**图论及应用章节学习指导**

**总体学习原则：以教材为主，参考各章课件进行学习；对定理、性质、推论的深奥复杂证明不要求，可忽略。**

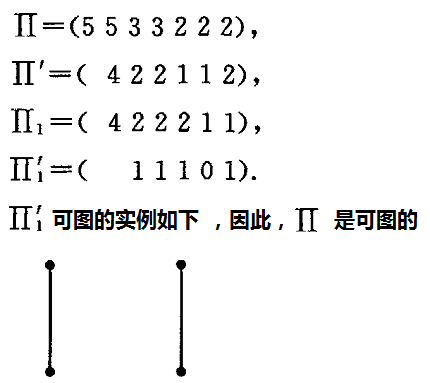
**第一章 图的基本概念**

该章基本概念与定义很重要，必须掌握，是以后各章节学习的基础。邻接矩阵、关联矩阵要求学。

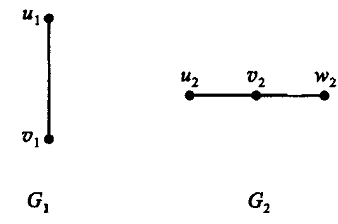
该章p17之后的内容，不要求学。

本章难点，给大家进一步解释如下。

**一、P5，一组整数序列是不是可图的例子：**

****

**二、p8页，定义4，积图概念。**

****

**（一）、要求G1、G2的积图，G1在前，G2在后，分为以下几步：**

1.G1图在积运算的前面，因此分布以G1的两个顶点u1、v1分别与G2的三个顶点配对。点对的位置不一定按照如下方式放，可以随意，连接出来的图都是同构的。

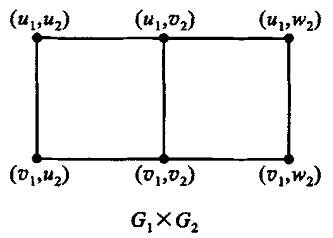
（u1，u2） （u1,v2） （u1,w2）

（v1,u2） （v1,v2） （v1,w2）

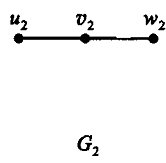
2.然后按照定义进行连接。注意邻接概念，由一条边直接连接的两个点是邻接。在G1，u1与v1邻接；在G2，u2、v2邻接，v2、w2邻接，u2、w2不邻接。

3.连接：（1）当点对（u1，u2）、（u1,v2）中第一个点相同，而第二个点相邻时，将（u1，u2）、（u1,v2）用一条线连接起来；如第二个点不邻接，则不连接。

（2）当（u1，u2）、（v1,u2）这一对中，第二个点相同，第一个邻接时，将（u1，u2）、（v1,u2）用一条线连接起来；如第一个点不邻接，则不连接。最后积图如下。

****

**（二）、要求G2、G1的积图，G2在前，G1在后，分为以下几步：**

** **

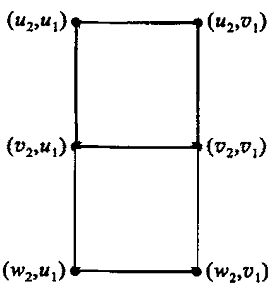
1.将G2的三个顶点分别与G1的三个顶点配对。点对的位置不一定按照如下方式放，可以随意，连接出来的图都是同构的。

（u2,u1） （u2,v1）

（v2,u1） （v2,v1）

(w2,u1) （w2，v1）

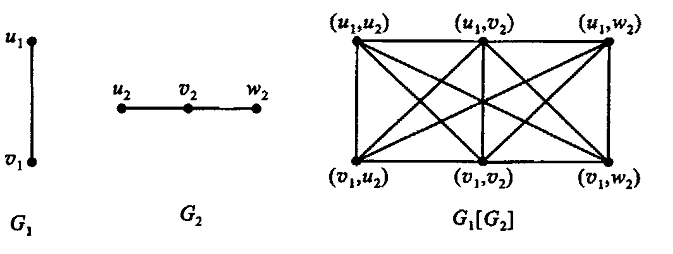
2.连接：（1）当点对（u2，u1）、（u2,v1）中第一个点相同，而第二个点相邻时，将（u2，u1）、（u2,v1）用一条线连接起来；（2）当（u2，u1）、（v2,u1）这一对中，第二个点相同，第一个邻接时，将（u1，u2）、（v1,u2）用一条线连接起来；如第一个点不邻接，则不连接。最后积图如下。

****

为什么（u2,u1）、（w2,v1）没连接，因为这一对点不符合连接条件；为什么（u2,u1）、（w2,u1）不连接，因为虽然后两点u1相同，但u2、w2不邻接，所以不能连接。

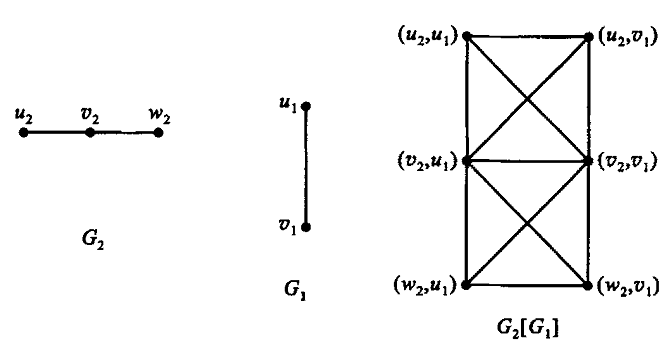
**（三）、合成图概念与积图相似，连接要求：当每对点中，分别取出第一个点，当这两个第一个点，如果邻接就连接；或相同时，再判断每对中第二个点，当这两个第二个点邻接时，连接这对点。**

**1.G1与G2的合成图如下：顶点配对方法与积图相同**

****

为什么（u1，u2）与（v1，w2）连接？因为u1与v1邻接；只要第一个点邻接就连接，或者第一个点相等第二个点邻接就连接。

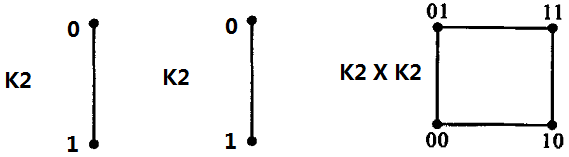
**2.G2与G1的合成图如下：顶点配对方法与积图相同**

****

为什么（u2，u1）与（v2，v1）连接？因为点对中的第一个点u2与v2邻接，根据定义，第一个点邻接就连接；如果第一个相同，第二个点邻接，就连接。只要第一个点邻接就连接，或者第一个点相等第二个点邻接就连接。

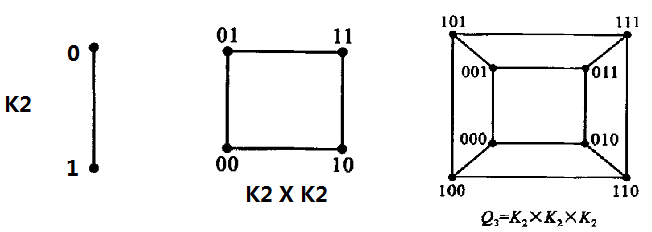
为什么（u2，u1）与（w2，v1）不连接？因为点对中的第一个点u2与w2不邻接，根据定义，第一个点不邻接就不连接，即使第二个点邻接，右边链接；如果第一个相同，第二个点邻接，就连接。只要第一个点邻接就连接，或者第一个点相等第二个点邻接就连接。

**（四）、方体的积图:由两个K2的积逐次形成**

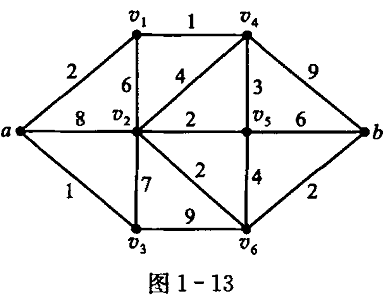
****

K2就是一条线段，顶点用0、1编号；两个K2就是两条线段，顶点分别用0、1编号；用第一个K2的顶点0、1编号，分别与第二个K2的0、1编号配对，两条K2线段 0、1编号的组合总共有4种（0,0）、（0,1）、（1,0）、（1,1），将这种表示方式简化为00、01、10、11，然后按积图的定义进行连接，连接后结果就是K2 X K2。

然后，再用一个K2与K2 X K2求积图得到K2 X K2 X K2，依次类推。如果让Q1=K2，Q2= K2 X Q1,…,Qn=K2 X Q(n-1).

****

**三、P11最短路Dantjig算法复杂难懂，只要把P12页的例子里的算法过程搞清楚就可以了。**

****

算法过程如下：（1）从a点开始，先将a加入已知点集A，再找与a直接连接的点中权值最小的点v3，并将其加入已知点集A；（2）在未进入点集A的点集中，找分别与点集A中的点a、v3直接相连的中的最小权值点v1，再将v1加入到点集A中；（3）逐次进行这个过程，直到所有点都加入到A几何为止；（4）在点集A中找到一条从a到b的最小权值的路，即为Dantjig算法的最短路。

有几个点就有几轮循环

\* 每轮选择距离最近的点，去更新其它点

\* 重复n次，就能使得每一个点都被选取到

遍历选出当前没被访问过，且距离原点最近的点

用这个距离最近的点，去更新其它的点距离原点的距离