.3.4

求右图中图G中的可达性矩阵。



分析 直接利用定理9.3.5，先计算图的邻接矩阵A布尔乘法的的2、3、4次幂，然后做布尔加即可。

解 在图9.3.4中，G的邻接矩阵及其2、3、4次布尔乘法幂分别为：

$$A=\begin{pmatrix}0&1&1&0\\1&0&1&1\\1&0&0&0\\0&0&0&1\end{pmatrix}\qquad A^{(2)}=A\odot A=\begin{pmatrix}1&0&1&1\\1&1&1&1\\0&1&1&0\\0&0&0&1\end{pmatrix}$$

例9.3.4 解（续）

$$A^{(3)} = A^{(2)} \odot A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A^{(4)} = A^{(3)} \odot A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

例9.3.4 解（续）

于是该图的可达性矩阵为：

$$\mathbf{P}=\mathbf{A}^{(1)}\vee\mathbf{A}^{(2)}\vee\mathbf{A}^{(3)}\vee\mathbf{A}^{(4)}=\begin{pmatrix}1&1&1&1\\1&1&1&1\\1&1&1&1\\0&0&0&1\end{pmatrix}$$

这与我们利用B4求得的结果完全一致。

9.3.2 无向图的连通性

定义9.3.4　若无向图G中的任何两个结点都是可达的，则称G是连通图(Connected Graph)，否则称G是非连通图(Unconnected Graph)或分离图(Separated Graph)。

无向完全图K_n（n≥1）都是连通图，而多于一个结点的零图都是非连通图。

利用邻接矩阵A和可达性矩阵P，显然有：

非平凡无向线图G是连通图当且仅当它的可达性矩阵P的所有元素均为1。

定理9.3.6

无向图G=<V, E>中结点之间的可达关系R定义如下:

R={<u, v>|u, v∈V, u到v可达},

则R是V上的等价关系。

分析 利用等价关系的定义，很容易证明R是自反、对称、传递的。

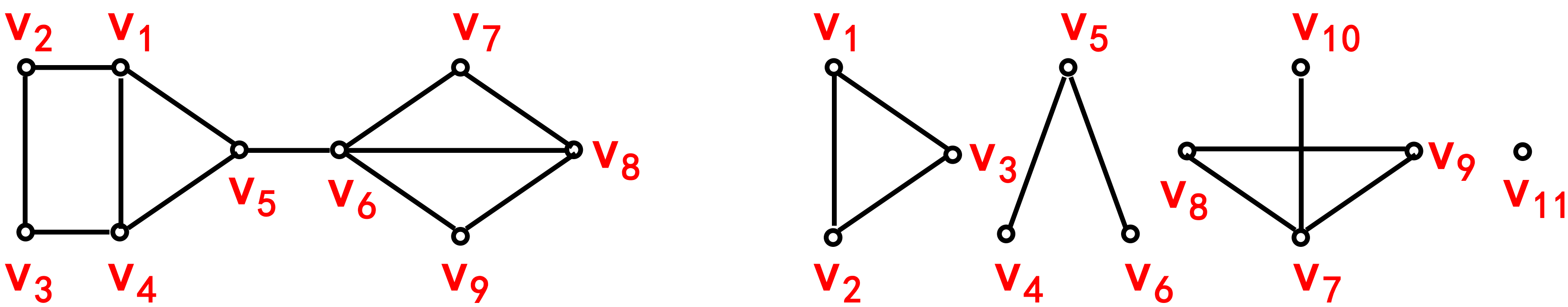
定义9.3.5

无向图G=<V, E>中结点之间的可达关系R的每个等价类导出的子图都称为G的一个连通分支(Connected Component)。用p(G)表示G中的连通分支个数。

显然，无向图G是连通图当且仅当p(G) = 1；每个结点和每条边都在且仅在一个连通分支中。

例9.3.5

判断下图中图G_1和G_2的连通性，并求其连通分支个数。



G_1　　　　G_2

分析　本题中图很简单，并且给出了图形，很容易看出G_1是连通图，G_2是非连通图。容易看出，G_2中可达关系的等价类为$\{v_1, v_2, v_3\}$，$\{v_4, v_5, v_6\}$，$\{v_7, v_8, v_9, v_{10}\}$，$\{v_{11}\}$，它们导出的子图即为$G_2$的4个连通分支。

解　在上图中，G_1是连通图，所以$p(G_1) = 1$。G_2是非连通图，且$p(G_2) = 4$。

9.3.3 有向图的连通性

定义9.3.5 设G

（1） 如

果

称 。否则

称G是

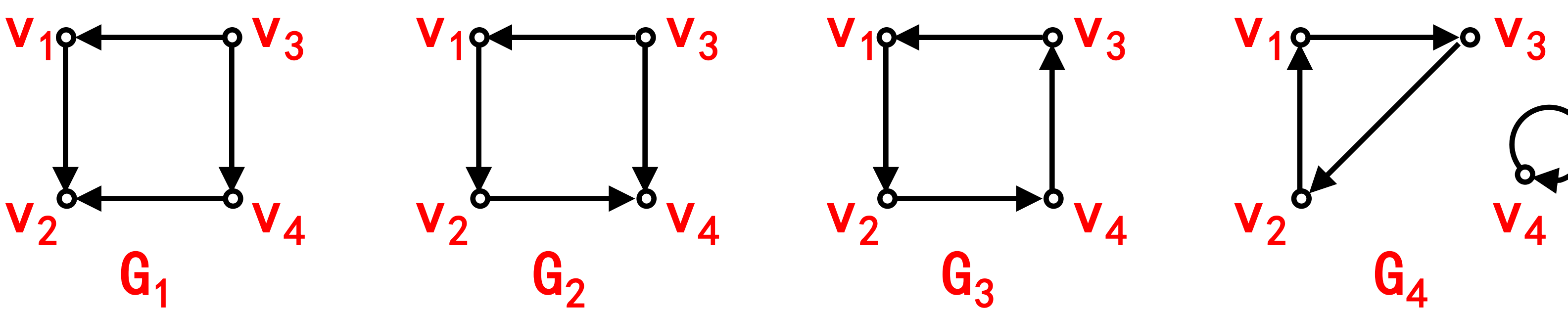
若有向图G是强连通图，则它必是单向连通图；若有向图G是单向连通图，则它必是（弱）连通图。但是上述二命题的逆均不成立。

（2）若G中任何一对结点之间至少有一个结点到另一个结点是可达的，则称G是单向连通图(Unilaterally Connected Graph)；

（3）若G中任何一对结点之间都是相互可达的，则称G是强连通图(Strongly Connected Graph)。

例9.3.6

判断下图中4个图的连通性。



分析 先看略去图中所有有向边的方向得无向图，容易看出G_1、G_2、G_3是连通的有向图，G_4是非连通的有向图。再看有向连通图中结点间的可达情况：G_1中v_1到v_4，不可达，v_4到v_1也不可达，所以G_1是弱连通图，G_2中任何一对结点之间至少有一个结点到另一个结点是可达的，所以G_2是单向连通图（当然它也是弱连通图）；G_3中任何一对结点之间都是可达的，所以G_3是强连通图（当然它也是单向连通图和弱连通图）。

解 在上图中，G_1是弱连通图，G_2是单向连通图（当然它也是弱连通图），G_3是强连通图（当然它也是单向连通图和弱连通图），G_4是非连通的有向图。

定理9.3.7

有向图G是强连通图的充分必要条件是G中存在一条经过所有结点的回路。

必要性 充分性 [illegible] 可达的，所以v_i到v_{i+1}存在通路，i=1,2,…,n-1，且v_n到v_1存在通路。让这些连通首尾相接，则得一回路C。显然所有结点均在该回路中出现。

利用A和P判断有向图的连通性

1. 有向线图G是强连通图当且仅当它的可达性矩阵P的所有元素均为1；
2. 有向线图G是单向连通图当且仅当它的可达性矩阵P及其转置矩阵P^T经过布尔并运算后所得的矩阵$P' = P \vee P^T$中除主对角元外其余元素均为1；
3. 有向线图G是弱连通图当且仅当它的邻接矩阵A及其转置矩阵A^T经布尔并运算所得的矩阵$A' = A \vee A^T$作为邻接矩阵而求得的可达性矩阵P’中所有元素均为1。

定义9.3.6

在有向图G = <V, E>中，设G’是G的子图，如果

（1）G’是强连通的（单向连通的、弱连通的）；

（2）对任意$G'' \subseteq G$，若$G' \subset G''$，则G"不是强连通的（单向连通的、弱连通的）；

那么称G’为G的强连通分支（单向连通分支、弱连通分支）（Strongly/Unilaterally/weakly Connected Component），或称为强分图（单向分图、弱分图）。

注

1. 如果不考虑边的方向，弱连通分支对应相应的无向图的连通分支。
2. 注意把握（强、单向、弱）连通分支的**极大性特点**，即任意增加一个结点或一条边就不是（强、单向、弱）连通的了。

3、无向赋权图的最短通路

在赋权图中，边的权也称为边的长度，一条通路的长度指的就是这条通路上各边的长度之和。从结点v_i到v_j的长度最小的通路，称为v_i到v_j的最短通路。

（1）求给定两结点间的最短通路—Dijkstra算法

如何求出简单无向赋权图G = <V, E>中从结点v_1到v_n的最短通路，目前比较好的算法是由Dijkstra在1959年提出的，称为Dijkstra算法，其基本思想是：

将结点集合V分为两部分：一部分称为具有P（永久性）标号的集合，另一部分称为具有T（暂时性）标号的集合。所谓结点v的P标号是指从v_1到v的最短通路的长度；而结点u的T标号是指从v_1到u的某条通路的长度。首先将v_1取为P标号，其余结点为T标号，然后逐步将具有T标号的结点改为P标号。当结点v_n也被改为P标号时，则找到了从v_1到v_n的一条最短通路。

算法9.3.1 Dijkstra算法

I. 初始化：将v_1置为P标号，$d(v_1)=0$，$P=\{v_1\}$，$v_i\in V$，$i\neq 1$，置v_i为T标号，即$T=V-P$且

$$d(v_i)=\begin{cases} w(v_1,v_i), & 若(v_1,v_i)\in E \\ \infty & 若(v_1,v_i)\notin E \end{cases}$$

II. 找最小：寻找具有最小值的T标号的结点。若为v_k，则将v_k的T标号改为P标号，且$P=P\cup\{v_k\}$，$T=T-\{v_k\}$。

算法9.3.1 Dijkstra算法(续)

III. 修改：修改与v_k相邻的结点的T标号值。$v_i \in V$,

$$d(v_i)=\begin{cases} d(v_k)+w(v_k,v_i), & 若 d(v_k)+w(v_k,v_i)<d(v_i) \\ d(v_i) & 否则 \end{cases}$$

IV. 重复(II)和(III)，直到v_n改为P标号为止。

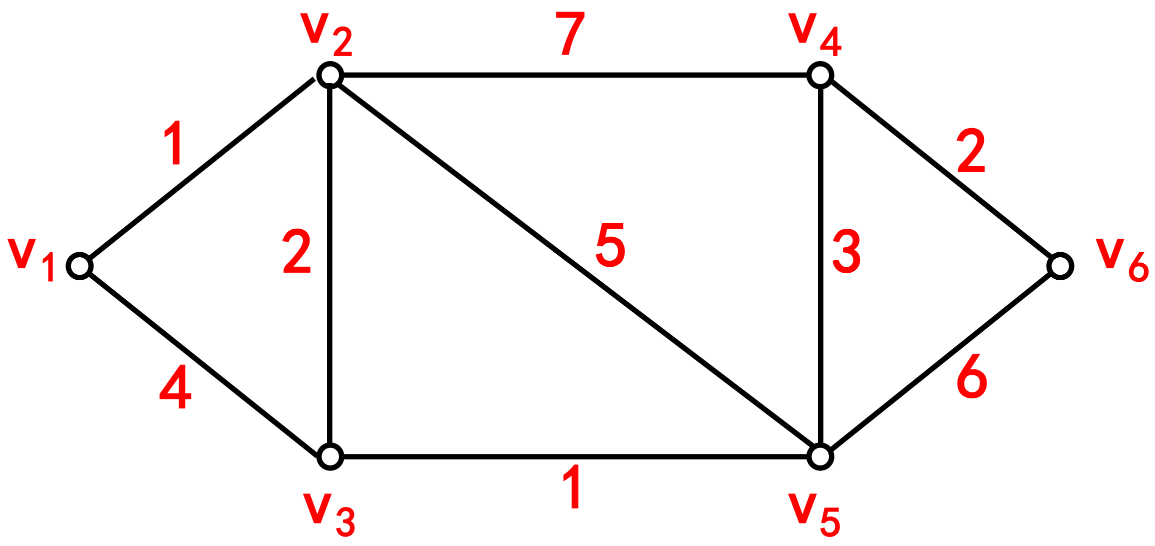
说明

当v_n归入P而正好P = V时，不仅求出了从v_1到v_n的最短通路，而且实际上求出了从v_1到所有结点的最短通路。

上述算法的正确性是显然的。因为在每一步，设P中每一结点的标号是从v_1到该结点的最短通路的长度（开始时，P = {v_1}，d(v_1) = 0，这个假设是正确的），故只要证明上述d(v_i)是从v_1到v_i的最短通路的长度即可。事实上，任何一条从v_1到v_i通路，若通过T的第一个结点是v_p，而$v_p \neq v_i$的话，由于所有边的长度非负，则这种通路的长度不会比d(v_i)小。

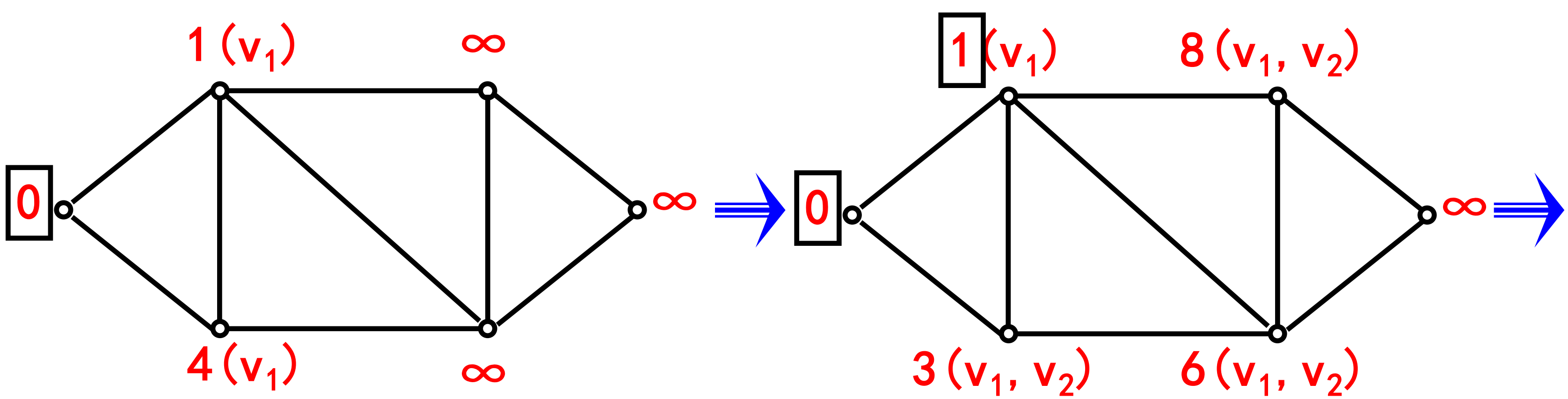
例9.3.9

试求简单无向赋权图中v_1到v_6的最短通路。

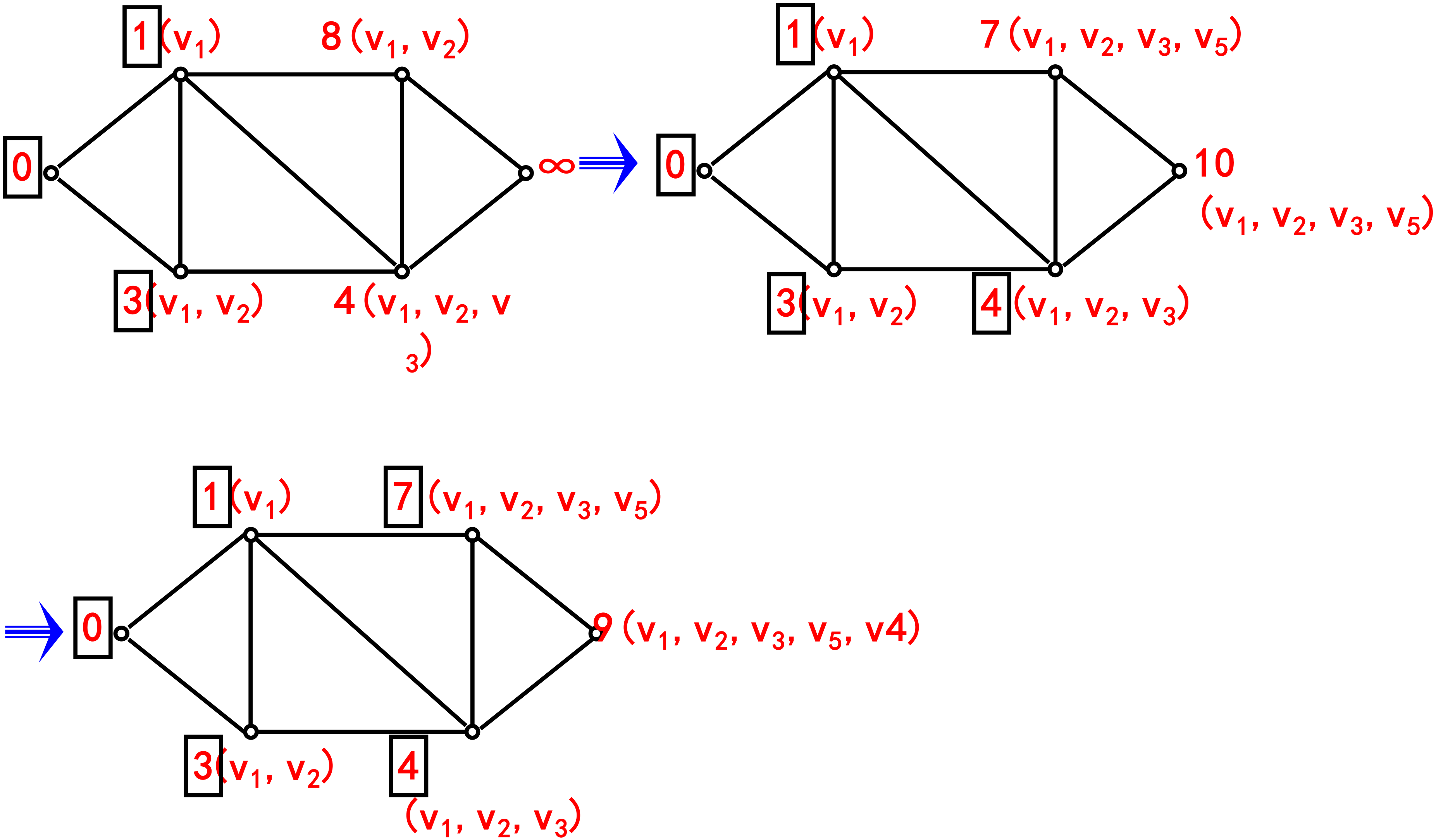


例9.3.9 解

根据Dijkstra算法，有如下图所示的求解过程。故v_1到v_6的最短通路为$v_1v_2v_3v_5v_4v_6$，其长度为9。实际上，也求出了v_1到所有结点的最短通路，如v_1到v_5的最短通路为$v_1v_2v_3v_5$，其长度为4，等等。



例9.3.9 解（续）



1 (v1)
8 (v1, v2)
0
∞
3 (v1, v2)
4 (v1, v2, v3)
1 (v1)
7 (v1, v2, v3, v5)
0
10
(v1, v2, v3, v5)
3 (v1, v2)
4 (v1, v2, v3)
1 (v1)
7 (v1, v2, v3, v5)
0
9 (v1, v2, v3, v5, v4)
3 (v1, v2)
4
(v1, v2, v3)

（2）求任意两结点间的最短通路——Floyd算法

算法9.3.2 Floyd算法：

从矩阵$D^{(0)}=(w_{ij})_{n\times n}$（这里$w_{ij}=w(v_i,v_j)$，称为图的长度矩阵）开始，依次构造出n个矩阵$D^{(1)}$、$D^{(2)}$、…、$D^{(n)}$，这里n为图中结点的个数。第k个矩阵$D^{(k)}=(d_{ij}^{(k)})$的元素$d_{ij}^{(k)}$表示从结点$v_i$到$v_j$而中间结点仅属于$v_1$到$v_k$的k个结点的所有通路中的最短通路长度。

若已知$D^{(k-1)}=(d_{ij}^{(k-1)})$，则$D^{(k)}=(d_{ij}^{(k)})$的元素规定为

$$d_{ij}^{(k)}=\min\{d_{ij}^{(k-1)},d_{ik}^{(k-1)}+d_{kj}^{(k-1)}\}$$

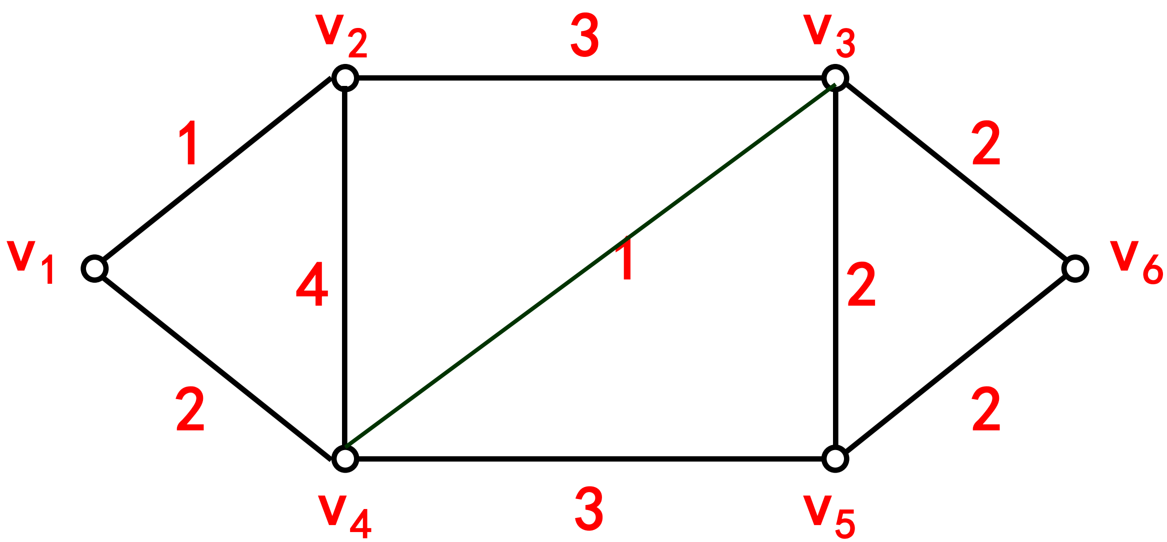
Floyd算法

运算过程从k = 1开始，让i和j分别取遍从1到n的所有值，然后k增加1，如此反复进行，直到k = n为止。这时$D^{(n)} = (d_{ij}^{(k)})$的元素就是从v_i到v_j的最短通路长度。

算法的正确性是显然的。Floyd算法算法求出了任意两个结点间的最短通路的长度，从而很容易得出相应的最短通路。

例9.3.10

试求简单无向赋权图9.3.13中的所有最短通路。

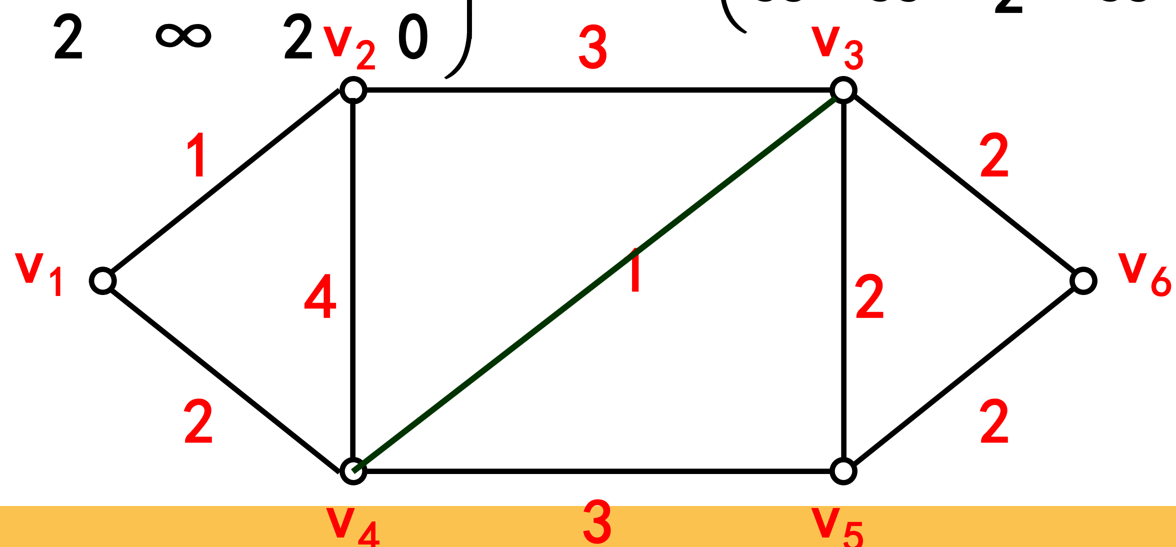


例9.3.10 解

根据Floyd算法，有：

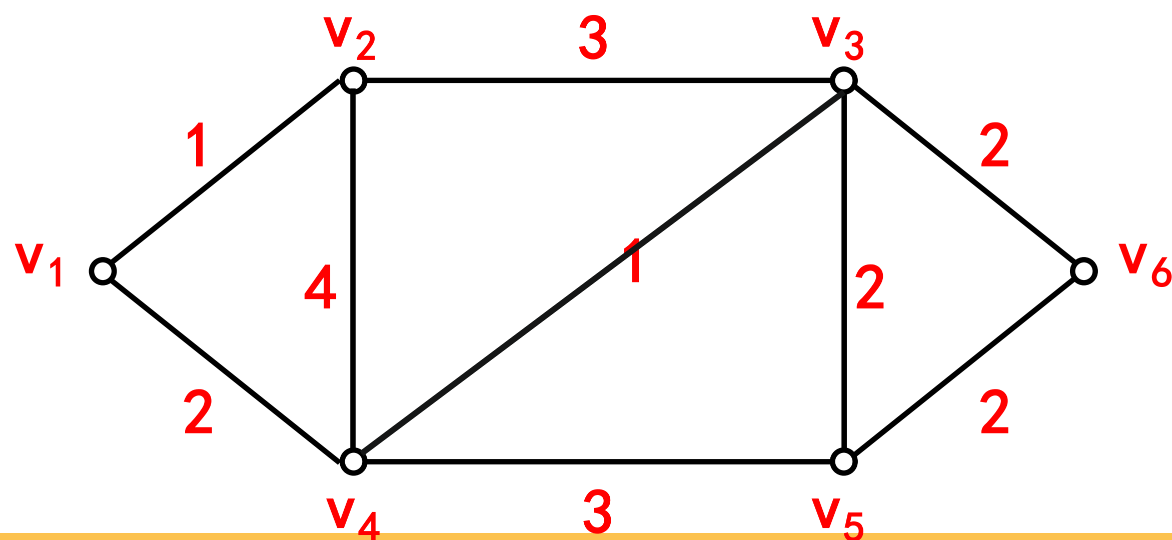
$$D^{(0)}=\begin{pmatrix} 0 & 1 & \infty & 2 & \infty & \infty \\ 1 & 0 & 3 & 4 & \infty & \infty \\ \infty & 3 & 0 & 1 & 2 & 2 \\ 2 & 4 & 1 & 0 & 3 & \infty \\ \infty & \infty & 2 & 3 & 0 & 2 \\ \infty & \infty & 2 & \infty & 2 & 0 \end{pmatrix}$$

$$D^{(1)}=\begin{pmatrix} 0 & 1 & \infty & 2 & \infty & \infty \\ 1 & 0 & 3 & 3 & \infty & \infty \\ \infty & 3 & 0 & 1 & 2 & 2 \\ 2 & 3 & 1 & 0 & 3 & \infty \\ \infty & \infty & 2 & 3 & 0 & 2 \\ \infty & \infty & 2 & \infty & 2 & 0 \end{pmatrix}$$



例9.3.10 解(续)

$$D^{(2)} = \begin{pmatrix} 0 & 1 & 4 & 2 & \infty & \infty \\ 1 & 0 & 3 & 3 & \infty & \infty \\ 4 & 3 & 0 & 1 & 2 & 2 \\ 2 & 3 & 1 & 0 & 3 & \infty \\ \infty & \infty & 2 & 3 & 0 & 2 \\ \infty & \infty & 2 & \infty & 2 & 0 \end{pmatrix} \qquad D^{(3)} = \begin{pmatrix} 0 & 1 & 4 & 2 & 6 & 6 \\ 1 & 0 & 3 & 3 & 5 & 5 \\ 4 & 3 & 0 & 1 & 2 & 2 \\ 2 & 3 & 1 & 0 & 3 & 3 \\ 6 & 5 & 2 & 3 & 0 & 2 \\ 6 & 5 & 2 & 3 & 2 & 0 \end{pmatrix}$$



例9.3.10 解(续)

$$D^{(4)}=D^{(5)}=D^{(6)}=\begin{pmatrix} 0 & 1 & 3 & 2 & 5 & 5 \\ 1 & 0 & 3 & 3 & 5 & 5 \\ 3 & 3 & 0 & 1 & 2 & 2 \\ 2 & 3 & 1 & 0 & 3 & 3 \\ 5 & 5 & 2 & 3 & 0 & 2 \\ 5 & 5 & 2 & 3 & 2 & 0 \end{pmatrix}$$

故v_2到v_6的最短通路长度为5，其最短通路为$v_2v_3v_6$，其余类似。

9.4 本章总结

1 主要知识点汇集

2 习题类型

3 解题分析和方法

1、主要知识点汇集

① 图的概念：图的定义、图的表示、图的操作、邻接点与邻接边、图的分类等。

② 图的基本性质：结点的度数、图的基本定理（握手定理）、图的同构、完全图、补图、子图、真子图、生成子图、导出子图等。

③ 通路与回路：通路与回路、简单（基本）通路与简单（基本）回路、通路与回路长度、结点间的短程线和距离、可达与可达性矩阵。

④ 图的连通性：无向连通图与连通分支、强（单向、弱）连通图与强（单向、弱）分图、利用邻接矩阵和可达性矩阵判断图的连通性。

⑤ 图的应用：通讯网络、渡河问题、均分问题、最短通路算法。

2、习题类型

① 基本概念题：主要观测点在于图的基本概念、分类与判断；

② 判断题：主要观测点在于判定图的连通性；

③ 计算题：主要观测点在于结点的度数、连通分支、通路数目等；

④ 证明题：主要观测点在于图的同构、结点的度数等的证明

3、解题分析和方法

① 图的集合、图形、矩阵3种表示方式在同构的意义下是惟一的；

② 图的邻接矩阵既描述了图中结点间的邻接关系，还可以利用它来计算结点间的通路数目、判断图的连通性；

③ 判断两个图同构，只能依据同构的定义，构造两个结点集之间的双射函数，没有简单的方法；

④ 在计算和证明与结点的度数有关的问题时，经常使用握手定理；

⑤ 反证法非常有用，特别是在证明惟一性和不存在的时候。

第六次作业（第九章课后习题）

第273—276页：

3	5	8
14	15	21
23	25	27

Thank You！

http://202.115.21.136:8080/lssx