涩图

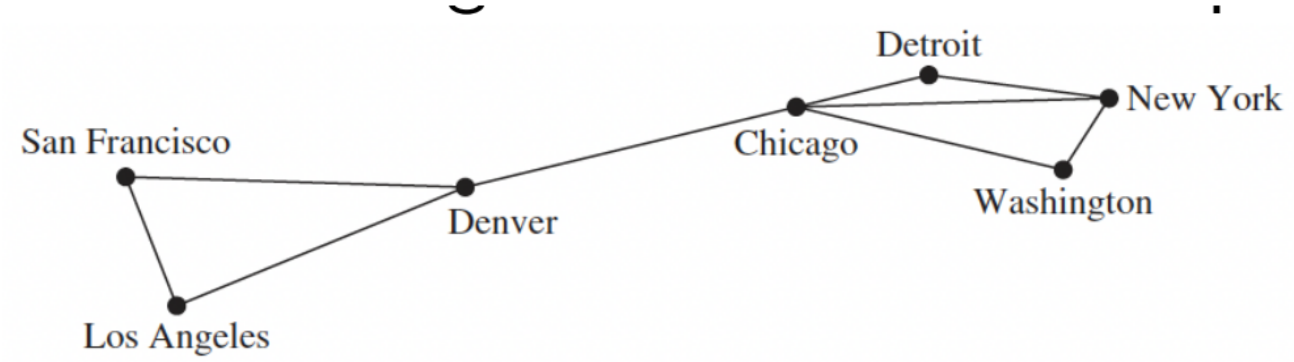
- A graph consists of a set of vertices V, $|V| = n$
- and a set of edges E, $|E| = m$
- Each edge has two **endpoints**
- An edge **joins** its endpoints, two vertices are **adjacent** if they are joined by an edge
- When a vertex is an endpoint of an edge, we say that the edge and the vertex are **incident** to each other.

Simple Graph, Multigraph, Pseudograph

simple graph (简单图)

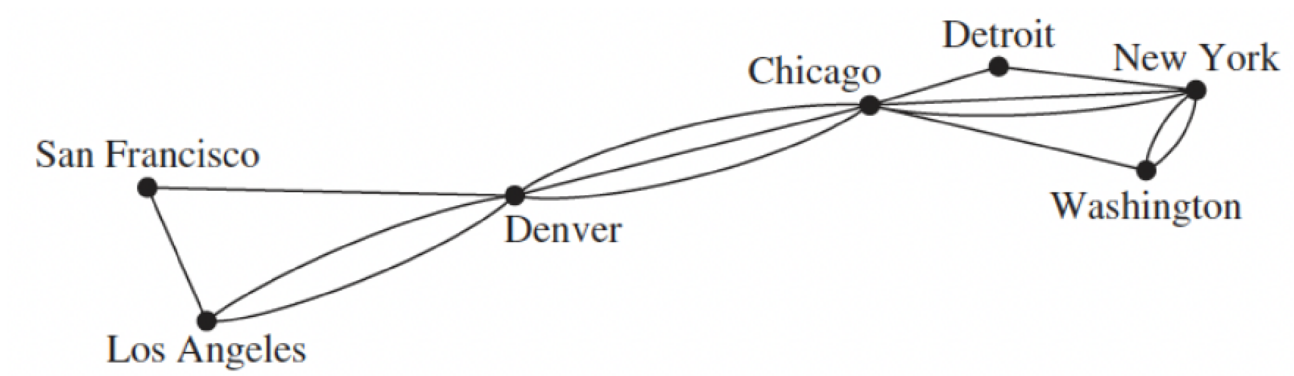
A graph in which each edge connects two **different** vertices and where **no** two edges connect the same pair of vertices.

意思是说，简单图中没有重复的边，也没有自环。



multigraph (多重图)

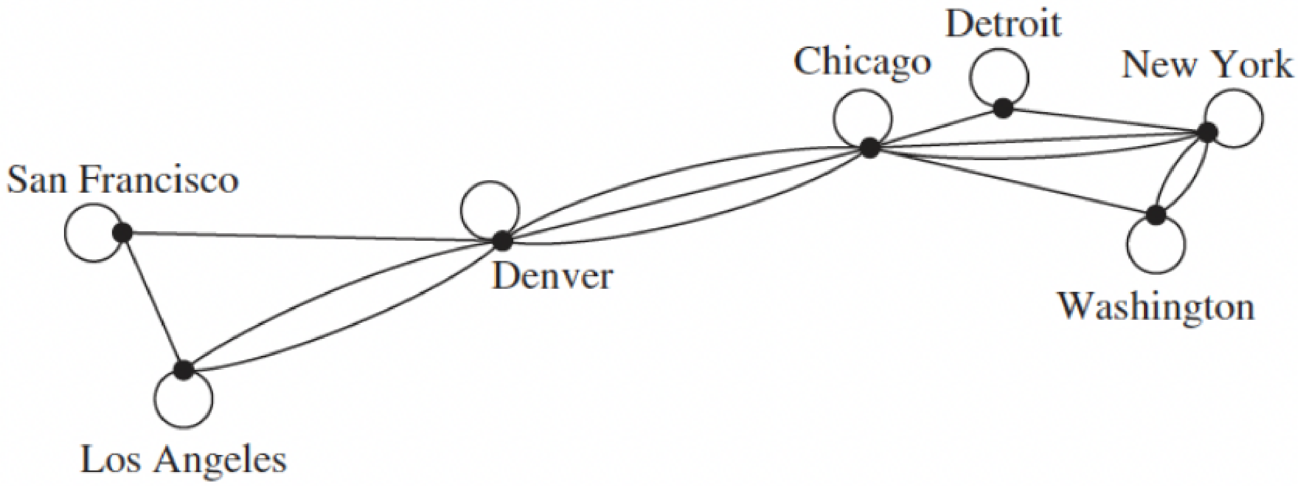
Graphs that may have **multiple edges** connecting the same vertices.
可能存在多条边连接相同顶点的图。



pseudograph (伪图)

Graphs that may include **loops**, and possibly multiple edges connecting the same pair of vertices or a vertex to itself.

可能存在自环，也可能存在多条边连接相同顶点的图。



Undirected Graph

一些定义和记号

- 如果两个顶点之间存在边，那么这两个顶点是**adjacent**的或者说是**neighbors**
- $N(v)$: 如果v是$G=(V,E)$中的一个顶点，那么$N(v)$是与v相邻的顶点的集合。
- $N(A)$: 如果A是$G=(V,E)$的一个子集，那么$N(A)$是与A中的顶点相邻的顶点的集合。
- $deg(v)$: 无向图的**degree**是指与v相邻的顶点的个数，但是一个环对degree的贡献是2。

Theorem (Handshaking Theorem)

If $G=(V,E)$ is an **undirected** graph with m edges, then

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

如果一个无向图有m条边，那么这个图中所有顶点的度数之和为$2m$（即使是有多重边或自环的图）。

Theorem

An undirected graph has an **even** number of vertices of **odd** degree.
一个无向图中，度数为奇数的顶点的个数为偶数。
证明:
假设V_{odd}是所有度数为奇数的顶点的集合，V_{even}是所有度数为偶数的顶点的集合，那么

$$2m = \sum_{v \in V} deg(v) = \sum_{v \in V_{odd}} deg(v) + \sum_{v \in V_{even}} deg(v)$$

由于$2m$是偶数，$\sum_{v \in V_{even}} deg(v)$也是偶数，所以$\sum_{v \in V_{odd}} deg(v)$必须也是偶数，而$\sum_{v \in V_{odd}} deg(v)$是所有度数为奇数的顶点的度数之和，所以度数为奇数的顶点的个数为偶数。

Directed Graph

一些定义和记号

- 每一条边都是一个有序对(u, v)，这条边的方向是从u指向v

- 假设(u, v)是$G = (V, E)$中的一条边，那么u是**initial vertex**并且**adjacent to** v，v是**terminal vertex**并且**adjacent from** u
- $deg^-(v)$: **in-degree** of v，指向v的边的条数
- $deg^+(v)$: **out-degree** of v，从v出发的边的条数
- 环对in-degree和out-degree的贡献都是1

Theorem

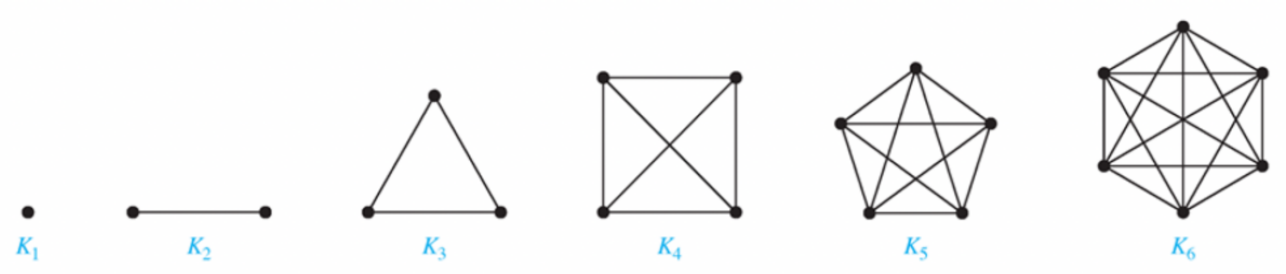
Let $G = (V, E)$ be a graph with directed edges. Then,

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

有向图的边的条数等于所有顶点的in-degree之和，也等于所有顶点的out-degree之和。

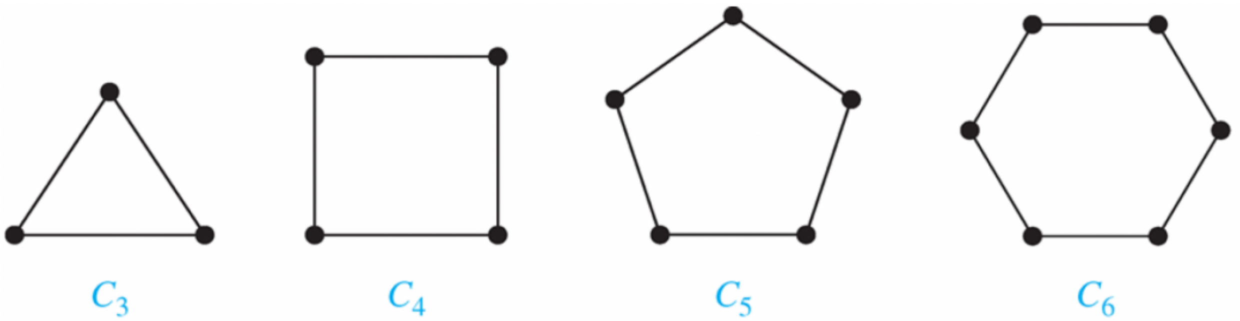
Complete Graphs (完全图)

A **complete graph** on n vertices, denoted by K_N, is the simple graph that contains exactly one edge between each pair of distinct vertices.
complete graph是一个简单图，它的任意两个顶点之间都有一条边。



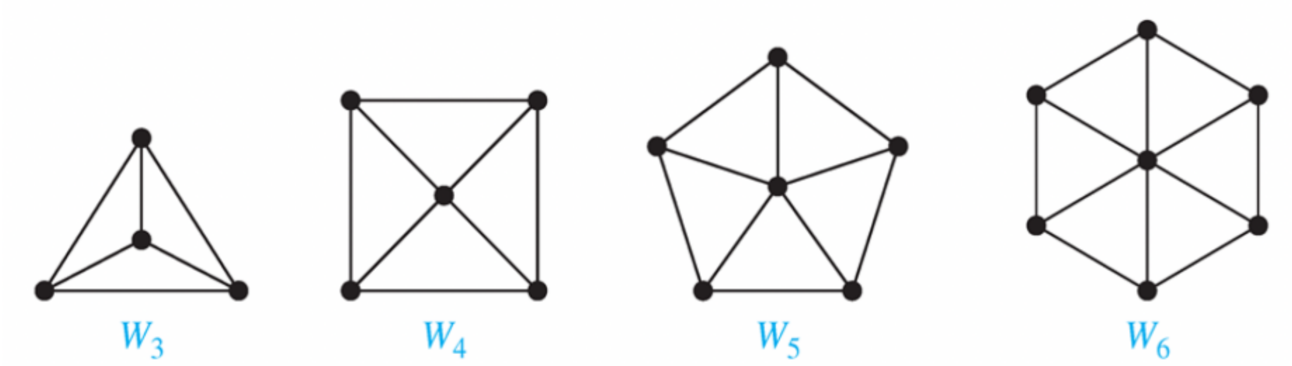
Cycles

A **cycle** C_n for $n \geq 3$ consists of n vertices $v_1, v_2, \cdots, v_n$, and edges $\{v_1, v_2\}, \{v_2, v_3\}, \cdots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.
cycle包含n个顶点$v_1, v_2, \cdots, v_n$，以及n条边 $\{v_1, v_2\}, \{v_2, v_3\}, \cdots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$。



Wheels

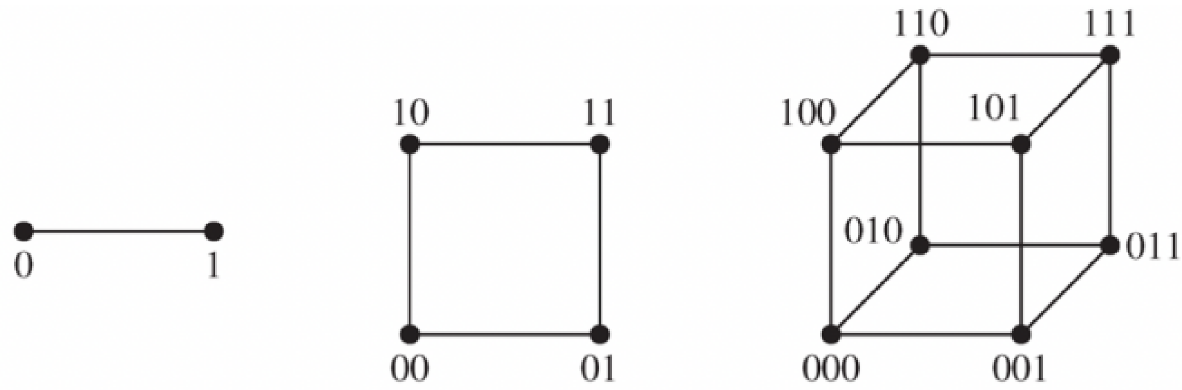
A **wheel** W_n is obtained by adding an additional vertex to a cycle C_n.
wheel是在一个cycle C_n上再加一个顶点。



N-dimensional Hypercube

An **n-dimensional hypercube**, or **n-cube**, Q_n is a graph with 2^n vertices representing all bit strings of length n, where there is an edge between two vertices that differ in exactly one bit position.

n-dimensional hypercube或者n-cube是一个图，它有2^n个顶点，每个顶点代表一个长度为n的bit string，如果两个顶点的bit string只有一个bit不同，那么这两个顶点之间有一条边。



有多少条边:

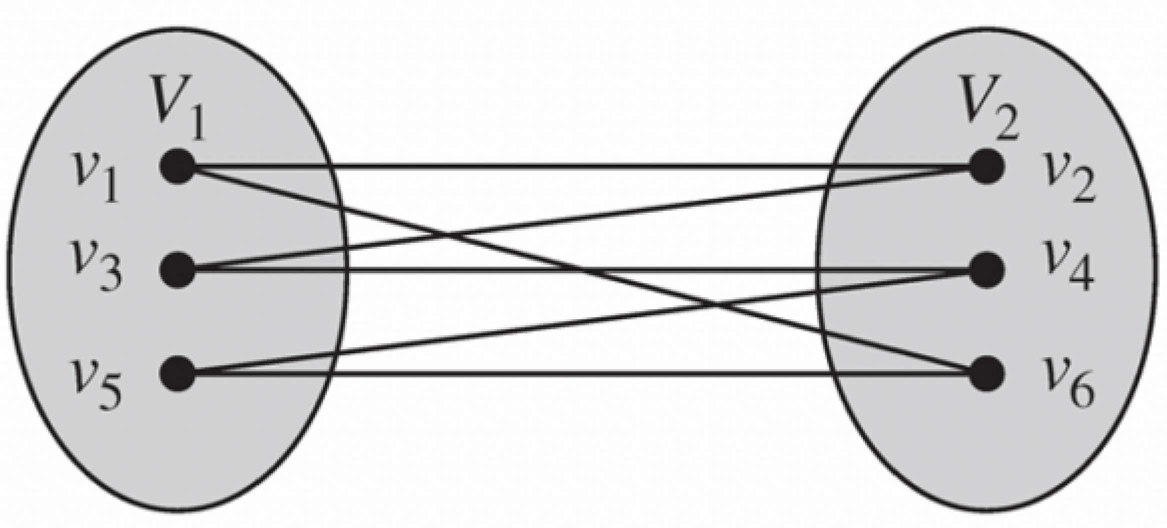
为了从n-cube Q_n构造$(n+1)$-cube Q_{n+1}，我们需要两个Q_n，一个Q_n的顶点的标签前面加一个0，另一个Q_n的顶点的标签前面加一个1，然后把两个Q_n中标签只有第一位不同的顶点之间加一条边。

Bipartite Graphs (二分图)

Definition: A simple graph G is bipartite if V can be partitioned into two disjoint subsets V_1 and V_2 such that every edge connects a vertex in V_1 and a vertex in V_2.

An equivalent definition of a bipartite graph is a graph where it is possible to color the vertices red or blue so that no two adjacent vertices are of the same color.

定义：一个简单图G是二分图，如果V可以被划分为两个不相交的子集V_1和V_2，并且每条边都连接一个V_1中的顶点和一个V_2中的顶点。
一个等价的定义是，如果可以把图中的顶点染成红色和蓝色，使得任意两个相邻的顶点颜色不同，那么这个图是二分图。

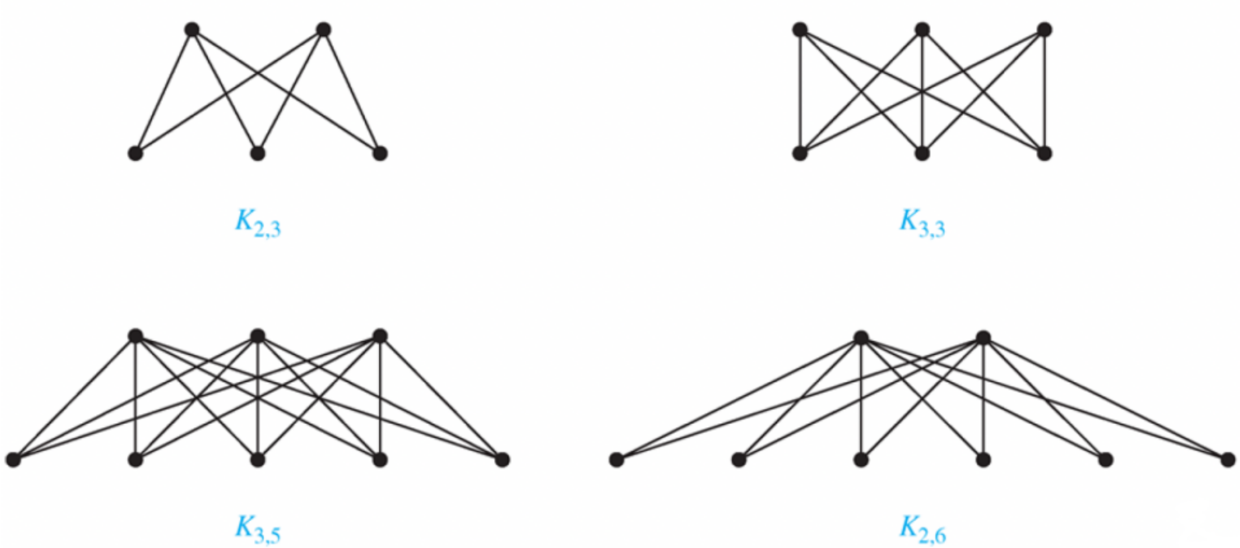


Bipartite and not bipartite

- **Bipartite**: Its vertex set is the union of two disjoint sets, $\{a, b, d\}$ **and** $\{c, e, f, g\}$, and each edge connects a vertex in one of these subsets to a vertex in the other subset.
- **Not bipartite**: Its vertex set cannot be partitioned into two subsets so that edges do not connect two vertices from the same subset.

Complete Bipartite Graphs (完全二分图)

Definition: A **complete bipartite graph** $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets V_1 of size m and V_2 of size n such that there is an edge from **every** vertex in V_1 to **every** vertex in V_2.
定义：一个**完全二分图**$K_{m,n}$是一个图，它的顶点集被划分为两个子集V_1和V_2，V_1的大小为m，V_2的大小为n，并且V_1中的每个顶点都与V_2中的每个顶点都有一条边。



$K_{2,3}$ $K_{3,3}$ $K_{3,5}$ $K_{2,6}$

Bipartite Graphs and Matchings

Matching是指把一个集合中的元素和另一个集合中的元素匹配起来。一个matching是边集的一个子集，使得任意两条边都不与同一个顶点关联。换句话说，一个matching是边集的一个子集，使得如果$\{s, t\}$和$\{u, v\}$是matching的两条边，那么s, t, u, v都是不同的。

Job assignments：顶点代表工作和员工，边连接员工和他们被训练过的工作。一个常见的目标是把工作分配

给员工，使得完成的工作最多。

A **maximum matching** is a matching with the **largest number of edges**.
一个**maximum matching**是一个matching，它的边数最多。

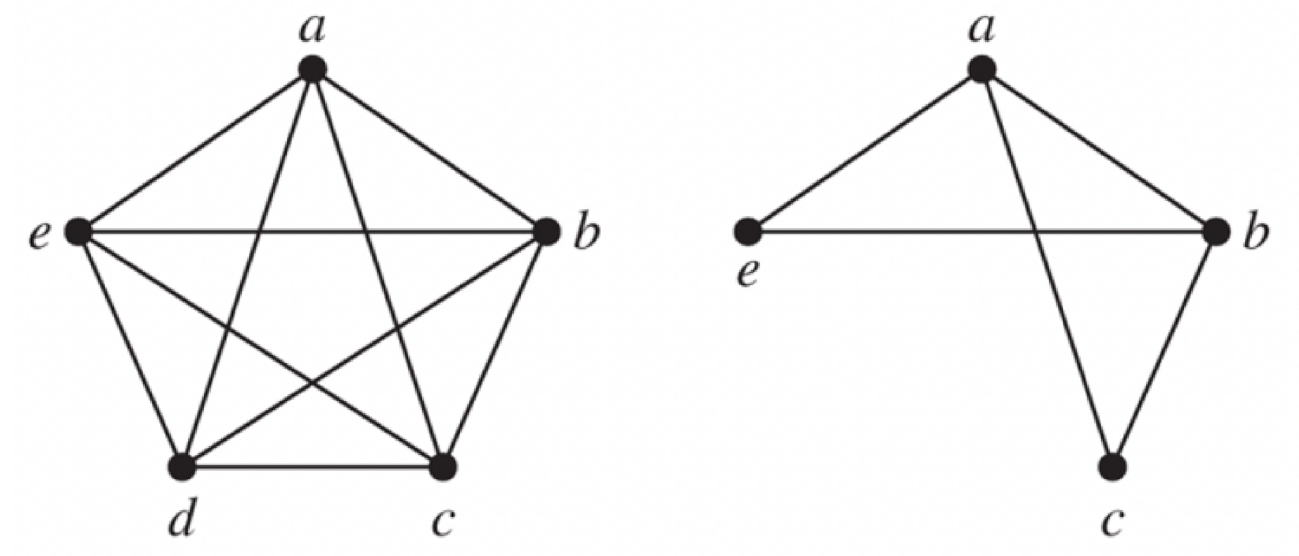
A matching M in a bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) is a **complete matching** from V_1 to V_2 if every vertex in V_1 is the endpoint of an edge in the matching, or equivalently, if $|M| = |V_1|$.
一个matching M是一个**complete matching**，如果M是从V_1到V_2的matching，并且V_1中的每个顶点都是M中一条边的端点，或者等价地，如果$|M| = |V_1|$。

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.
Hall's Marriage Theorem：如果一个二分图$G = (V, E)$，它的顶点集被划分为两个子集V_1和V_2，那么G有一个从V_1到V_2的complete matching，当且仅当对于V_1的任意子集A，$|N(A)| \geq |A|$。

Subgraphs

Definition: A **subgraph** of a graph $G = (V, E)$ is a graph (W, F), where $W \subseteq V$ and $F \subseteq E$.

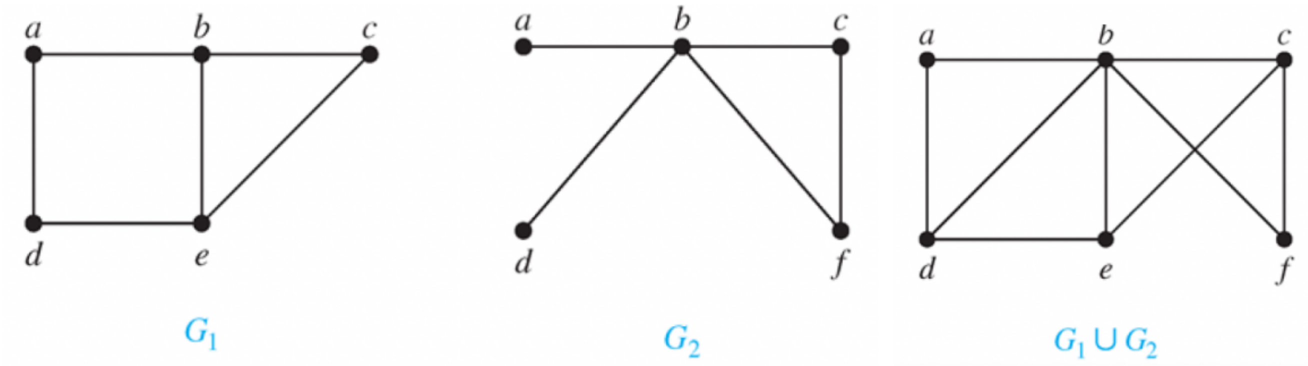
定义：图$G=(V,E)$的一个**subgraph**是一个图(W,F)，其中$W\subseteq V$并且$F\subseteq E$。



Union of Graphs

Definition: The **union** of two simple graphs $G_1=(V_1,E_1)$ and $G_2=(V_2,E_2)$ is the simple graph with vertex set $V_1\cup V_2$ and edge set $E_1\cup E_2$, denoted by $G_1\cup G_2$.

定义：两个简单图$G_1=(V_1,E_1)$和$G_2=(V_2,E_2)$的**union**是一个简单图，它的顶点集是$V_1\cup V_2$，边集是$E_1\cup E_2$，记作$G_1\cup G_2$。



图的表示

- adjacency list (邻接表)
- adjacency matrix (邻接矩阵)
- incidence matrix (关联矩阵)

Adjacency List (邻接表)

定义：adjacency list (邻接表)可以用来表示一个**没有重复边**的图，它指定了每个顶点的邻接顶点。

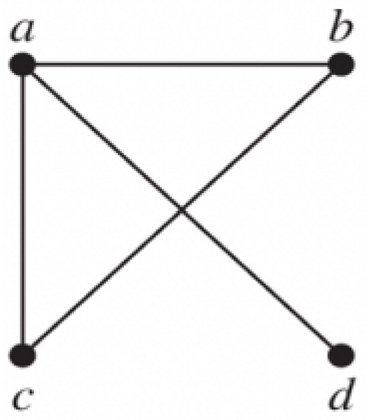
Adjacency Matrix (邻接矩阵)

简单图的邻接矩阵

定义：假设$G=(V,E)$是一个**简单图**，$|V|=n$。任意地把G的顶点列出来，$v_1, v_2, \cdots, v_n$。G的**adjacency matrix** A_G是一个$n \times n$的0-1矩阵，当v_i和v_j是adjacent的时候，A_G的(i,j)位置是1，当v_i和v_j不是adjacent的时候，A_G的(i,j)位置是0。

$$A_G = [a_{ij}], \text{where}$$

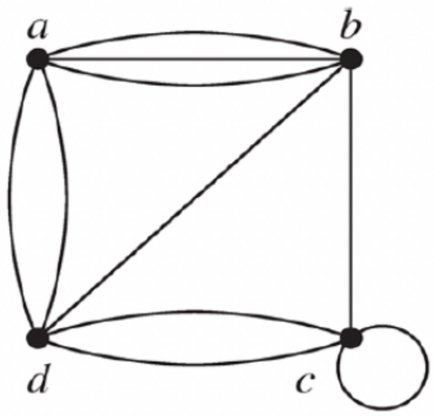
$$a_{ij} = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ are adjacent} \\ 0 & \text{if } v_i \text{ and } v_j \text{ are not adjacent} \end{cases}$$



$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

有环或多重边的图的邻接矩阵

邻接矩阵也可以用来表示有环或多重边的图。邻接矩阵不再是0-1矩阵。a_{ij}的值是v_i和v_j之间的边的条数。

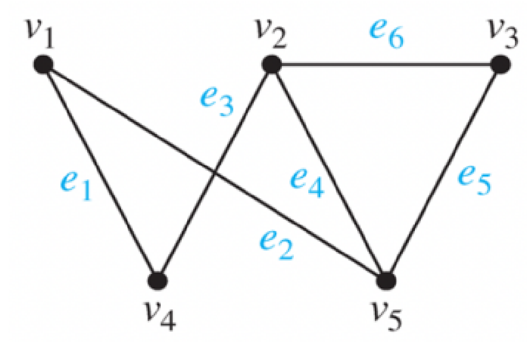


$$\begin{bmatrix} 0 & 3 & 0 & 2 \\ 3 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

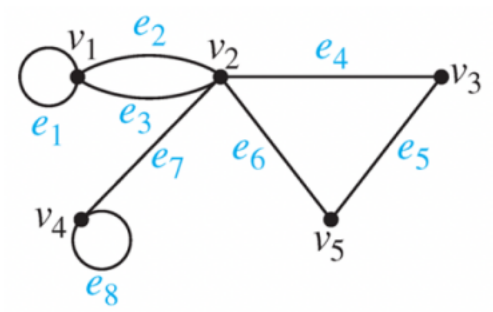
Incidence Matrix (关联矩阵)

定义：假设$G = (V, E)$是一个无向图，顶点$v_1, v_2, \cdots, v_n$，边$e_1, e_2, \cdots, e_m$。G的**incidence matrix** M_G是一个$n \times m$的0-1矩阵$M = [m_{ij}]$，

$$m_{ij} = \begin{cases} 1 & \text{if } e_j \text{ is incident with } v_i \\ 0 & \text{otherwise} \end{cases}$$



$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

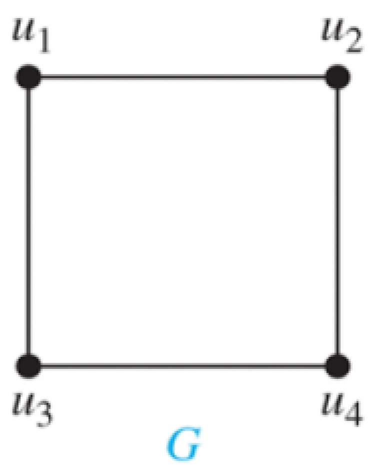


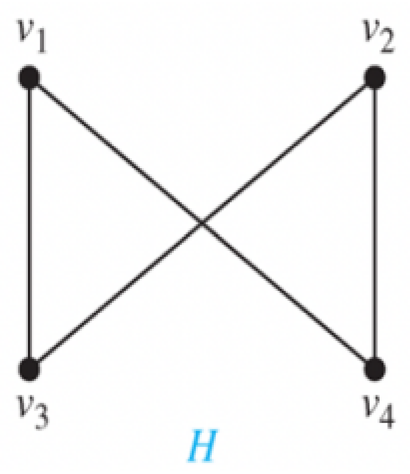
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Isomorphism of Graphs (同构)

定义：简单图$G_1 = (V_1, E_1)$和$G_2 = (V_2, E_2)$是**isomorphic**的，如果存在一个从V_1到V_2的双射，并且满足对于V_1中的任意两个顶点a和b，a和b是adjacent的当且仅当$f(a)$和$f(b)$是adjacent的。这样的函数f被称为**isomorphism**。

举个例子，下面的两个图是isomorphic的





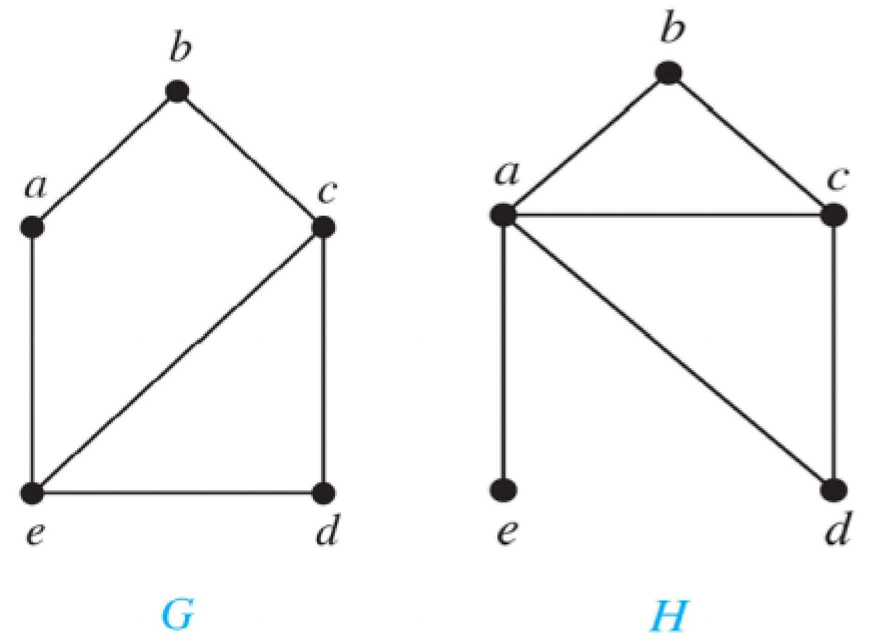
双射函数可以是
$f(u_1) = v_1, f(u_2) = v_4, f(u_3) = v_3, f(u_4) = v_2$。

验证两个图是isomorphic是较为困难的
但是，如果两个图不是isomorphic的，那么可以通过检查两个图的一些**invariants**来证明
常见的invariants有：

- number of vertices (顶点的个数)
- number of edges (边的条数)
- degree sequence (度数序列)
- etc.

例1

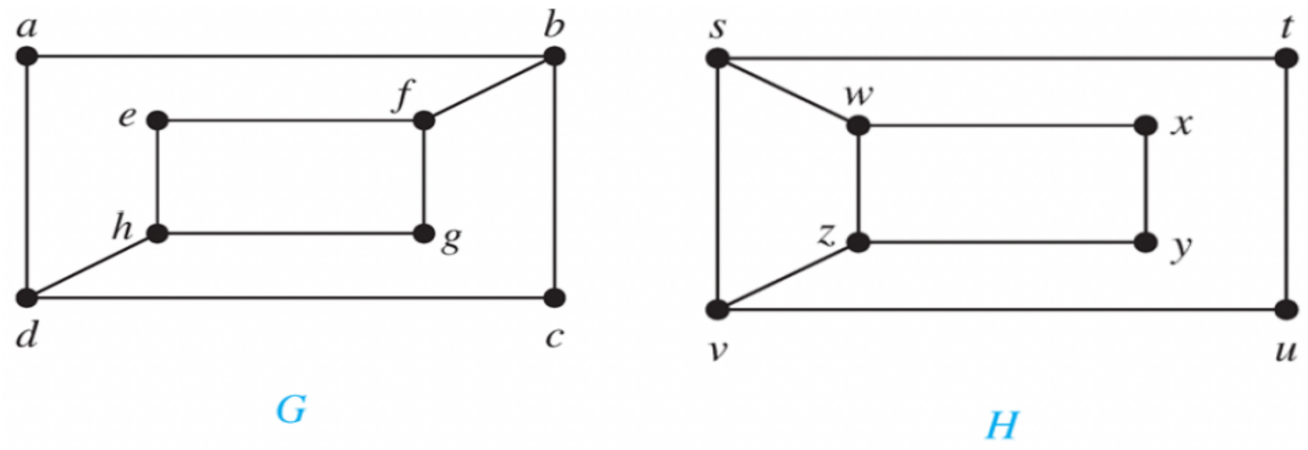
下面两个是不是isomorphic的？



H有度数为1的顶点(e)，而G没有，所以不是isomorphic的。

例2

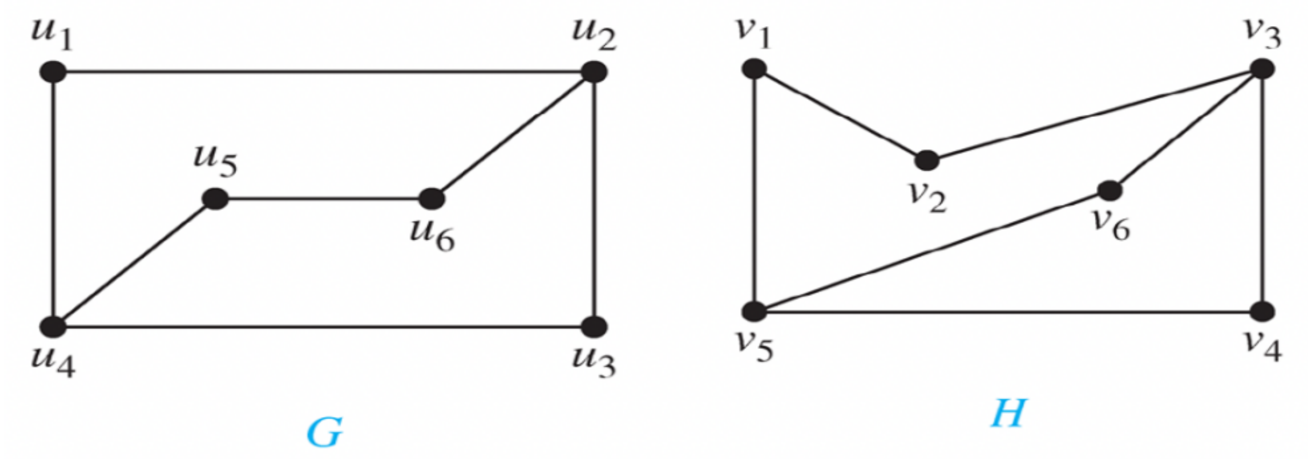
下面两个是不是isomorphic的?



G和H不是isomorphic的。因为G中的顶点a的度数是2，而a必须对应于H中的t, u, x, 或y。然而，H中的这四个顶点中的每一个都与H中的另一个度数为2的顶点相邻，而这对于G中的a来说是不正确的。

例3

下面两个是不是isomorphic的?



$f(u_1) = v_6, f(u_2) = v_3, f(u_3) = v_4, f(u_4) = v_5, f(u_5) = v_1, f($

我们也看看邻接矩阵:

$$\mathbf{A}_H = \begin{array}{c} v_6 \\ v_3 \\ v_4 \\ v_5 \\ v_1 \\ v_2 \end{array} \overset{\begin{array}{cccccc} v_6 & v_3 & v_4 & v_5 & v_1 & v_2 \end{array}}{\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}}. \quad \mathbf{A}_G = \begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{array} \overset{\begin{array}{cccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{array}}{\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}},$$

所以是一个isomorphism。

Path: Undirected Graph

Definition: Let n be a nonnegative integer and G an **undirected** graph. **A path of length n from u to v** in G is a sequence of n edges $e_1, e_2, \cdots, e_n$ of G for which there exists a sequence $x_0 = u, x_1, \cdots, x_{n-1}, x_n = v$ of vertices such that e_i has the endpoints x_{i-1} and x_i for $i = 1, \cdots, n$.
定义：n是一个非负整数，G是一个**无向图**。G中从u到v的**长度为n的path**是一个边的序列$e_1, e_2, \cdots, e_n$，满足存在一个顶点的序列$x_0 = u, x_1, \cdots, x_{n-1}, x_n = v$，使得$e_i$的端点是$x_{i-1}$和$x_i$，$i = 1, \cdots, n$。

一个path被称作**circuit**或cycle，如果它的起点和终点是同一个顶点，但长度大于0。

一个path或者circuit是**simple**的，如果它不包含重复的edge。

path的长度=path里面的边的条数

Path: Directed Graph

Definition: Let n be a nonnegative integer and G an **directed** graph. **A path of length n from u to v** in G is a sequence of n edges $e_1, e_2, \cdots, e_n$ of G for which there exists a sequence $x_0 = u, x_1, \cdots, x_{n-1}, x_n = v$ of vertices such that e_i **is associated with initial vertex x_{i-1} and terminal vertex x_i** for $i = 1, \cdots, n$.
定义：n是一个非负整数，G是一个**有向图**。G中从u到v的**长度为n的path**是一个边的序列$e_1, e_2, \cdots, e_n$，满足存在一个顶点的序列$x_0 = u, x_1, \cdots, x_{n-1}, x_n = v$，使得$e_i$的**initial vertex是$x_{i-1}$**，**terminal vertex是$x_i$**，
$i = 1, \cdots, n$。

cycle和simple path的定义和无向图中的一样。

Connectivity

一个无向图是**connected**的，如果图中任意两个不同的顶点之间都存在一条路径。
一个不是connected的无向图是**disconnected**的。

Lemma: 如果图G中两个不同的顶点x和y之间存在一条路径，那么G中x和y之间存在一条simple path。
Proof:
删除里面的circuit即可。

Theorem: 一个connected的无向图中任意两个不同的顶点之间都存在一条simple path。

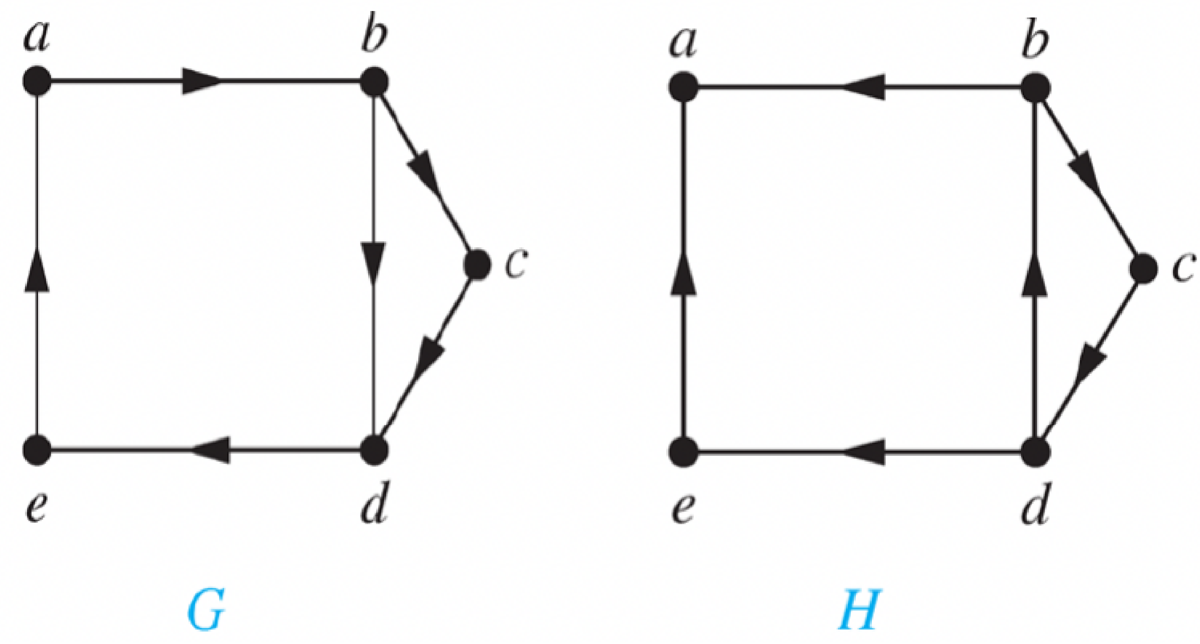
一个图G的**connected component**是一个connected的子图，它不是G的另一个connected的子图的proper subgraph。
也就是说，connected component要满足两个条件：

- 连通性：它是connected的
- 最大性：这个子图是最大的连通子图，意味着它不是另一个更大的连通子图的真子图（即，不是另一个连通子图的一部分）

Connectedness in Directed Graphs

定义：

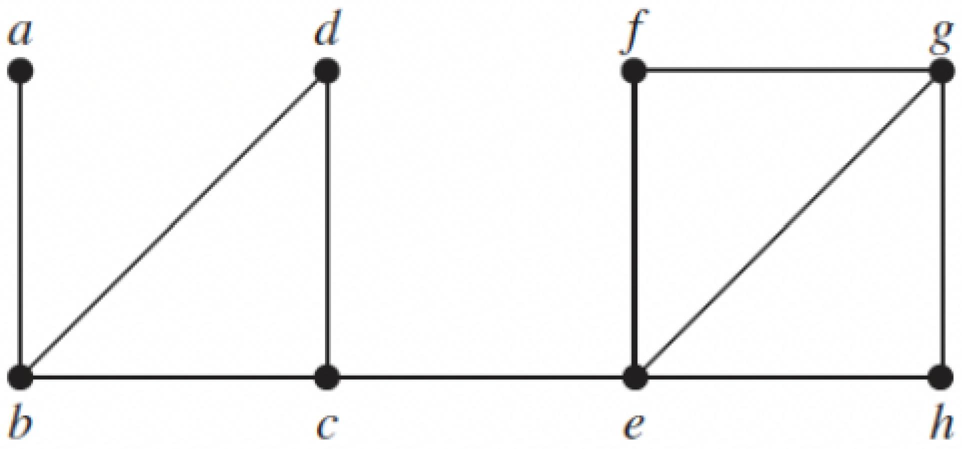
- 一个有向图是**strongly connected**的，如果对于图中的任意两个顶点a和b，a到b有一条path，b到a也有一条path。
- 一个有向图是**weakly connected**的，如果它的underlying undirected graph是connected的。



G是strongly connected的，H是weakly connected的。

Cut Vertices and Cut Edges

Sometimes the removal from a graph of a vertex and all incident edges disconnect the graph.
Such vertices are called cut vertices. Similarly we may define cut edges.
有时候，从一个图中删除一个顶点和所有与它关联的边会使得图不再是connected的。这样的顶点被称为**cut vertices**。类似地，我们也可以定义**cut edges**。

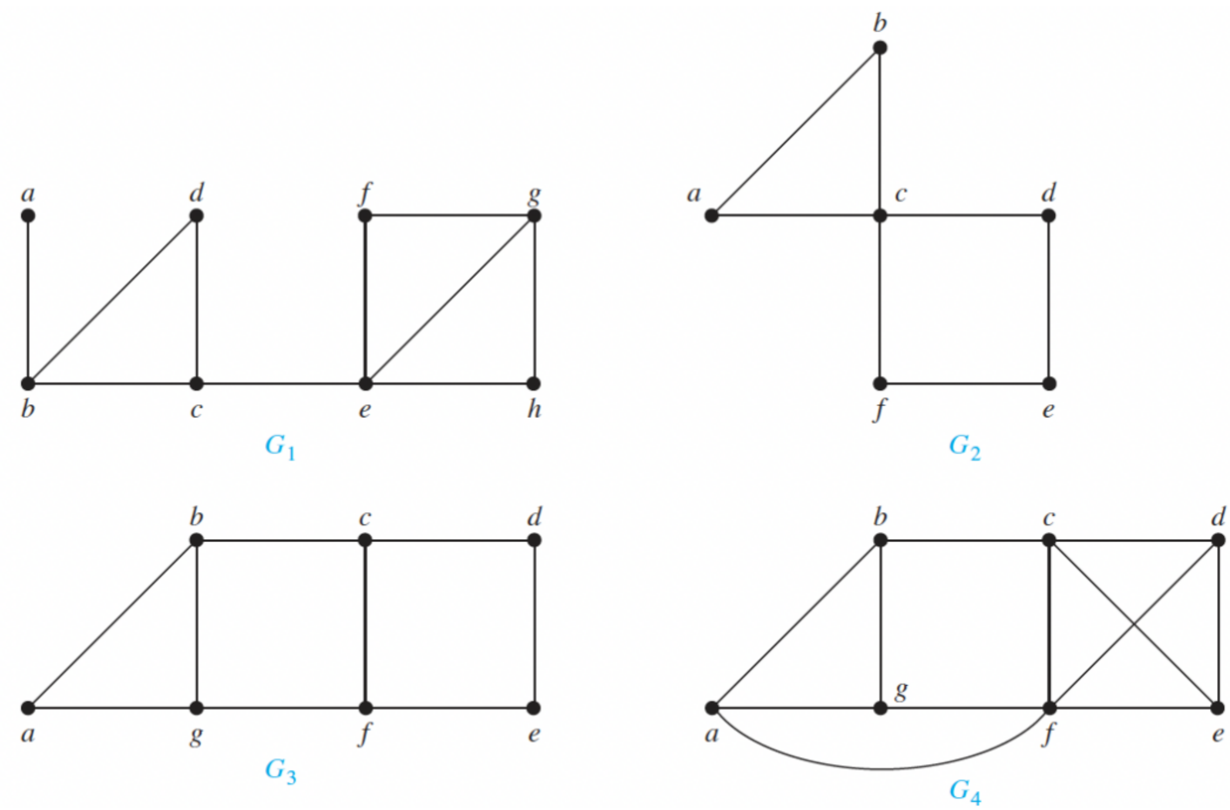


cut vertices: b, c, e

cut edges: $\{a, b\}, \{c, d\}$

一个图G的**edge connectivity** $\lambda(G)$是一个edge cut中边的最小条数。

就是最少需要删除多少条边，才能使得图不再是connected的。这个数值就是edge connectivity。



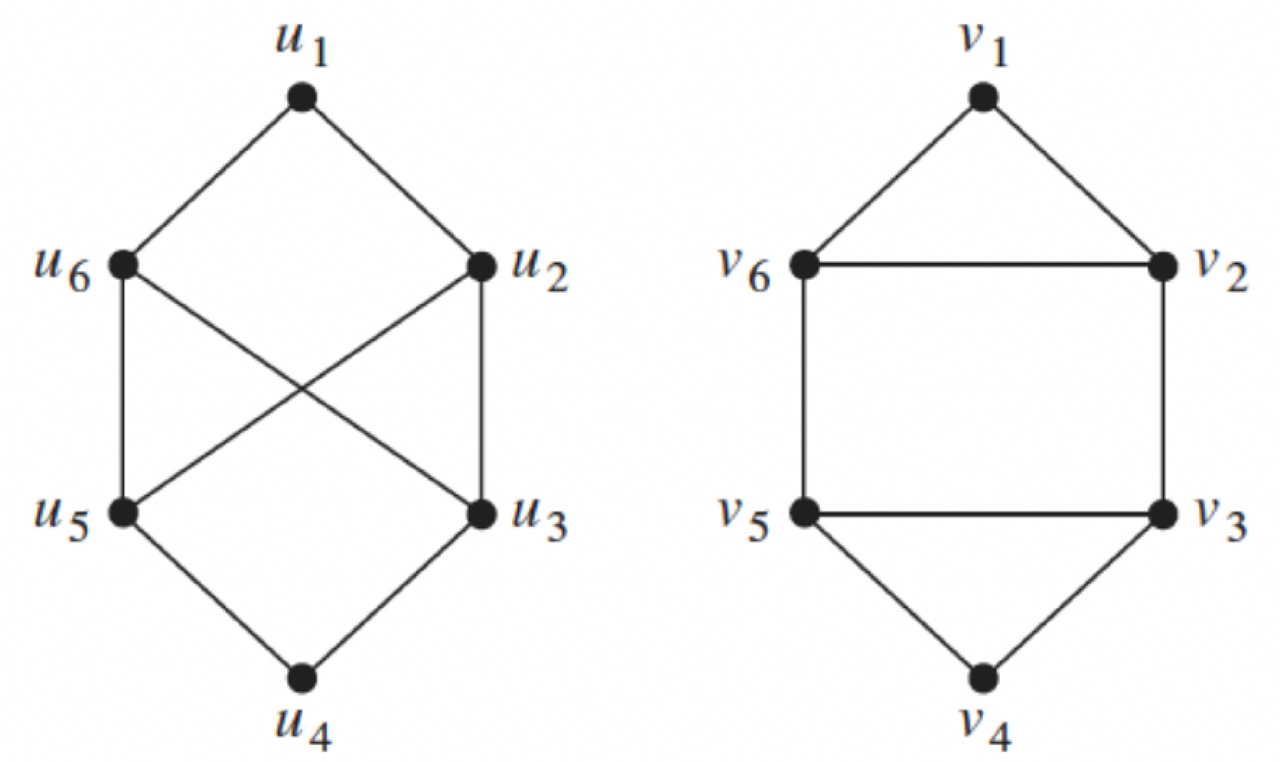
$\lambda(G_1) = 1; \lambda(G_2) = 2; \lambda(G_3) = 2; \lambda(G_4) = 3$

Paths and Isomorphism

长度为k的simple circuit的存在是isomorphic invariant。这可以用来说明两个图不是isomorphic的。

例1

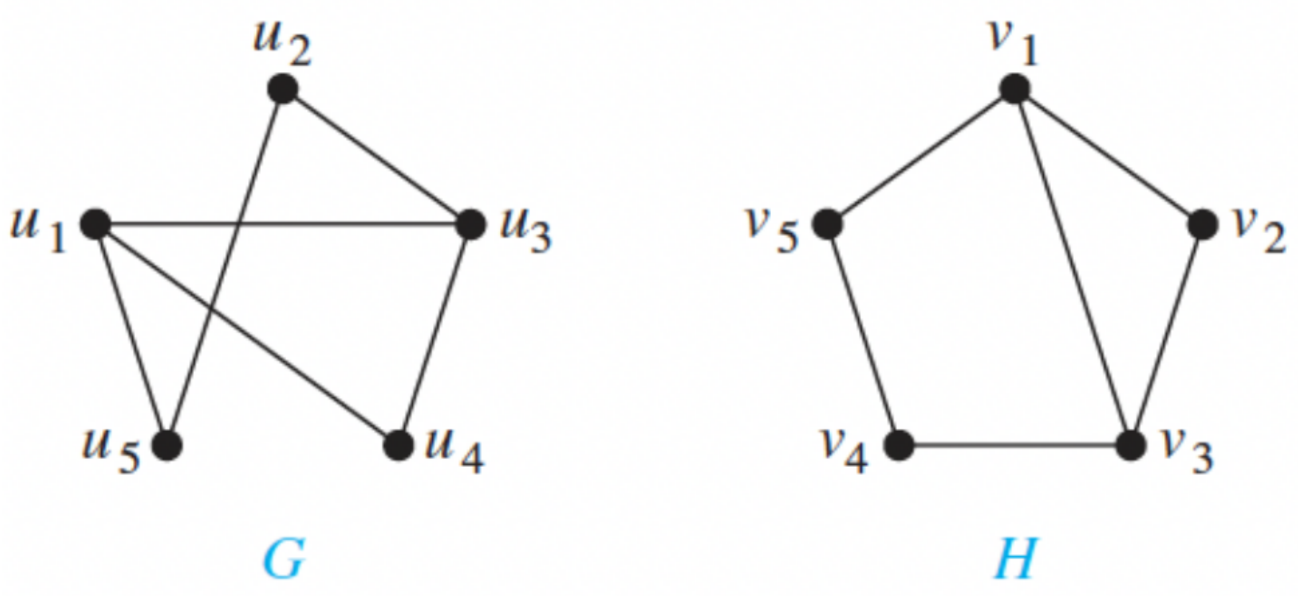
下面两个图是不是isomorphic的？



不是isomorphic的。H有一个长度为3的simple circuit，即v_1, v_2, v_6, v_1，而G没有长度为3的simple circuit。

例2

下面两个图是不是isomorphic的？



因为很多isomorphic invariants（例如，顶点/边的个数，度数，circuit）都是相同的，所以G和H可能是isomorphic的。
令
$f(u_1) = v_3, f(u_4) = v_2, f(u_3) = v_1, f(u_2) = v_5, f(u_5) = v_4$
。我们可以证明f是一个isomorphism，所以G和H是isomorphic的。

总结

至此，我们来总结一下几个isomorphic invariants：

- number of vertices (顶点的个数)
- number of edges (边的条数)
- degree sequence (度数序列)
- existence of simple circuits of various lengths (长度为k的simple circuit的存在)

Counting Paths between Vertices

Theorem: 假设G是一个图，A是G的adjacency matrix，顶点的顺序是$v_1, v_2, \cdots, v_n$。从v_i到v_j的长度为r的不同的path的个数，其中r是一个正整数，等于A^r的(i, j)位置的值。

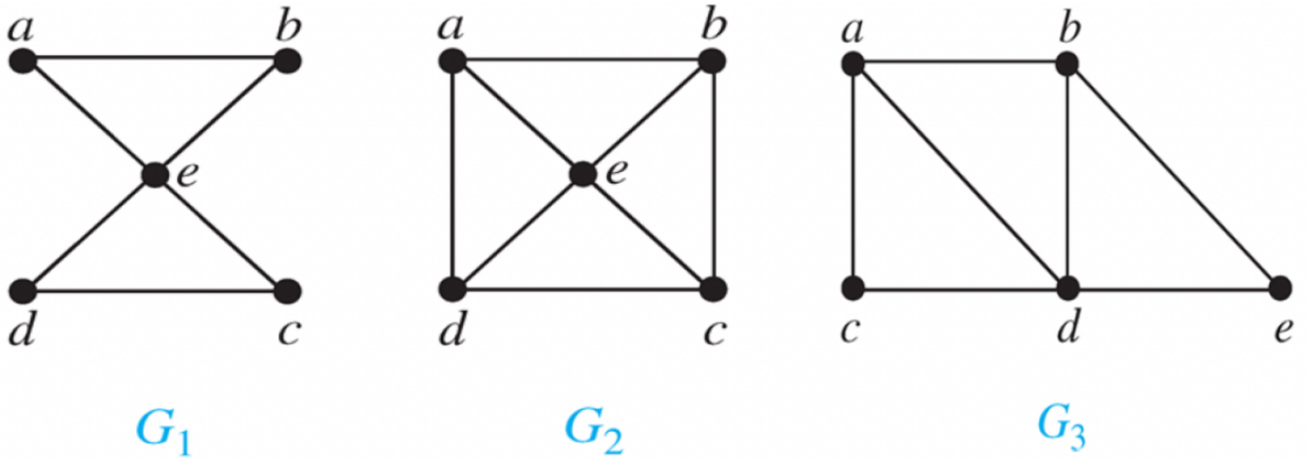
Euler Paths and Circuits

引入： Ko¨nigsberg seven-bridge problem: 有人想知道是否可以从城镇的某个位置出发，穿过所有的桥一次而不重复，然后回到起点。

定义: 一个图G中的**Euler circuit**是一个包含G中所有边的simple circuit。一个图G中的**Euler path**是一个包含G中所有边的simple path。

例1

下面那些有Euler circuit，哪些有Euler path？



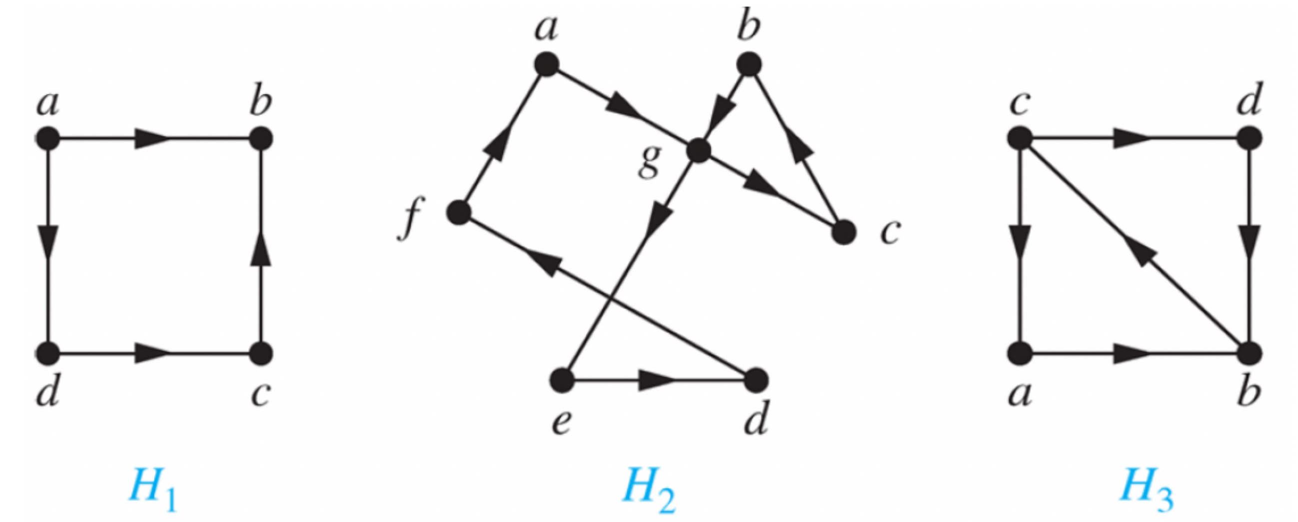
G1: 有Euler circuit，例如a, e, c, d, e, b, a;

G2: 既没有Euler circuit，也没有Euler path;

G3: 有Euler path，例如a, c, d, e, b, d, a, b

例2

下面那些有Euler circuit，哪些有Euler path？



H1: 既没有Euler circuit，也没有Euler path;

H2: 有Euler circuit，例如a, g, c, b, g, e, d, f, a;

H3: 有Euler path，例如c, a, b, c, d, b

Necessary Conditions for Euler Circuits and Paths

- Euler Circuit: 每个顶点的度数都是偶数
- Euler Path: 除了两个顶点的度数是奇数，其他顶点的度数都是偶数。这条path的起点和终点是这两个度数为奇数的顶点。
  这里书里分了充分条件和必要条件，我不是很明白

Lec 18 P41 看不懂

Hamilton Paths and Circuits

Euler path是每个边都只经过一次
Hamilton path是每个顶点都只经过一次

Necessary Conditions for Hamilton Circuits and Paths

没有已知的充要条件可以判断存在Hamilton circuit或path。

但是有一些**sufficient conditions**：

- Dirac's Theorem: 如果G是一个简单图，$|V| \geq 3$，并且G中每个顶点的度数都大于等于$\frac{|V|}{2}$，那么G有一个Hamilton circuit。
- Ore's Theorem: 如果G是一个简单图，$|V| \geq 3$，并且对于G中的任意两个不相邻的顶点u和v，u和v的度数之和都大于等于$|V|$，那么G有一个Hamilton circuit。
  例子：证明K_n有Hamilton circuit

Hamilton path问题是NP-completez·的

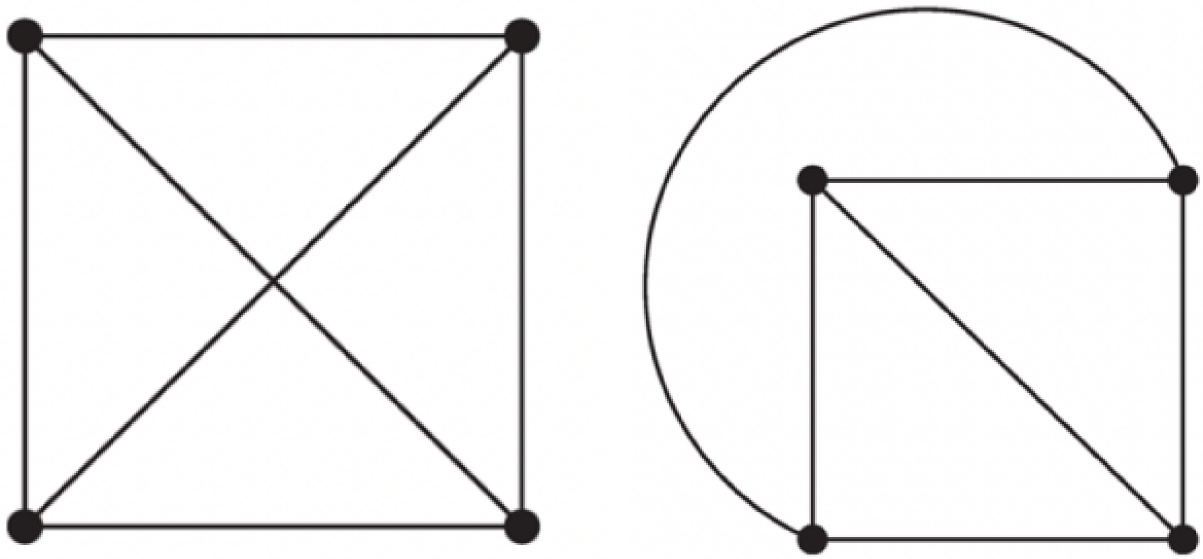
最短路劲问题

想必学过ECE220的各位应该都已经知道该用什么算法了

Planar Graphs

Definition: A graph is called **planar** if it can be drawn in the **plane without any edges crossing**. Such a drawing is called a **planar representation** of the graph.
定义：如果一个图可以在平面上画出来，而且没有边相交，那么这个图是**planar**的。这样的画法被称为这个图的**planar representation**。

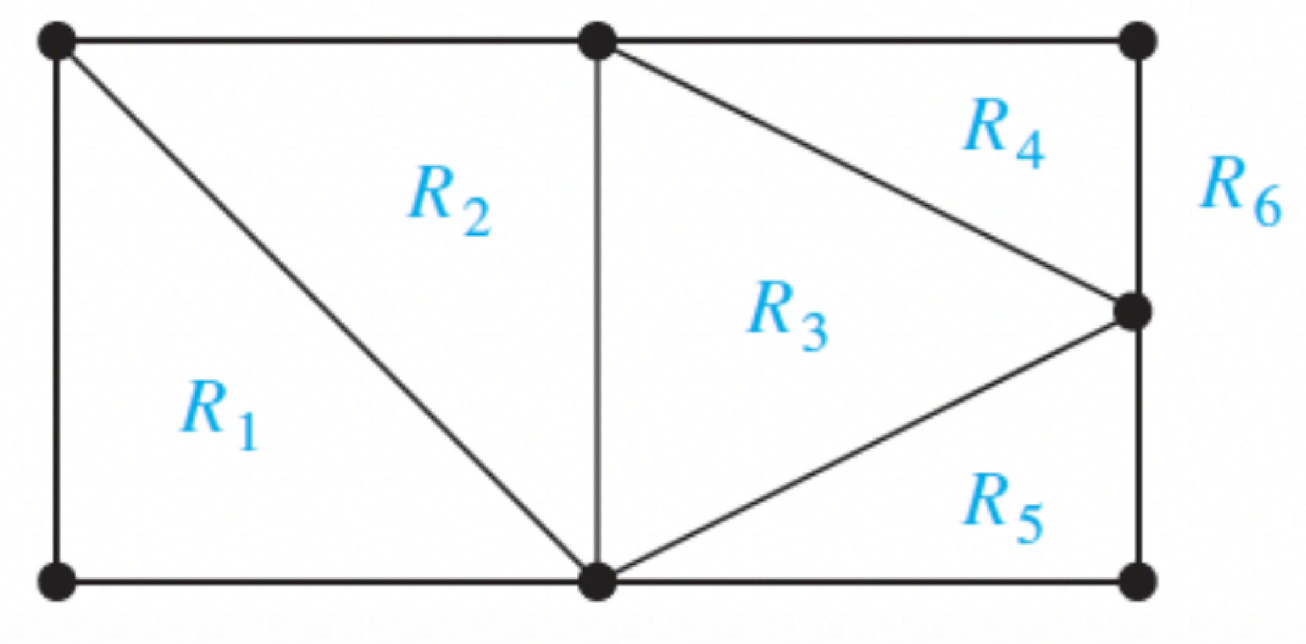
例子：K_4是planar的吗？



关于怎么找planar representation，可以试试先画一个闭合的多边形，然后再把剩下的顶点一个一个加进去

Euler's Formula

一个图的planar representation把平面分成了一些区域，包括一个无界区域。



Theorem (Euler's Formula): 假设G是一个connected的planar simple graph，e是边的条数，v是顶点的个数，r是G的planar representation中的区域的个数。那么，$r = e - v + 2$。

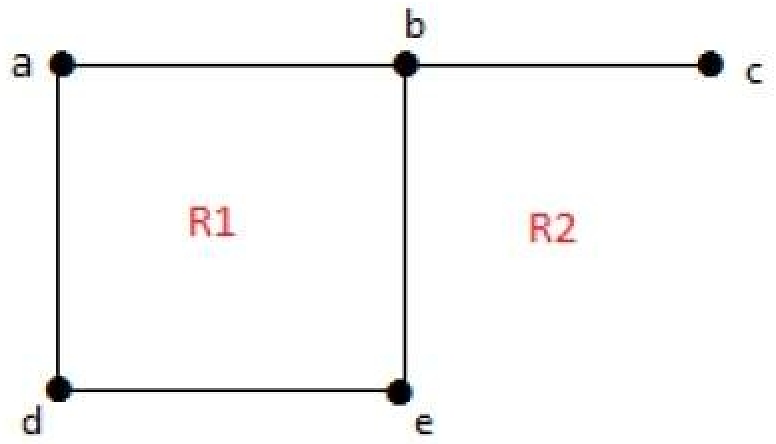
The Degree of Regions

定义：一个region的**degree**是指这个region的边的条数。当一个边在边界上出现两次的时候，它对degree的贡献是2。
这句话非常抽象，我们来举两个例子：
例1： #[Euler's Formula]] 中的图（就是上面那个图），
degree of region 1 = 3
degree of region 2 = 3
degree of region 3 = 3
degree of region 4 = 3

degree of region 5 = 3

degree of region 6 = 7

例2： 上面那个比较正常，我们看个抽象的：



degree of region 1 = 4

degree of region 2 = 6 (这里bc被计算了两次)

Corollaries (推论)

Corollary 1

如果G是一个connected的planar simple graph，e是边的条数，v是顶点的个数，$v \geq 3$，那么$e \leq 3v - 6$。

Proof:

$$\begin{aligned} 2e &= \sum_{v \in V} deg(v) \\ &\geq 3r \\ &= 3(e - v + 2) \end{aligned}$$

Hence, $e \leq 3v - 6$.

Corollary 2

如果G是一个connected的planar simple graph，那么G有一个顶点的度数不超过5。
Proof (by contradiction):
如果G有一个或两个顶点，那么结论显然成立。
如果G至少有三个顶点，那么由Corollary 1，$e \leq 3v - 6$。如果G中每个顶点的度数都大于5，那么有$2e = \sum_{v \in V} deg(v) \geq 6v$，所以$e \geq 3v$，这与$e \leq 3v - 6$矛盾。

Corollary 3

如果G是一个connected的planar simple graph，e是边的条数，v是顶点的个数，$v \geq 3$，并且G中没有长度为3的circuit，那么$e \leq 2v - 4$。

Graph Coloring

Four-color theorem

Four-color theorem: 给定一个平面，把它分成一些连续的区域，产生一个叫做map的图形，**最多只需要四种颜色来给map中的区域染色**，使得任意两个相邻的区域的颜色不同。

Chromatic number

色数是指给图中的顶点染色，使得相邻的顶点颜色不同，所需的最少颜色数。
根据Four-color theorem，平面图的色数不超过4。
记号：$\chi(G)$表示图G的色数。
K_n的色数是n。
K_{mn}的色数是2。
C_n的色数是2（当n是偶数的时候）或者3（当n是奇数的时候）。

Trees

Definition

定义：一个**tree**是一个connected的无向图，它没有simple circuit。
Theorem: 一个无向图是一个tree，当且仅当它的任意两个顶点之间都有唯一的simple path。

Rooted Trees

定义：一个**rooted tree**是一个tree，其中一个顶点被指定为root，每条边都是从root指向其他顶点的。
我们可以通过选择任意一个顶点作为root，把一个unrooted tree变成一个rooted tree。
在一个rooted tree中，边的方向可以省略，因为root的选择决定了边的方向。
定义：顶点v的**parent**是指唯一的顶点u，使得u到v有一

条有向边。
当u是v的parent的时候，v被称为u的**child**。
有相同parent的顶点被称为**siblings**。
顶点v的**ancestors**是指从root到v的路径上的顶点，不包括v，包括root。
顶点v的**descendants**是指v的祖先。
一个rooted tree的顶点被称为**leaf**，如果它没有child。
有child的顶点被称为**internal vertices**。
以a为root的**subtree**包括a和a的descendants，以及所有与这些descendants关联的边。

m-ary Trees

定义：如果一个rooted tree的每个internal vertex都有不超过m个children，那么这个rooted tree被称为**m-ary tree**。如果每个internal vertex都有m个children，那么这个rooted tree被称为**full m-ary tree**。特别地，如果$m = 2$，那么这个rooted tree被称为**binary tree**。

Counting Vertices in a Full m-Ary Trees

Theorem: 一个有n个顶点的tree有$n - 1$条边。
Theorem: 一个有i个internal vertices的full m-ary tree有$n = mi + 1$个顶点。
Theorem: 一个有n个顶点的full m-ary tree有$i = \frac{n-1}{m}$个internal vertices和$l = \frac{(m-1)n+1}{m}$个leaves。
Theorem: 一个有i个internal vertices的full m-ary tree

有$n = mi + 1$个顶点和$l = (m-1)i + 1$个leaves。
Theorem: 一个有l个leaves的full m-ary tree有$n = \frac{ml-1}{m-1}$个顶点和$i = \frac{l-1}{m-1}$个internal vertices。

Level and Height

定义：一个rooted tree中顶点v的**level**是指从root到v的唯一的path的长度。
一个rooted tree的**height**是指它的顶点的level的最大值。
Balanced m-ary Trees: 一个高度为h的rooted m-ary tree是**balanced**的，如果所有的leaves都在level h或$h-1$。
Theorem: 一个高度为h的rooted m-ary tree最多有m^h个leaves。
Corollary: 如果一个高度为h的rooted m-ary tree有l个leaves，那么$h \geq \lceil \log_m l \rceil$。如果这个m-ary tree是full和balanced的，那么$h = \lceil \log_m l \rceil$。