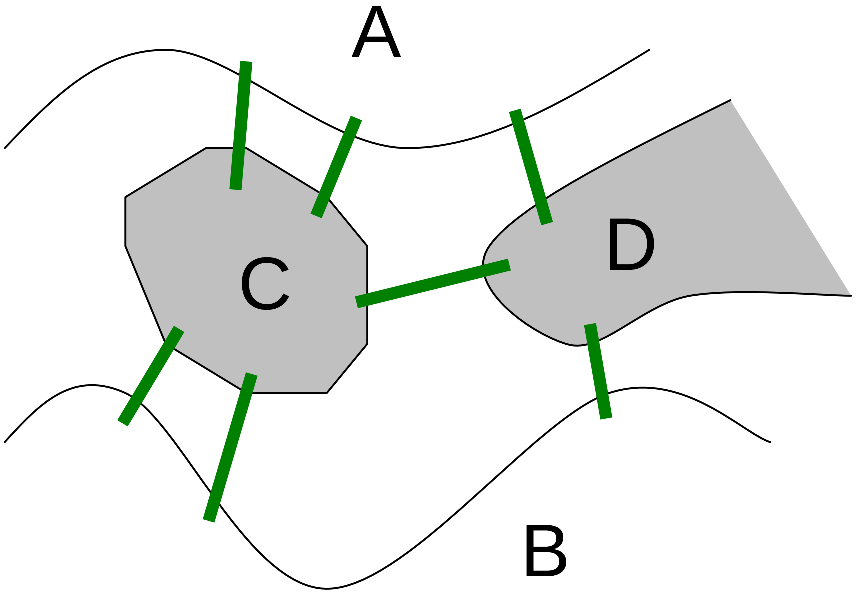
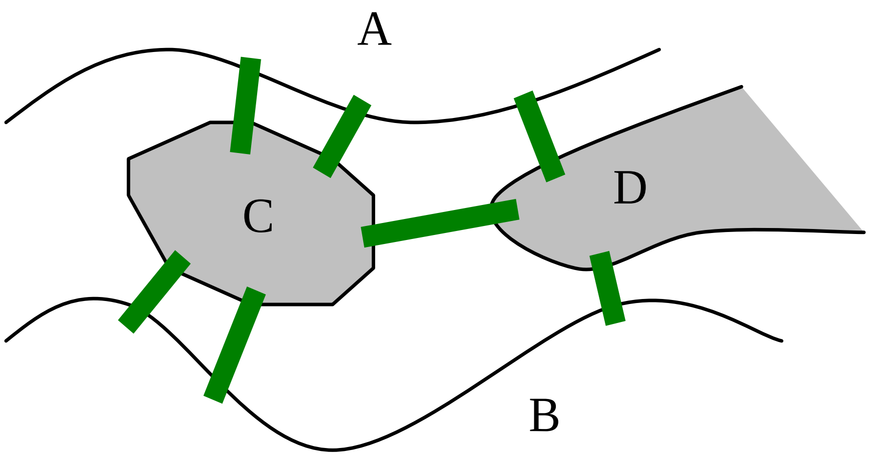
图　论

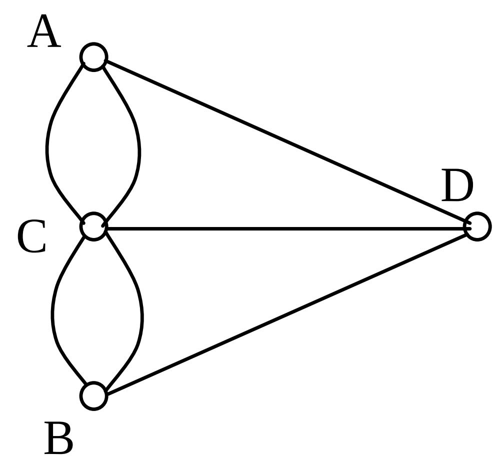
- *History. The famous Konigsburg problem and Euler.* 七桥问题



Seven Bridges at Königsberg

- Abstraction 抽象
  - Vertices （结点） representing objects - areas
  - Edges （边） representing the relationship between objects – connected by a bridge

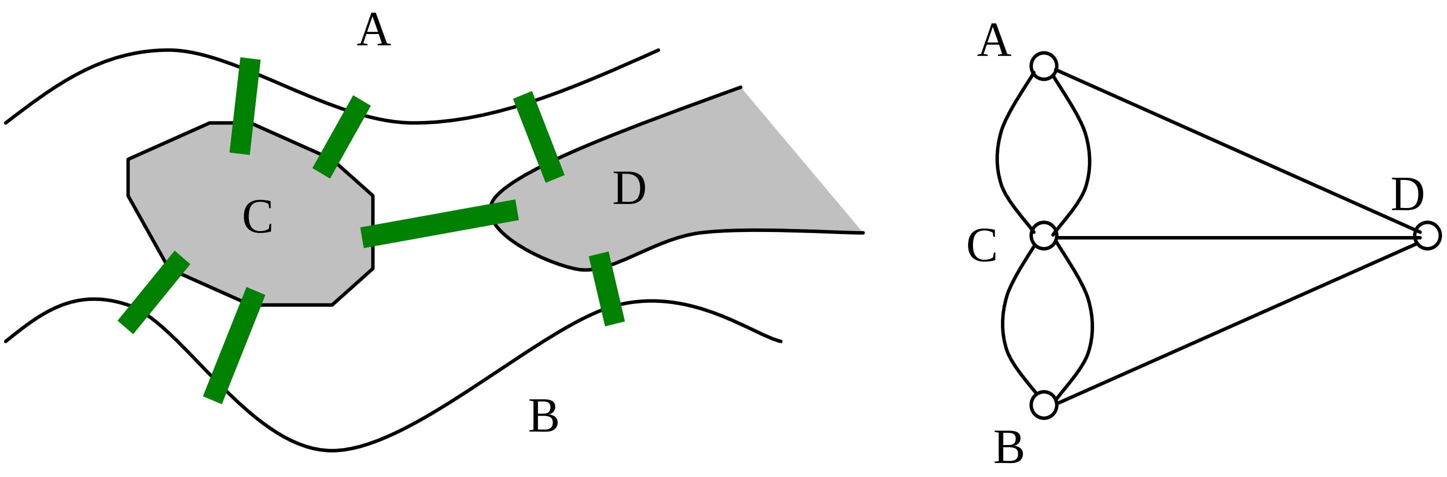




the real-world map

抽象后的图模型graph model of the real world

思考问题



- 问题： 思考一下，从上面左边的图到右边的图，欧拉做了什么重要的工作？

Introduction

- Graph 图——The tool or model for solving some problems using computer

Problem $\xrightarrow{\text{abstracting}}$ Maths Model $\xrightarrow{\text{solve}}$ algorithms

algorithms $\downarrow$ **Programming**

verifications $\xleftarrow{\text{testing}}$ **Programming**

Why Study Graph Theory?

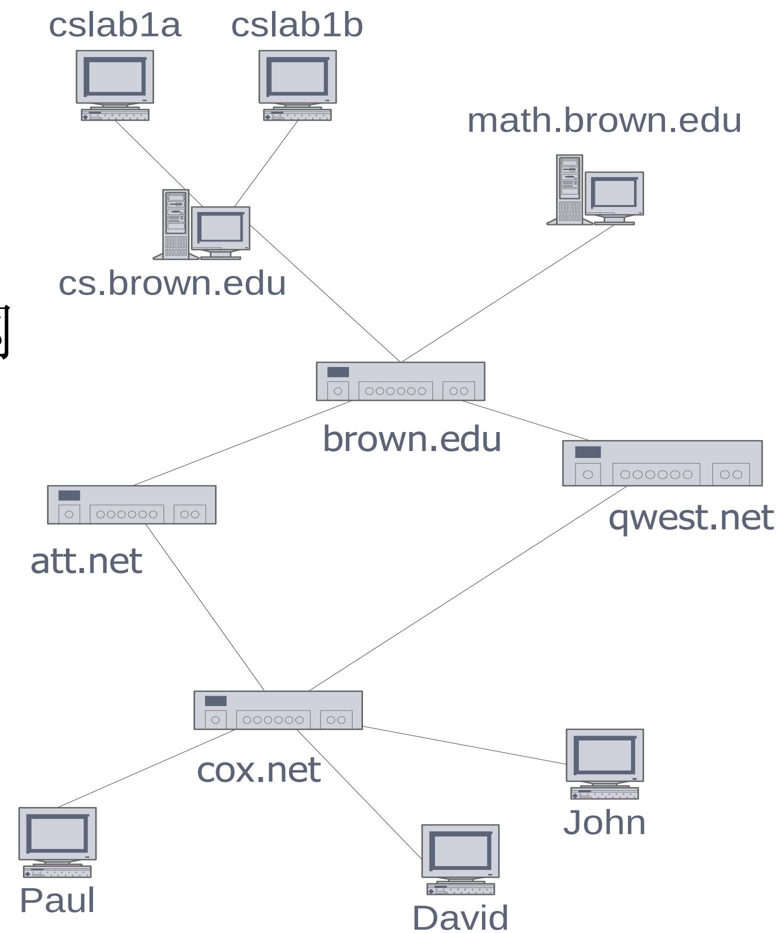
- Model a problem
- Wide variety of real-world applications：

  computer network (e.g., LAN), social network (e.g., Facebook), sensor network (e.g., ecological sensing), robotics (e.g., exploration), AI in games (e.g., path planning), multimedia (e.g., computer vision)

- Advantages of using graph models: useful structural properties, efficient algorithms for solving graph problems
- *Note: Not about drawing a curve in x-y plane!*
- *要注意的是我们这里要说的图，不仅仅是平面上画的图。*

Applications

- Electronic circuits 电子电路
  - Printed circuit board
  - Integrated circuit
- Transportation networks 交通网
  - Highway network
  - Flight network
- Computer networks 计算机网
  - Local area network
  - Internet
  - Web
- Databases 数据库
  - Entity-relationship diagram

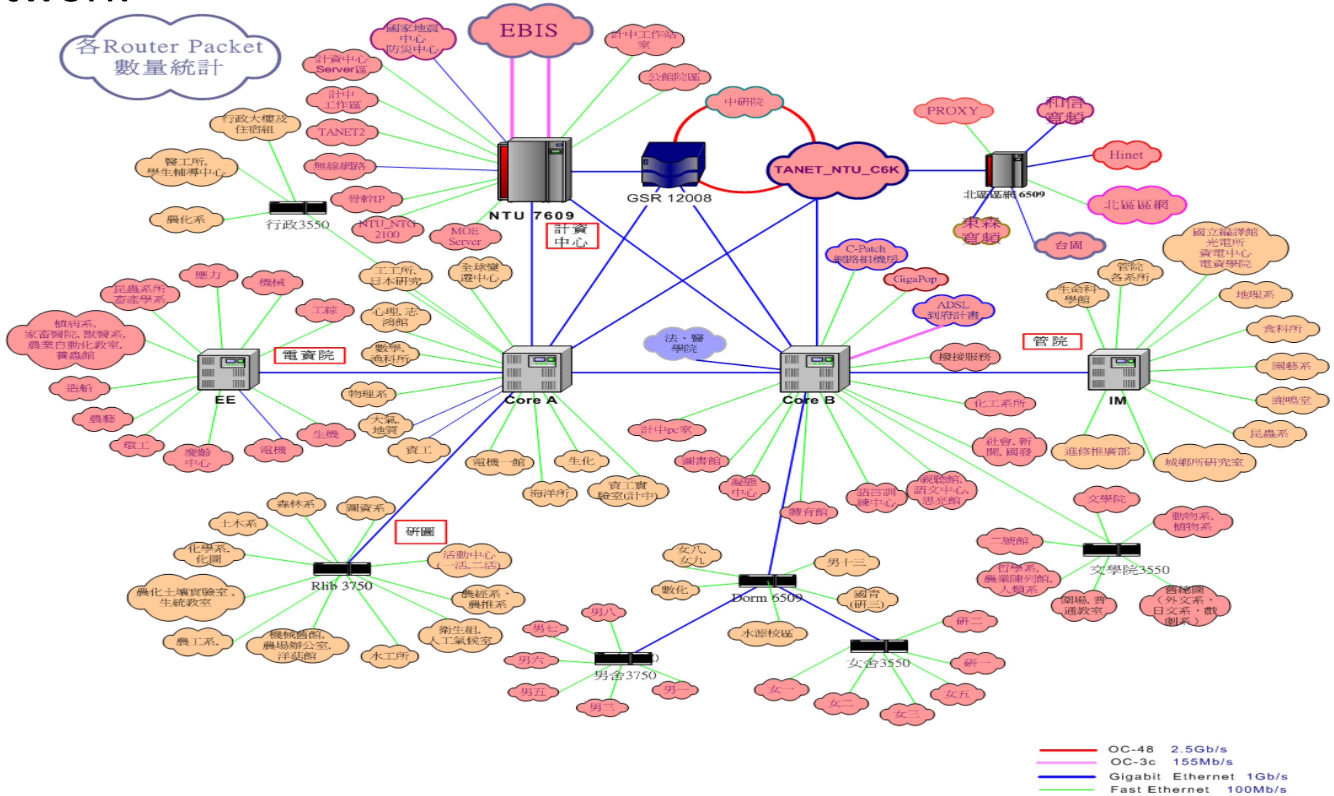


Applications

- Car navigation system 汽车导航系统
- Efficient database 数据库
- Build about to retrieve info of WWW
- Representing computational models 表示计算模型

Examples

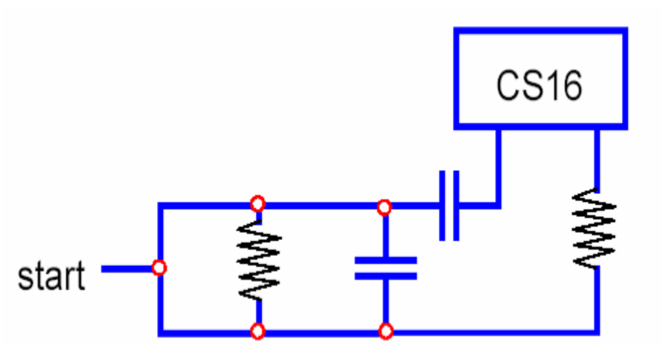
- **Application Background**
  - Network

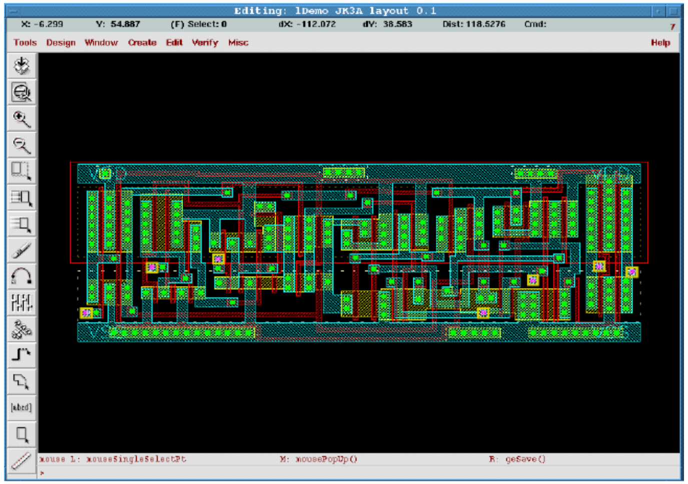


Network Diagram of a College

Examples

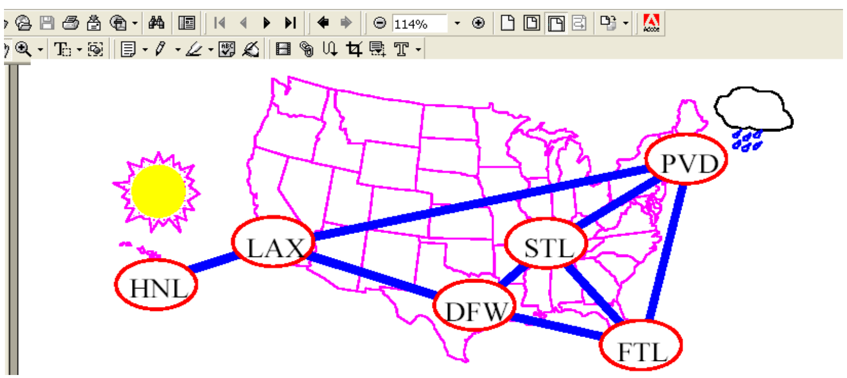
- Network
- circuit

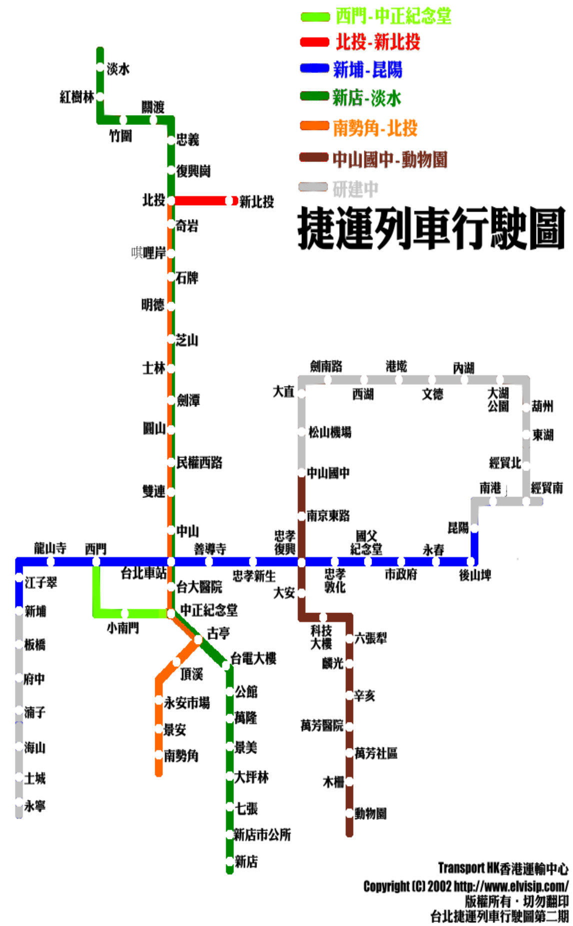




Examples

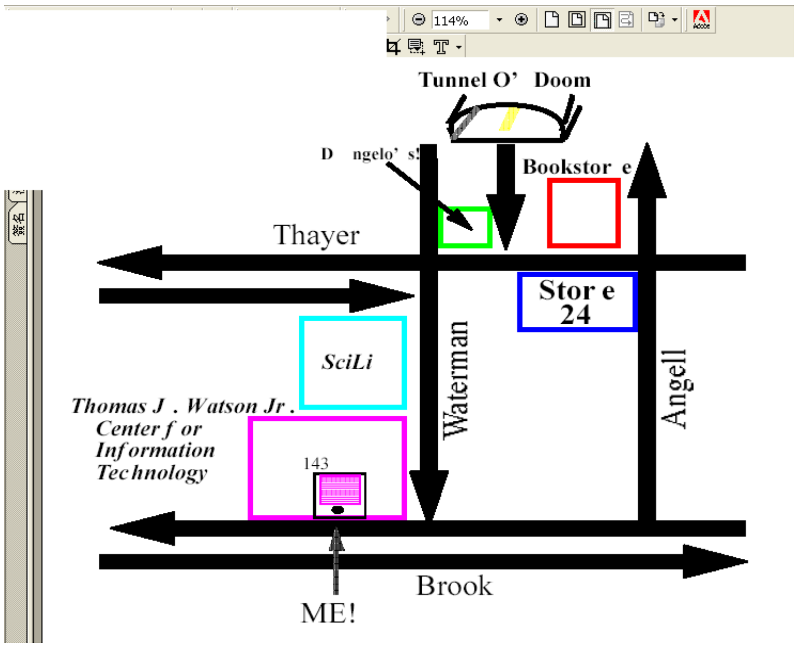
- Network
- circuit
- Transportation Network



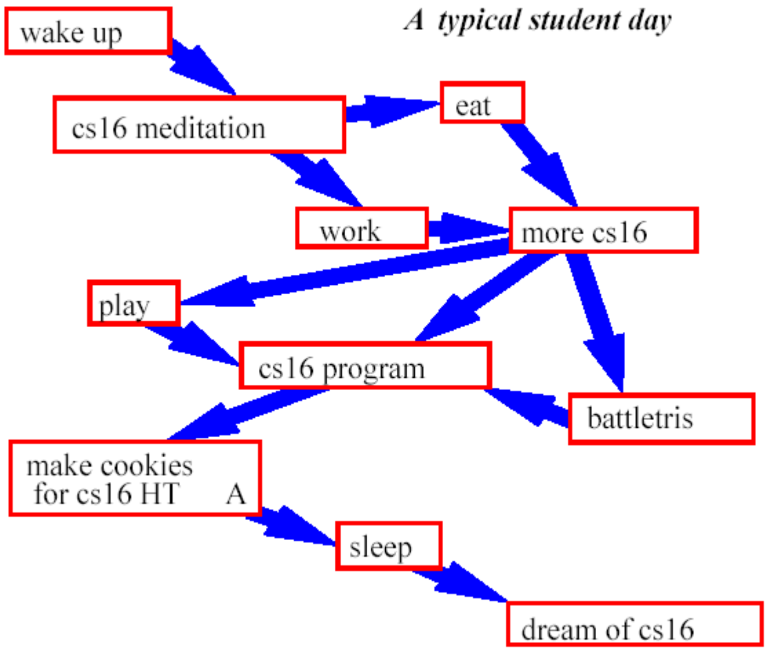


捷運路線图！

Examples



The streets with one-way



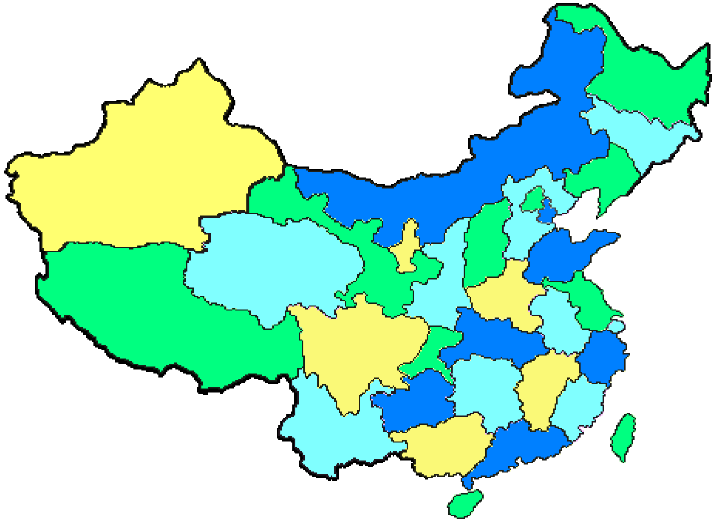
Route table

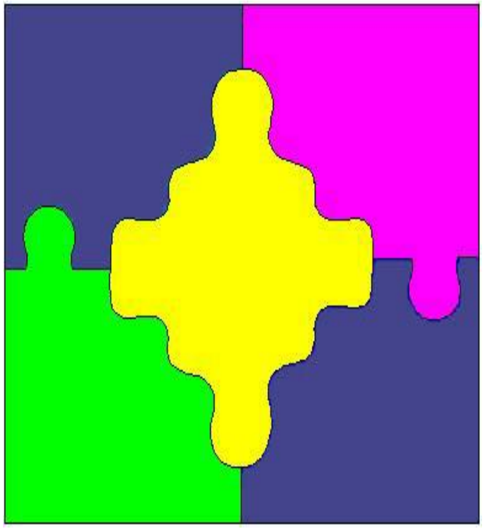
More Applications

- Hypertexts 超文本
- Matching 匹配
- Schedules 规划
  - Projects schedules
- Program structure 程序结构
  - Relations between/among multiple modules in large software system.

more…

– **Graph coloring 图着色**

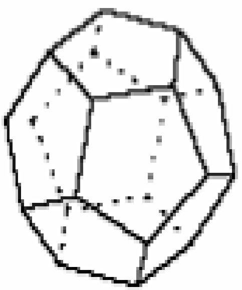




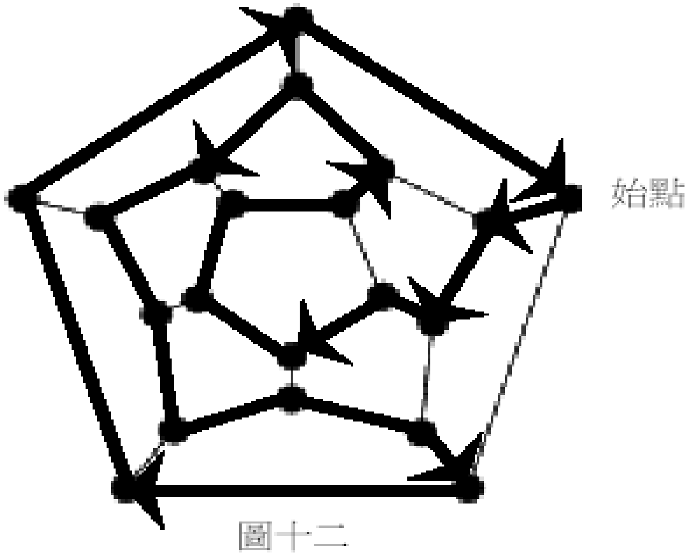
1976年，Appel, Haken 和 Koch 利用计算机辅助证明了四色猜想，但其数学证明仍不理想。

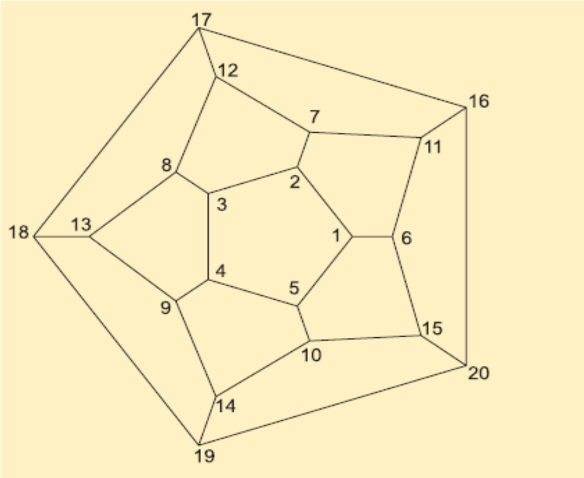
Examples

- 3. Hamilton Traveling Problem 环球旅游问题



Project to a plain

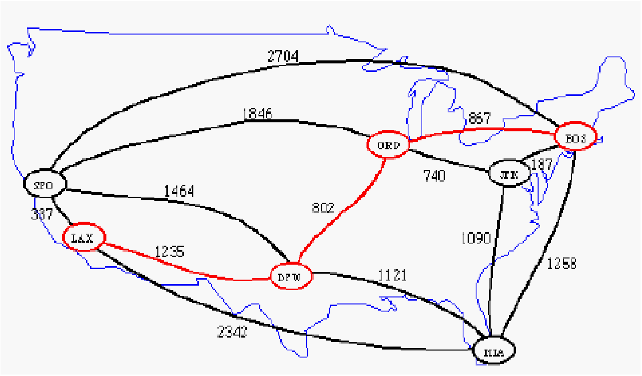




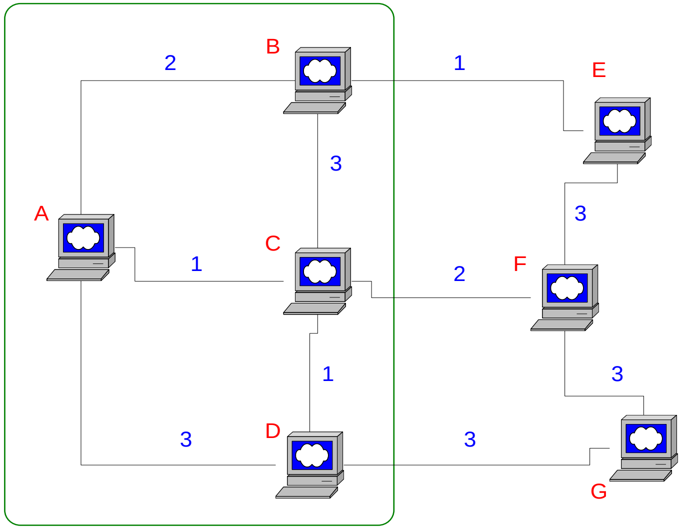
Whether there is Hamilon route

Examples

- 4. **Shortest Path Problem 最小通路问题**



The fastest routing



Examples—crossing river

- Problem: 人、狼、羊、白菜过河问题

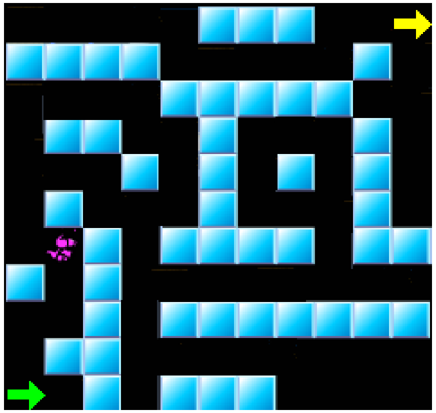
A person(P), a wolf(W), a lamb(L) and a cabbage(C) will cross a river by a boat which can carry any two of them once. Wolf and lamb, or, lamb and cabbage, cannot stay together without the person present. Remember that only the person can run the boat.

船很小，每次人只能带一个“客”过河，只有人会划船。怎样布局和安排，才能保证所有的“客”都安全过河，而不被吃掉？

（建立图模型，用图论的知识解决）

Examples

- **5. Depth-First Search**(走迷宫与**深度优先搜索** 问题)



老鼠走迷宫→深度优先搜索

两点之间有无道路可通？有多少条道路可通？哪条路最短？

What we need to know first

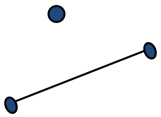
- 1. What is graph? 什么是图？
- 2. Related concepts and properties 相关概念和性质
- 3. How to model the real-world applications using graph structure? 如何用图来建模？

- Not about drawing a curve in x-y plane!
- It is a kind of mathematics structure! 是一种数学结构、模型和工具

Graphs –Intuitive Notion

A graph is a bunch of vertices (or nodes) represented by circles which are connected by edges, represented by line segments.

一些点以及可能存在的连接这些点的边



Mathematically, graphs are binary-relations on their vertex set (except for multigraphs).

某种意义上说，图就是结点集上的一种二元关系

Motivation

- Many real-world problems can be expressed as sets of objects and (binary) relations between these objects. For example, see some previous figures.
- 很多实际问题可以表示成集合及对象之间的关系，这种关系就可以用直观的图模型来表示
- Binary relations may be represented as graphs. However, graphs provide better intuition for humans. This is probably due to our brain structure.

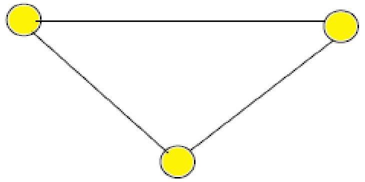
图的定义

- Definition. $G =_{def} (V;E)$, where V is a set of vertices (a vertex 顶点或结点 denotes an object) and $E \subseteq V \times V$ is a set of edges.
- 问题：还记得$V \times V$ 的子集是什么不？
- A vertex is also called a node in some literature.
- （结点，节点，顶点等）

- An edge could be directed (for directed graphs) or undirected (i.e. bi-directed, for undirected graphs).
- 边：可以是有向的，也可以是无向的

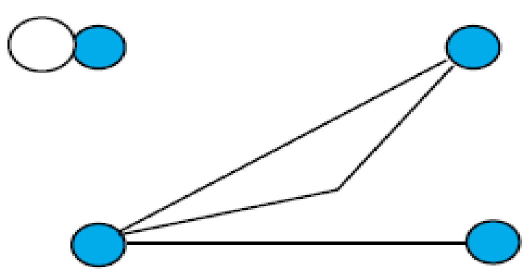
Different Kind of Graphs 不同类型的图

- Simple Graph简单无向图: A *simple* graph (V,E) consists of vertices, V, and edges, E, connecting distinct elements of V.



  - no arrows
  - no loops 无环
  - can't have multiple edges joining vertices

Pseudograph伪图 : is a multigraph which permits loops.



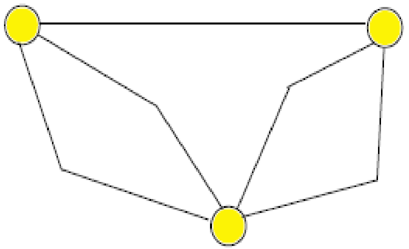
A pseudograph

Different Kind of Graphs

- Multiple-Graph 多重图:A *multigraph* allows multiple edges for two vertices

  - for example, it is imaginable that there are two or more highways between two cities. Thus, in some graphs, we may allow there are two or more edges between two vertices.

  - *redundancy in networks*

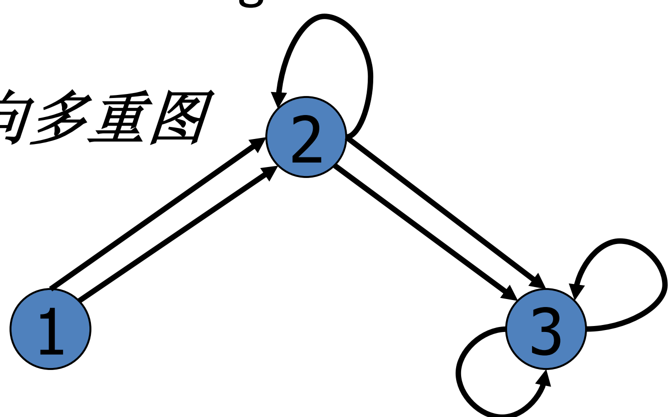


- Directed graph 有向图

  - single directed edges between vertices

  -A ***directed graph*** (or ***digraph***) $G = (V,E)$ consists of a non-empty set V of ***vertices*** (or ***nodes***) and a set E of ***edges*** with $E \subseteq V \times V$. The edge (a,b) is also denoted by $a \rightarrow b$ and a is called the ***source*** of the edge while b is called the ***target*** of the edge.

- Example: a ***directed multigraph 有向多重图***



- Undirected graph 无向图:

- Weighted Graphs (later) 有权图（加权图）
  - For some purpose, each edge has a length.

Graph Models(总结）

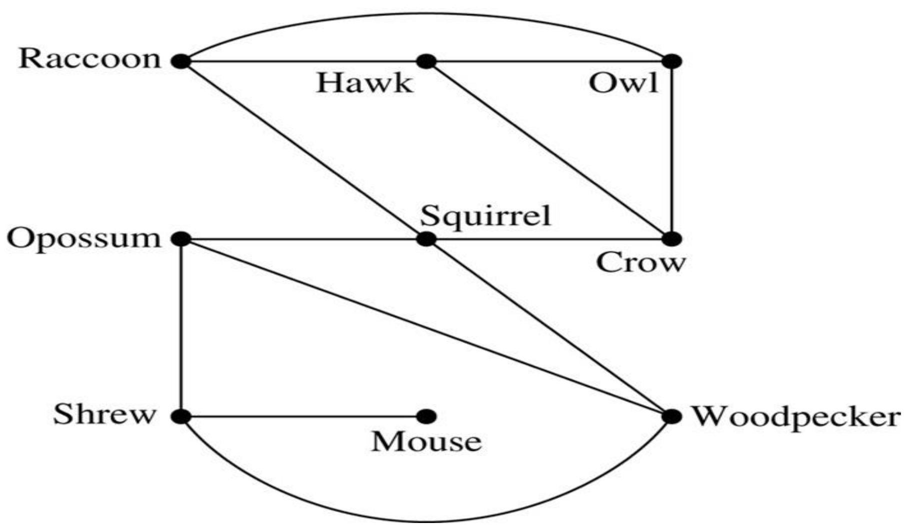
- As long as we deal with something regarding the binary relations (or connections) between or among some objects (in a set, or multiple set), we probably can build graph model to solve the problems in real applications.
- 只要是涉及到事物对象之间关系（联系）的，就有可能用图的建模表示
- A graph is actually a binary relation! 图就是一种关系

图论应用建模

- 以下的多个例子都是围绕着如何用图模型去表示表达实际问题的。
- 目的就是学习如何用图模型进行数学建模，走出图论应用的第一步。
- 图论模型是数学模型的一个分支。在图论模型中，顶点可以代表任何事物或对象，边表示的是顶点之间的某种关系或者某种联系。图的实质可以说是关系的一种数学表示。图论建模就是对一些客观事物进行抽象、化简、并用图来描述事物特征及内在联系的过程

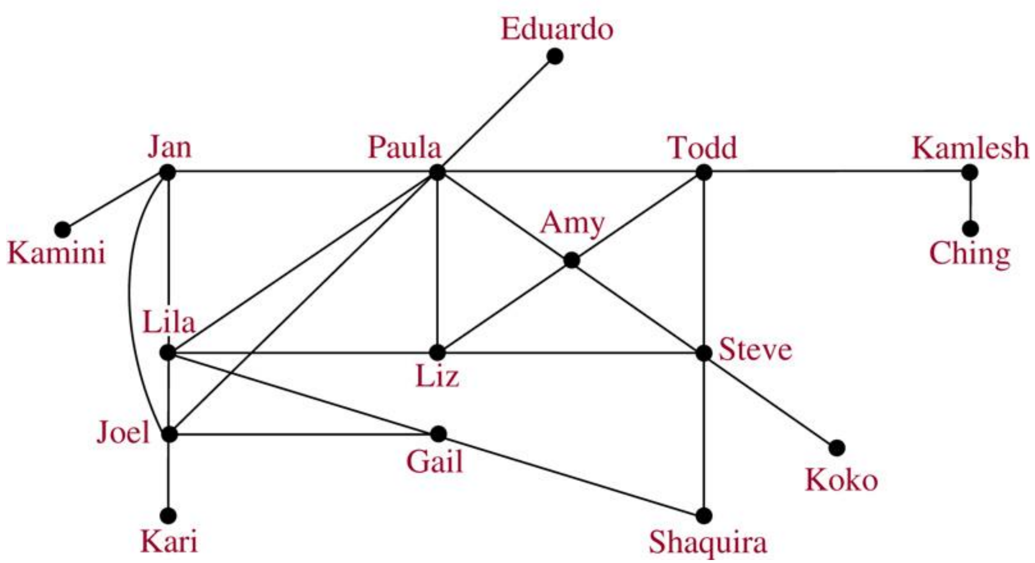
Graph Models 图模型举例

Example 1 Niche Overlap Graphs in Ecology Graphs are used in many models involving the interaction of different species of animals. For instance, the competition between species in an ecosystem can be modeled using a niche overlap graph. Each species is represented by a vertex. An undirected edge connects two vertices if the two species represented by these vertices compete (that is, some of the food resources they use are the same). A niche overlap graph is a simple graph because no loops or multiple edges are needed in this model. We see from this graph that squirrels and raccoons compete but that crows and shrews do not.



Graph Models 图模型举例

Example 2 Acquaintanceship Graphs We can use graph models to represent various relation-ships between people. For example，we can use a simple graph to represent whether two people know each other, that is, whether they are acquainted. Each person in a particular group of People is represented by a vertex. An undirected edge is used to connect two people when these people know each other. **<u>No multiple edges and usually no loops are used</u>**. (If we want to include the notion of self-knowledge, we would include loops.) The acquaintanceship graph of all people in the world has more than six billion vertices and probably more than one trillion edges!

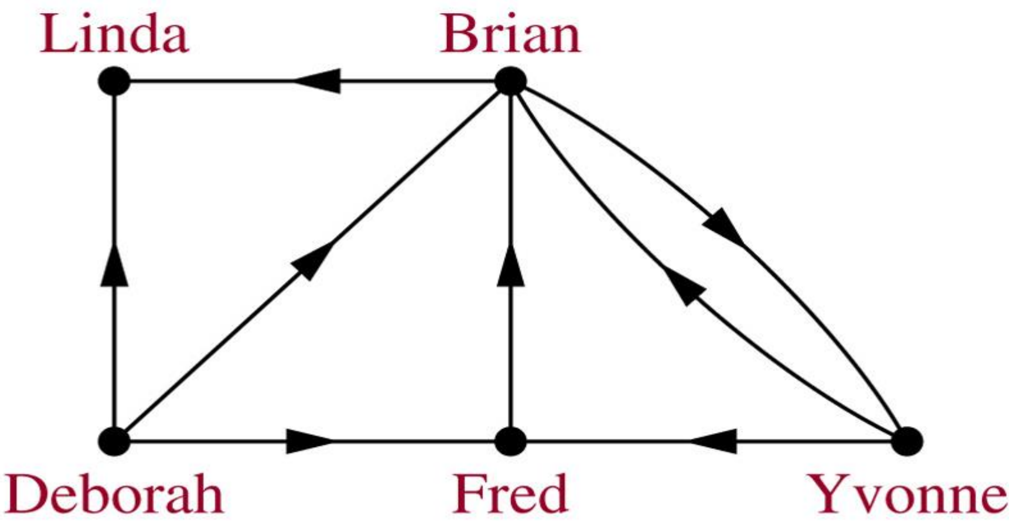


思考问题

- 在上面这个例子中，如果是表达一个人领导另一个人，或者是被领导的关系。该用什么样的图来表示？

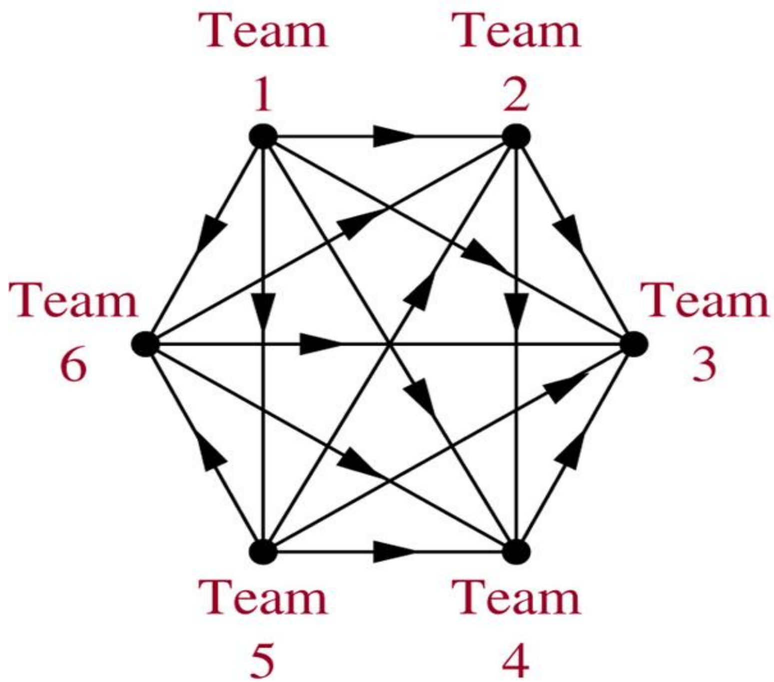
Graph Models 图模型举例

Example 3 Influence Graphs In studies of group behavior it is observed that certain people can influence the thinking of others. A directed graph called an influence graph can be used to model this behavior. Each person of the group is represented by a vertex. There is a directed edge from vertex a to vertex b when the person represented by vertex a influences the person represented by vertex b. This graph does not contain loops and it does not contain multiple directed edges. In the group modeled by this influence graph, Deborah can influence Brian, Fred, and Linda, but no one can influence her. Also, Yvonne and Brian can influence each other.



Graph Models 图模型举例

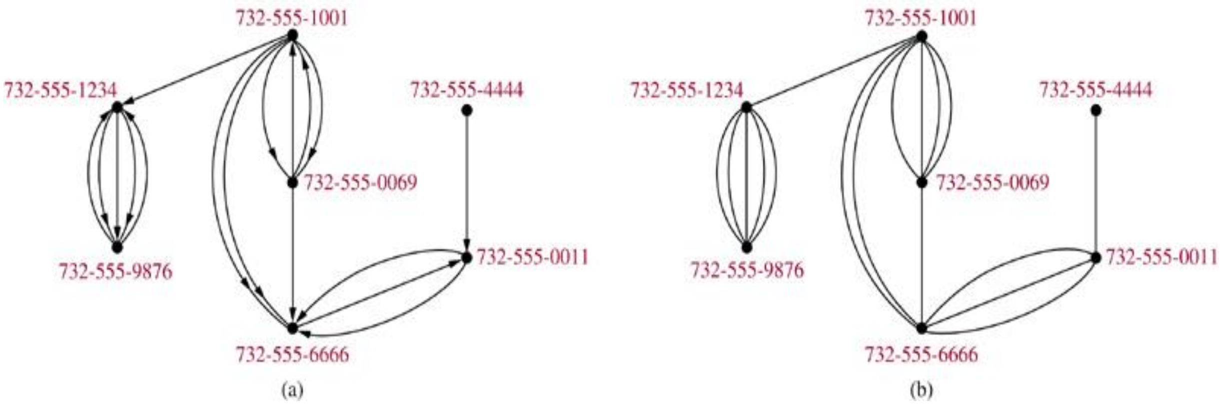
Example 5 Round-Robin Tournaments(单循环赛) A tournament where each team plays each other team exactly once is called a round-robin tournament. Such tournaments can be modeled using directed graphs where each team is represented by a vertex. Note that (a, b) is an edge if team a beats team b. This graph is a simple directed graph, containing no loops or multiple directed edges(because no two teams play each other more than once). We see that Team I is undefeated in this tournament, and Team 3 is winless.



- 思考问题：在上面这个例子中，如果是表达队与队之间是否有过比赛，有过多少场比赛，每一场比赛胜负关系如何，还有平局的出现等。该用什么样的图来表示？

Graph Models 图模型举例

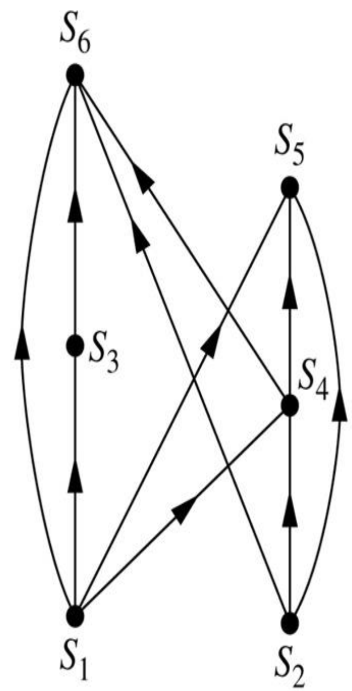
Example 7 Call Graphs Graphs can be used to model telephone calls 电话呼叫 made in a network, such as a long-distance telephone network. In particular, a directed multigraph can be used to model calls where each telephone number is represented by a vertex and each telephone call is represented by a directed edge. The edge representing a call starts at the telephone number from which the call was made and ends at the telephone number to which the call was made. We need directed edges because the direction in which the call is made matters. We need multiple directed edges because we want to represent each call made from a particular telephone number to a second number.



Graph Models

Example 9 Precedence Graphs 优先图 and Concurrent Processing 并行处理
Computer programs can be executed more rapidly by executing certain statements concurrently. It is important not to execute a statement that requires results of statements not yet executed. The dependence of statements on previous statements can be represented by a directed graph. Each statement is represented by a vertex, and there is an edge from one vertex to a second vertex if the statement represented by the second vertex cannot be executed before the statement represented by the first vertex has been executed. This graph is called a precedence graph. For instance, the graph shows that statement S5 cannot be executed before statements *S1*,*S2* and *S4* are executed.

S_1 $a := 0$
S_2 $b := 1$
S_3 $c := a + 1$
S_4 $d := b + a$
S_5 $e := d + 1$
S_6 $e := c + d$



练习

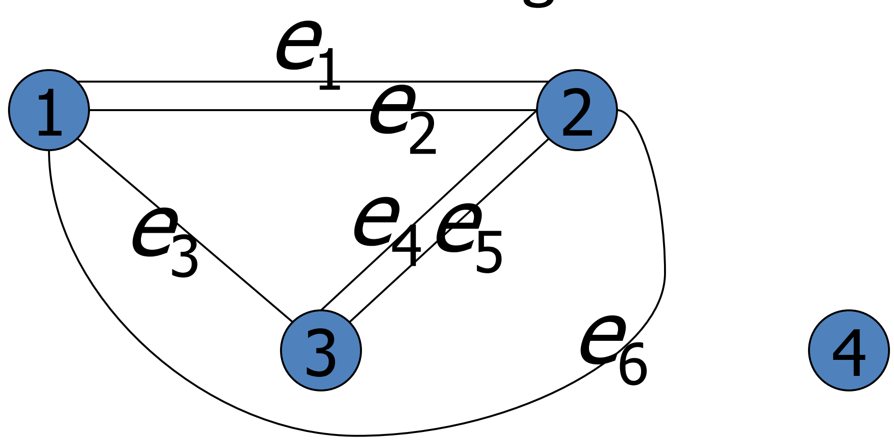
- 6.1节 中文版 T6, T11， T14， T18
- 英文精简版：T6, T10, T13, T17

Graph Terminology
有关术语

Undirected Graphs Terminology
无向图有关术语

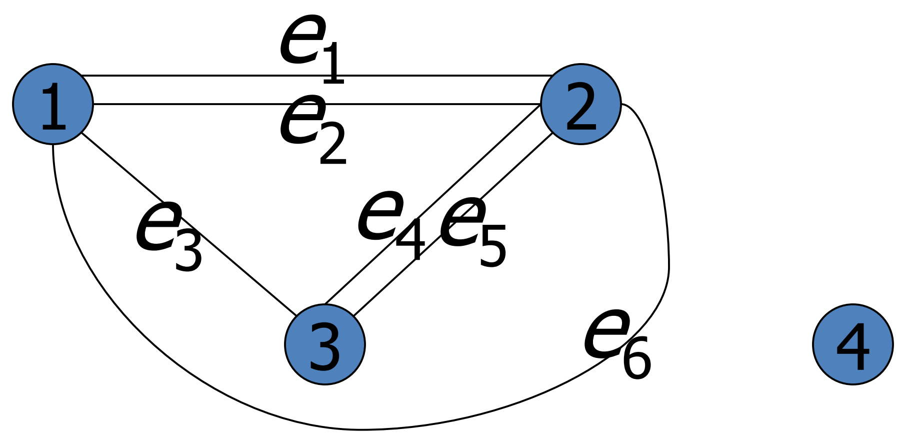
图的两个要素：结点、边

Vertices are ***adjacent***(点与点邻接的) if they are the endpoints of the same edge.



Q: Which vertices are adjacent to 1? How about adjacent to 2, 3, and 4?

Undirected Graphs Terminology



A: 1 is adjacent to 2 and 3

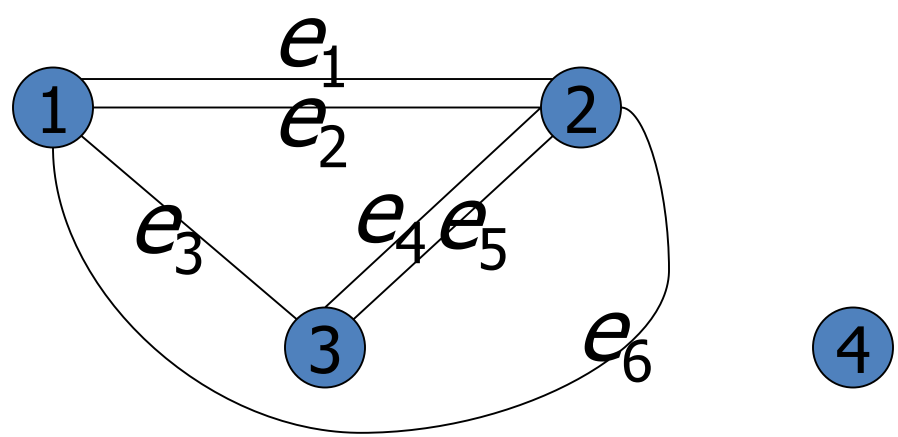
2 is adjacent to 1 and 3

3 is adjacent to 1 and 2

4 is not adjacent to any vertex

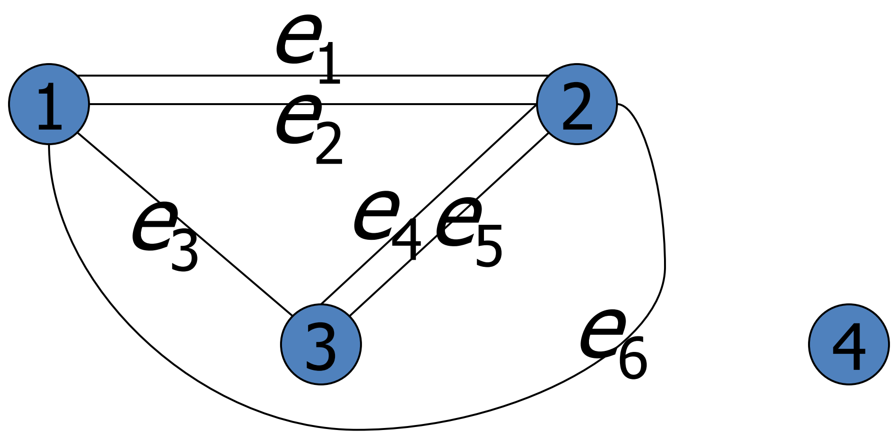
Undirected Graphs Terminology

A vertex is ***incident(点与边的关联)*** with an edge (and the edge is incident with the vertex) if it is the endpoint of the edge.



Q: Which edges are incident to 1? How about incident to 2, 3, and 4?

Undirected Graphs
Terminology



A: e_1, e_2, e_3, e_6 are incident with 2

2 is incident with e_1, e_2, e_4, e_5, e_6

3 is incident with e_3, e_4, e_5

4 is not incident with any edge

Graph Terminology

- Definition: Degree of vertex （结点的度）
  – *deg*(v) = number of edge incident to v （与点关联的边的数目）

    Obviously, deg(v) is a nonnegative integer.

  -- *a loop (环） at a vertex contributes twice to the degree of that vertex.（小圆弧对结点的度的贡献是2）*
- Definition: If the degree of a vertex v is zero 度为零的点, vertex is called Isolated(孤立的、孤立点).
  – A vertex is pendant （悬挂点）iff it has degree one only.

- 思考：那一个图的所以结点的度数之和如何？

Handshaking Theorem 握手定理

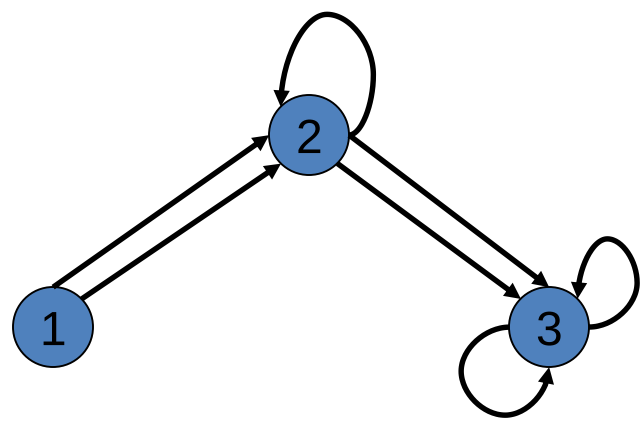
结点度的数量特征

- **Theorem 1**:
  - Sum of degree of all vertices in a graph is even.
  - $\sum_{i=1}^{n} d(v_i) = 2m$, m is the number of edges in the graph.
- 定理：总度数是边数目的2倍
- Why?

- **Theorem 2:** The number of vertices with odd degree must be even! 奇数度的节点数为偶数
- Why?

Oriented Degree when Edges Directed 有向图

The ***in-degree* （入度）** of a vertex (deg^-) counts the number of edges that stick *in* to the vertex. The ***out-degree* 出度** (deg^+) counts the number sticking *out*.



Q: What are in-degrees and out-degrees of all the vertices? 所有结点的出度和入度的关系？

Oriented Degree when Edges Directed

A: $\deg^-(1) = 0$

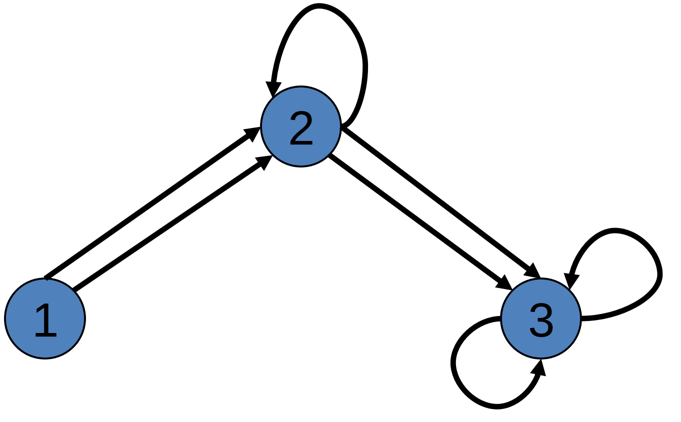
$\deg^-(2) = 3$

$\deg^-(3) = 4$

$\deg^+(1) = 2$

$\deg^+(2) = 3$

$\deg^+(3) = 2$



Handshaking Theorem

Theorem: In an undirected graph

$$|E| = (\sum deg(v))/2$$

In a directed graph 有向图握手定理,如何?

握手原理应用

- 思考问题1: In a party of 5 people can each person be friends with exactly three others?

- 思考问题2：n支球队（n>3)进行比赛，比赛已经进行了n+1场，则存在一支球队，它至少参加了3场比赛。

Solutions of the questions

Solution: Imagine a simple graph with 5 people as *vertices* and edges being undirected edges between friends (simple graph assuming friendship is symmetric and irreflexive). Number of friends each person has is the degree of the person.

Handshaking would imply that

$|E|$ = (sum of degrees)/2 or

$2|E|$ = (sum of degrees) = $(5 \cdot 3) = 15$.

Impossible as 15 is not even.

Paths 路

（基础但重要的概念）

Introduction

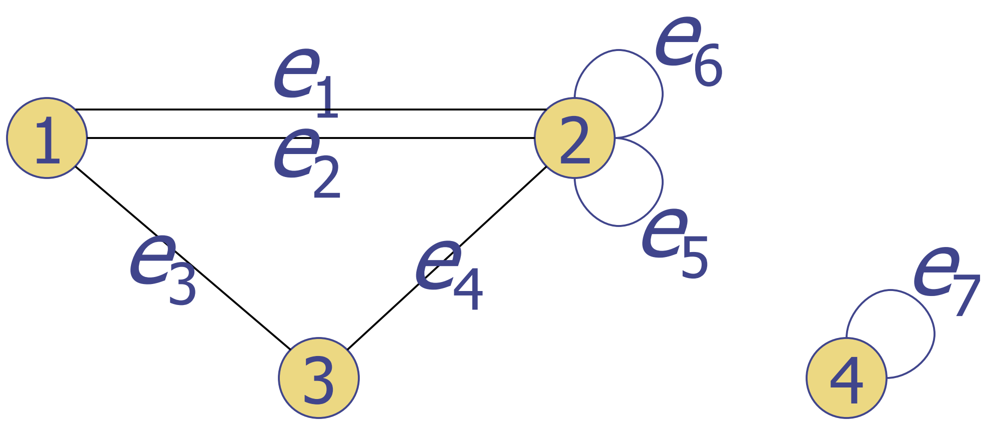
- Many problems can be modeled with paths formed by traveling along the edges of graphs. For example a message routing the internet.

  很多问题都可以用图中的路来建模。如计算机网络中的路由问题。

- The salesman problem…

Paths 路(通路)

图中的一条路，指的是从一个结点到另一个结点的一些连续的**边的序列**.



例如：could get from 1 to 3 circuitously as follows:

$$1\text{-}e_1 \rightarrow 2\text{-}e_1 \rightarrow 1\text{-}e_3 \rightarrow 3\text{-}e_4 \rightarrow 2\text{-}e_6 \rightarrow 2\text{-}e_5 \rightarrow 2\text{-}e_4 \rightarrow 3$$

Paths

Def: 无向图中长度为n的路是 a sequence of n edges e_1, e_2, … ,e_n such that each consecutive pair e_i , e_{i+1} share a common vertex.

在简单无向图中，一条长度为n的路也可以定义为 a sequence of $n+1$ vertices v_0, v_1, v_2, … ,v_n such that each consecutive pair v_i , v_{i+1} are adjacent.

Paths of length 0 are also allowed according to this definition.

思考: Why does the second definition work for simple graphs only?

Paths

简单路的定义：A ***simple path*** 是一条没有重复边的路.

开路：起点跟终点不同的路

A ***cycle*****回路** (or ***circuit***)：起点和终点相同的路

简单回路：没有重复边的回路。

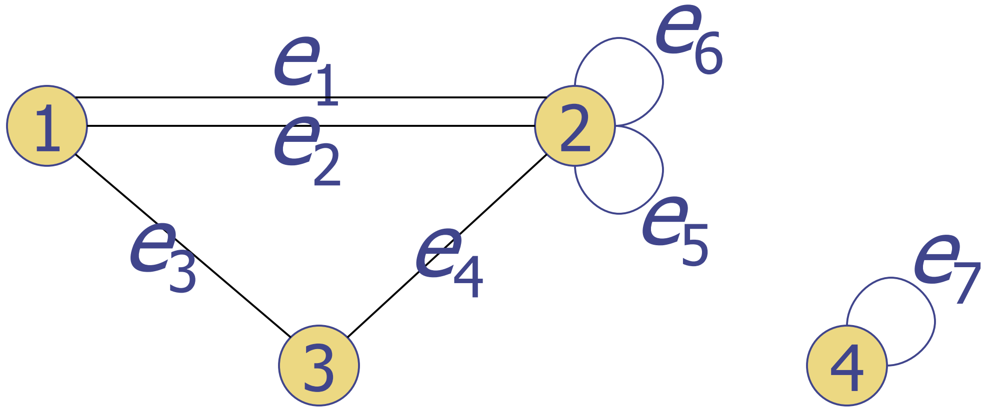
真路：简单图中的路 v_0, v_1, v_2, … , v_n , 如果这些结点互不相同，则成为真路。

（注：简单路未必是真路）

注意：简单路并不意味着只有在简单图里才有。Simple paths need not be in simple graphs. E.g., may contain loops.

Paths

Q: Find a longest possible simple path in the following graph:



有向图中的路

Def: A ***path*** of length n in a directed graph is a sequence of n edges $e_1, e_2, \ldots, e_n$ such that the target of e_i is the source e_{i+1} for each i.

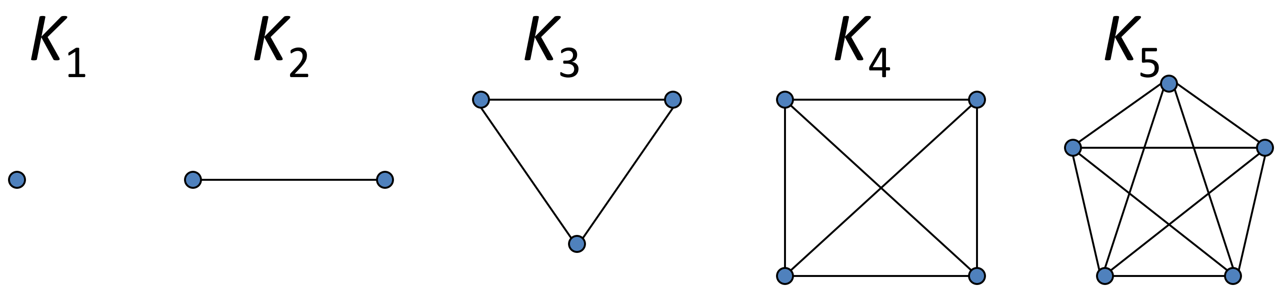
In a *digraph*, one may instead define a path of length n as a sequence of $n+1$ vertices $v_0, v_1, v_2, \ldots, v_n$ such that for each consecutive pair v_i , v_{i+1} there is an edge from v_i to v_{i+1} .（这里有问题吗？）

如果有多重边，这个定义就会有问题

介绍一些特殊的图

Special Graph--*complete graphs Kn* 完全图

A simple graph简单图 is ***complete*** if every pair of distinct vertices share an edge. The notation ***K_n*** denotes the complete graph(n个结点的完全图) on *n* vertices.



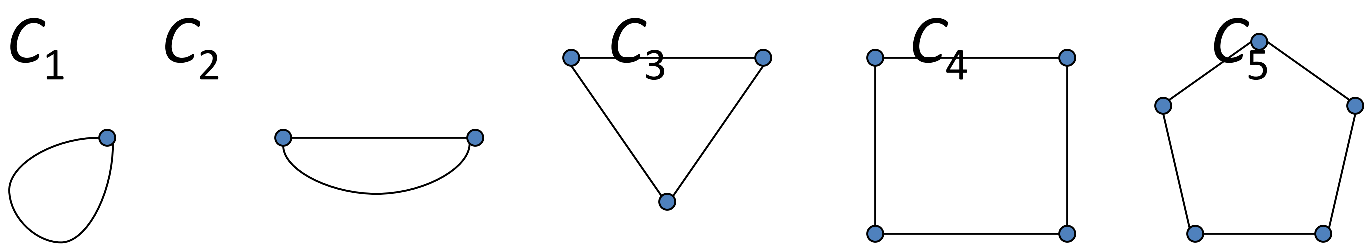
K_n有多少条边?

Special Graph--1. *complete graphs Kn*

- A ***graph*** G and its ***complement graph(补图)*** have the same set of vertex, and their edges set constitute a partition of the edge set of the complete graph with the same vertex set.
- G与G的补图有相同的顶点集合，他们的边集合形成该结点集合上的完全图的边集的分划。
- Example…

Graph Patterns—Cycles（圈图） - C_n

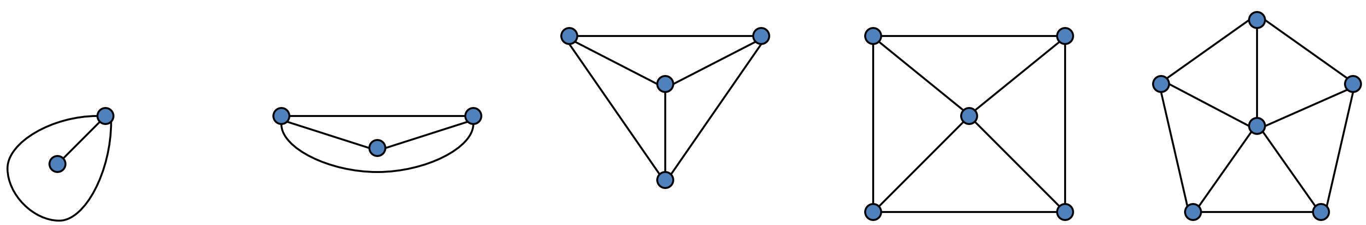
The ***cycle graph*** C_n is a circular graph with V = {0,1,2,…,n-1} where vertex i is connected to i +1 **mod** n and to i -1 **mod** n. They look like polygons:



Q: What type of graph are C_1 and C_2 ?

Graph Patterns Wheels轮图 - W_n

The ***wheel graph*** W_n is just a cycle graph C_n with an extra vertex in the middle, and the vertex connects to each of the vertexs of $C_{n.}$



W_1 W_2 W_3 W_4 W_5

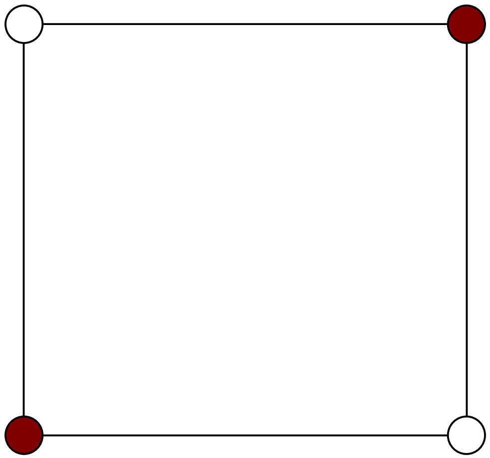
Usually consider wheels with 3 or more spokes only.

Bipartite Graphs(偶图或二部图或二分图)

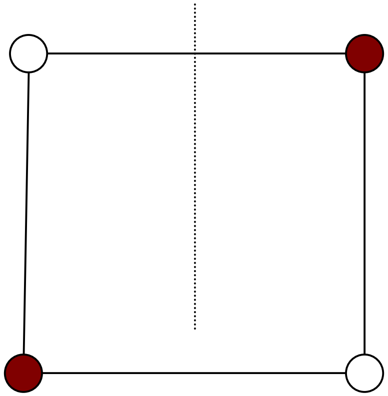
A simple graph is ***bipartite*** if V can be partitioned into $V = V_1 \cup V_2$ so that any two adjacent vertices (任意两个邻接的结点) are in different parts of the partition V_1, V_2. Another way of expressing the same idea is ***bichromatic*** *双色图*: vertices can be colored using two colors so that no two vertices of the same color are adjacent.

Bipartite Graphs偶图

举例：C_4 is a bichromatic:

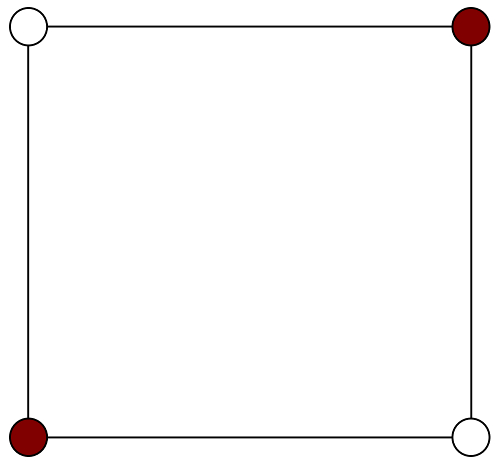


And so is bipartite, if we redraw it:

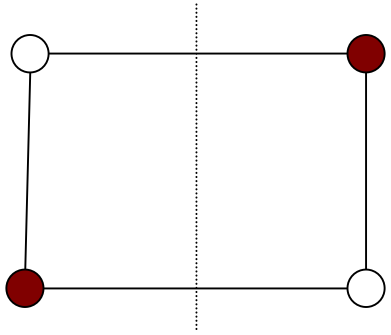


Bipartite Graphs

EG: C_4 is a bichromatic:

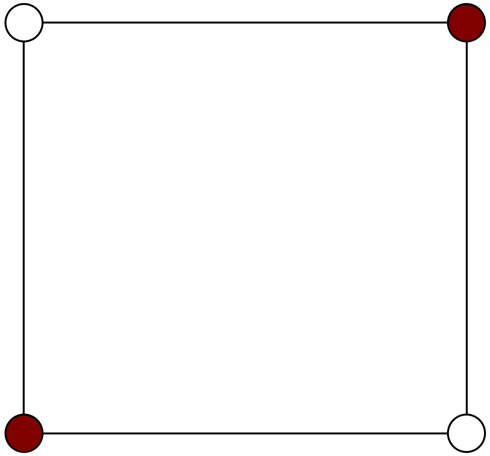


And so is bipartite, if we redraw it:

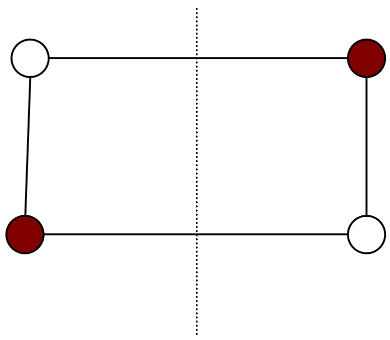


Bipartite Graphs

EG: C_4 is a bichromatic:

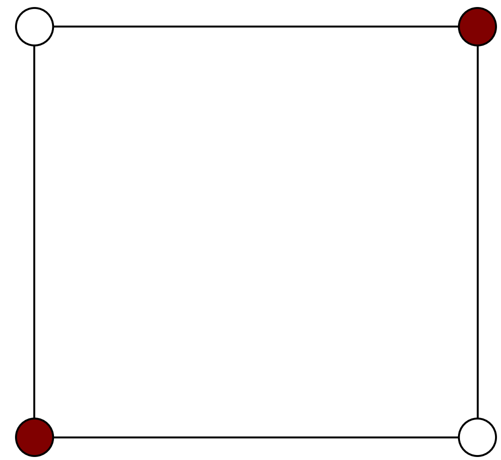


And so is bipartite, if we redraw it:

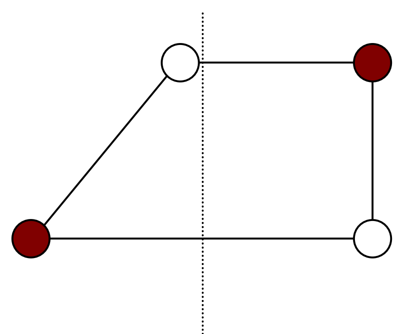


Bipartite Graphs

EG: C_4 is a bichromatic:

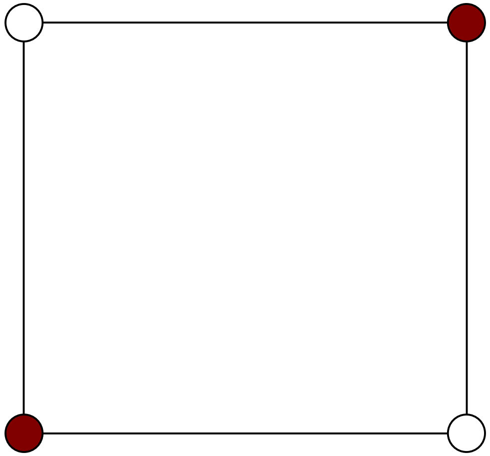


And so is bipartite, if we redraw it:

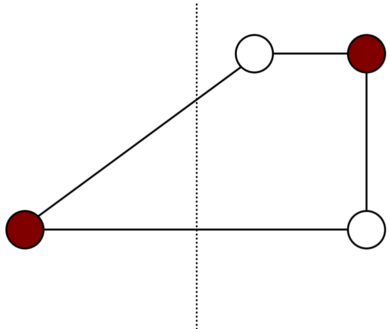


Bipartite Graphs

EG: C_4 is a bichromatic:

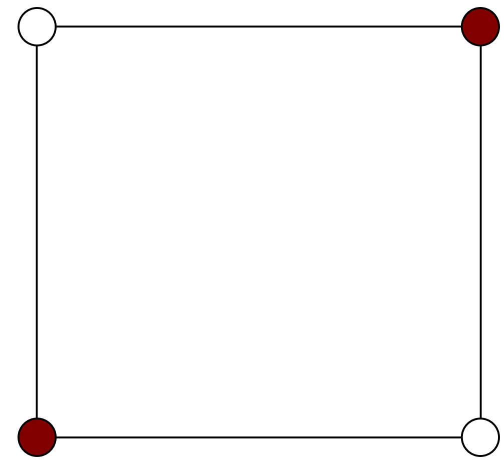


And so is bipartite, if we redraw it:

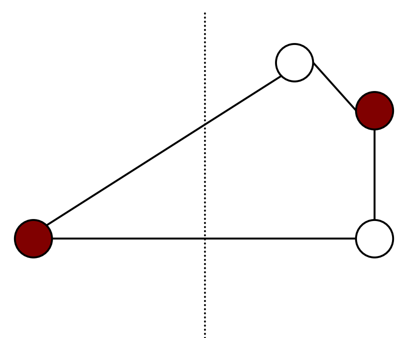


Bipartite Graphs

EG: C_4 is a bichromatic:

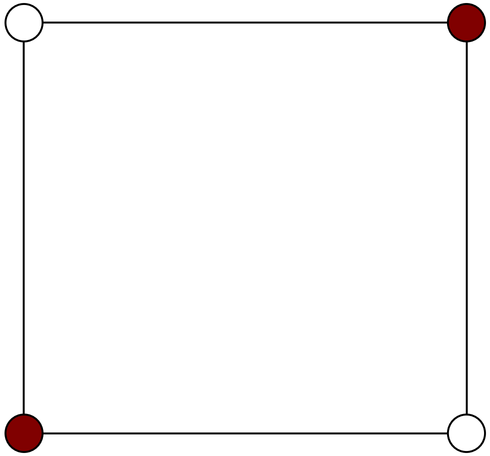


And so is bipartite, if we redraw it:

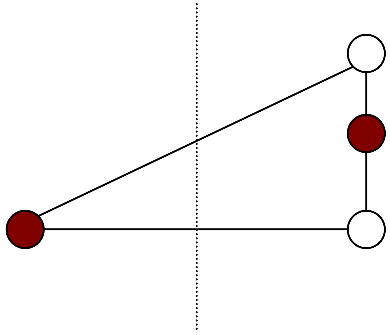


Bipartite Graphs

EG: C_4 is a bichromatic:

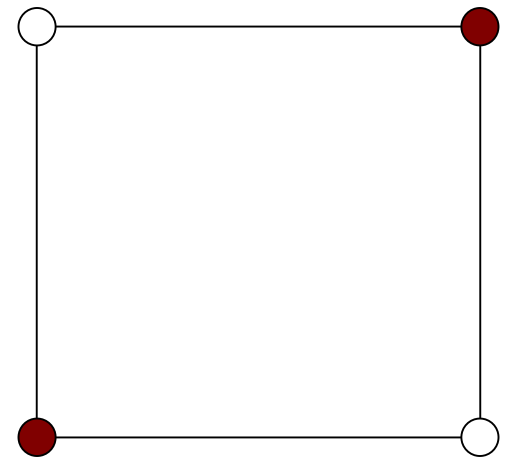


And so is bipartite, if we redraw it:

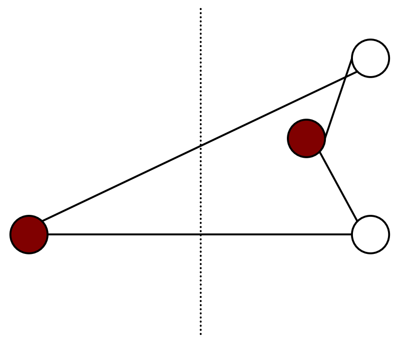


Bipartite Graphs

EG: C_4 is a bichromatic:

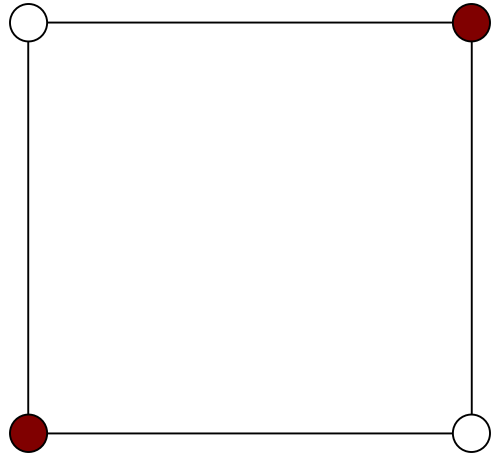


And so is bipartite, if we redraw it:

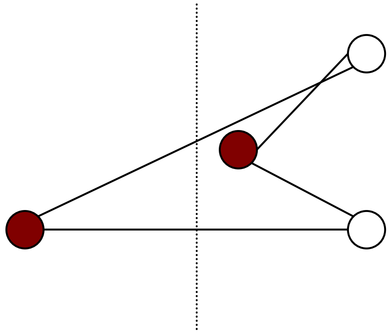


Bipartite Graphs

EG: C_4 is a bichromatic:

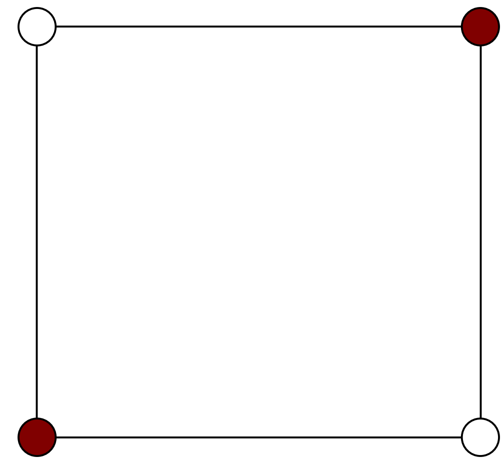


And so is bipartite, if we redraw it:

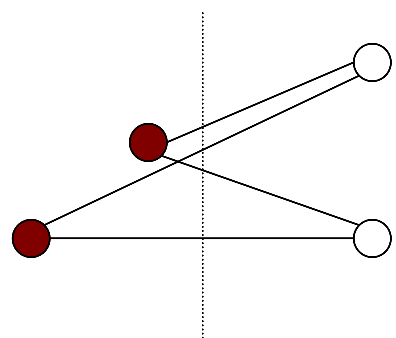


Bipartite Graphs

EG: C_4 is a bichromatic:

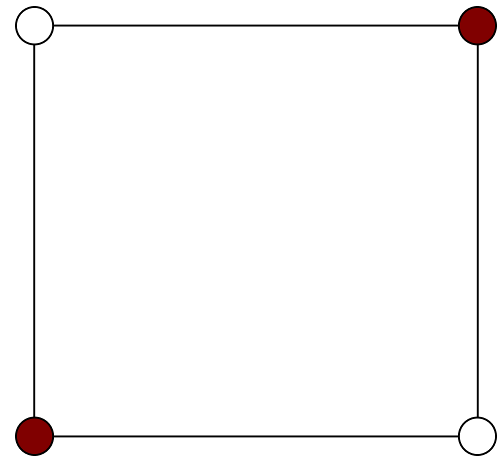


And so is bipartite, if we redraw it:

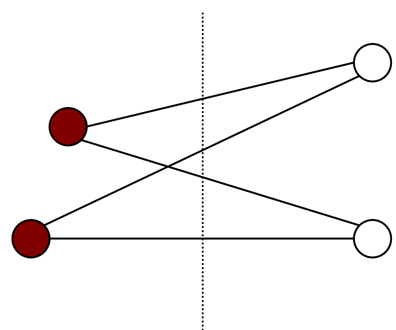


Bipartite Graphs

EG: C_4 is a bichromatic:

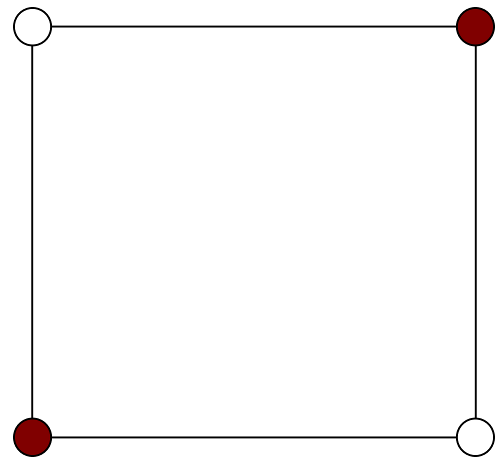


And so is bipartite, if we redraw it:

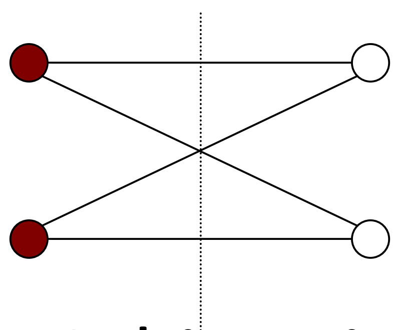


Bipartite Graphs

EG: C_4 is a bichromatic:



And so is bipartite, if we redraw it:



Q: For which n is C_n bipartite?

- 圈图C_n是偶图吗？ Why?

Bipartite Graphs 偶图

A: 圈图C_n is bipartite when n is even. For even n color all odd numbers red and all even numbers green so that vertices are only adjacent to opposite color.

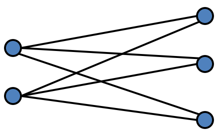
If n is odd, C_n is not bipartite. If it were, color 0 red. So 1 must be green, and 2 must be red. This way, all even numbers must be red, including vertex n-1. But n-1 connects to 0.

完全偶图 Complete Bipartite - $K_{m,n}$

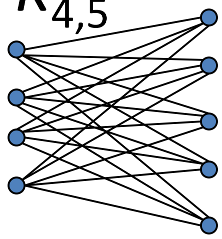
When all possible edges exist in a simple bipartite graph with m red vertices and n green vertices, the graph is called ***complete bipartite*** and the notation $K_{m,n}$ is used.

分别在两个部分的结点集合之间总是有一条边

$K_{2,3}$



$K_{4,5}$



- Think about $K_{3,3}$

偶图的判断

- Question: 如何判断一个图是否是偶图？
- Theorem: a simple graph is bipartite if and only if it is possible to assign one of two different colors to each vertex of the graph so that no two adjacent vertices are assigned to the same color. (2色图）
- Why?
- Some examples for using this theorem to determine whether a given graph is bipartite…

Bipartite Graphs

- **Another theorem**:
  - a bipartite graph is a graph that does not contain any odd-length cycles(回路), or all cycles have even-length.
  - 后面再讲其中的道理

- Why?
- Examples to use this theorem.

偶图的应用

- Modelling matching problems 建模匹配问题. 一个例子就是 Job assignments任务指派。
- Job assignments任务指派: 假定P是一群人的集合， *J是任务集合*, 而且并非所有人适合每一件工作。

建模：model this as a bipartite graph ($P \cup J$, E). If a person p_x is suitable for a certain job j_y there is an edge between p_x and j_y in the graph.

- Examples...

Matching 匹配

- 匹配定义：a **matching *M in a simple graph G = (V ,E) is a subset of the set E of*** edges of the graph such that no two edges are incident with the same vertex （ E的子集满足：没有两条边会关联于相同的结点）.

  In other words, a matching is a subset of edges such that if 边{*s, t*} *and* 边{*u, v*} *are distinct edges of the matching,* then *s, t , u, and v are distinct vertices.*

- A vertex that is the endpoint of an edge of a matching *M is said to* be **matched in *M; otherwise it is said to be unmatched.***
- A **maximum matching (最大匹配)** **is a matching** with the largest number of edges.
- **Complete matching** V_1 **to** V_2 **in bipartite**: if every vertex in V_1 is the endpoint of an edge in the matching, or equivalently, if $|M| = |V_1|$.

偶图中的完全匹配

- 偶图G=（V,E)， $V = V_1 \cup V_2$。V_1 对V_2 的一个完全匹配M是边集E的一个子集，如果它由$|V_1|$条从V_1 **引向**V_2边组成，这些边在V_2中对于不同的结点， 也不关联于V_1的相同结点。

- **HALL'S MARRIAGE THEOREM** (HALL匹配原理) **:**

The bipartite graph G = (V ,E) with bipartition (V1, V2) has a complete matching from V_1 to V_2 iff |N(A)| ≥ |A| for all Subsets A of V1. （其中：N(A) 是所有与A中结点邻接的结点的集合。）

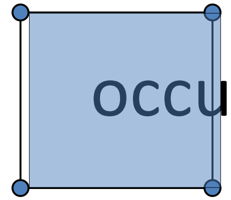
- 求解任务分配可以看成在建立起来的图模型（偶图）上，寻找一个匹配 。 这里不再细讲（自己看书学习）。

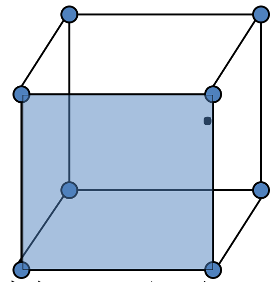
Applications of Bipartite Graph （不讲）

- Bipartite graphs are extensively used in modern Coding theory, especially to decode codewords received from the channel. Factor graphs and Tanner graphs are examples of this.

- In computer science, a Petri net is a mathematical modeling tool used in analysis and simulations of concurrent systems. A system is modeled as a bipartite directed graph with two sets of nodes
- Actually there are more and more applications using bipartite graph model.

Subgraphs 子图

Notice that the 2-cube  occurs

inside the 3-cube 

In other words, Q_2 是Q_3 的子图

定义: Let $G = (V,E)$ and $H = (W,F)$ be graphs. H is said to be a ***subgraph*** of G, if $W \subseteq V$ and $F \subseteq E$.

If V=W, G and its subgraph H have same vertex set, H is called a **spanning subgraph** of G(生成子图)

Examples…

- 思考问题：
- 如果一个图有一个子图不是偶图，那这个图是否一定不是偶图？为什么？

Unions 图的并

DEF: Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ 是两个简单图 (and V_1,V_2 may or may not be disjoint). The ***union*** of G_1, G_2 规定为 $G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$.

类似的定义也针对有向图、多重图、伪图等给出。

问题：G_1、G_2分别是并图的什么？

练习

- 第6章第2节练习
- T10, T13, T21 (c), (e), T24

- 此课件两次课