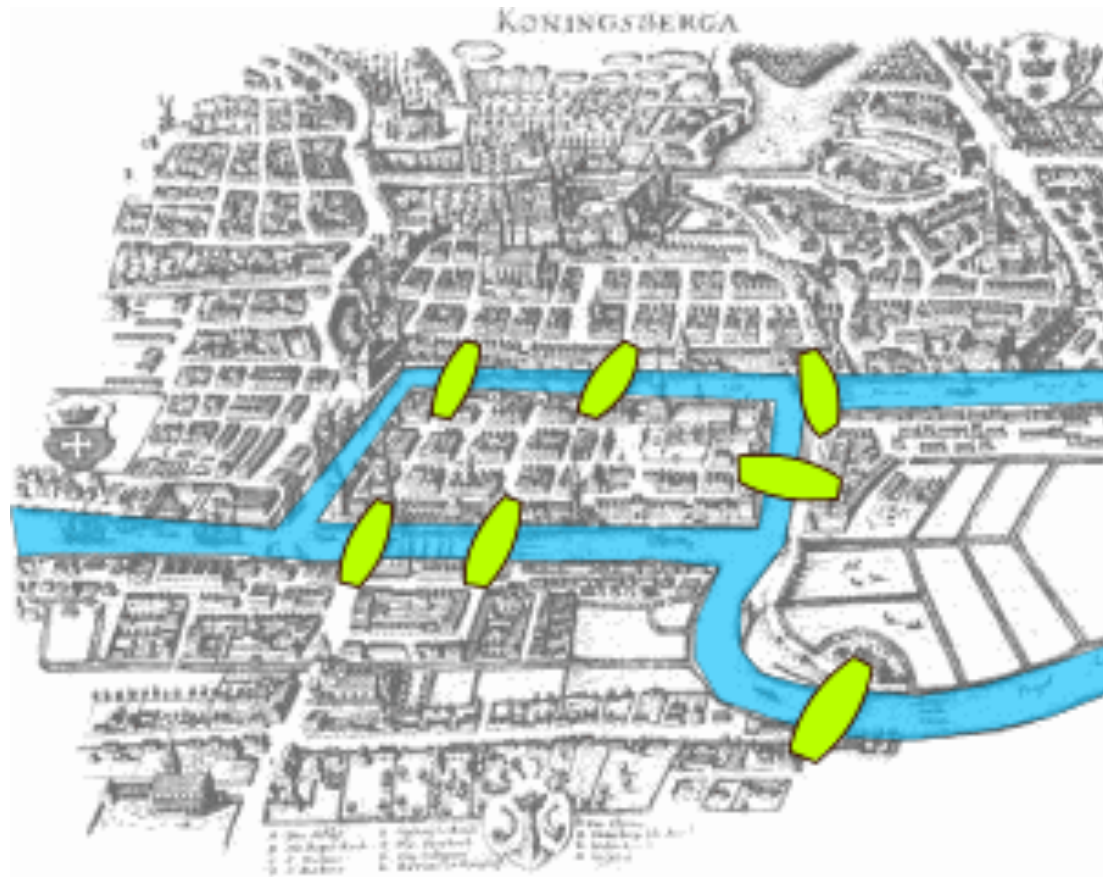


# Introduction to Graphs



Slides by Lap Chi Lau

The Chinese University of Hong Kong

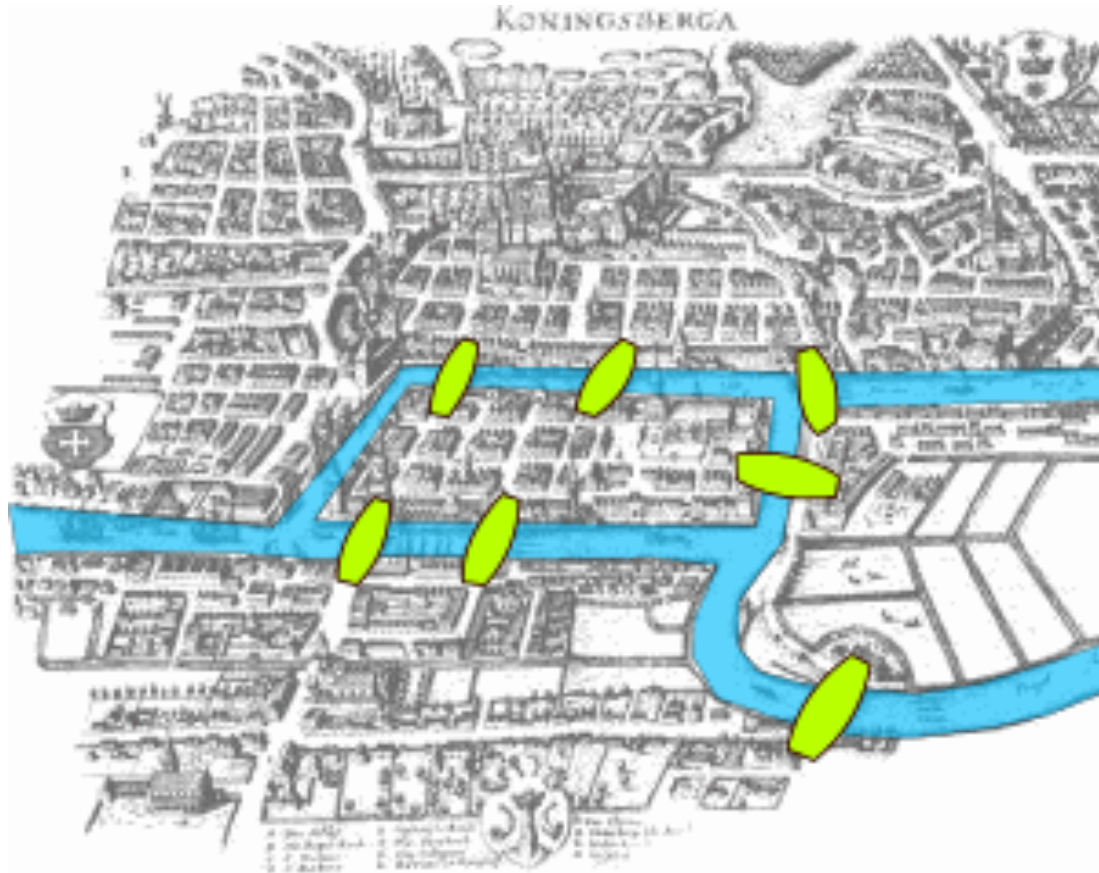
# This Lecture

In this part we will study some basic graph theory.

Graph is a useful concept to model many problems in computer science.

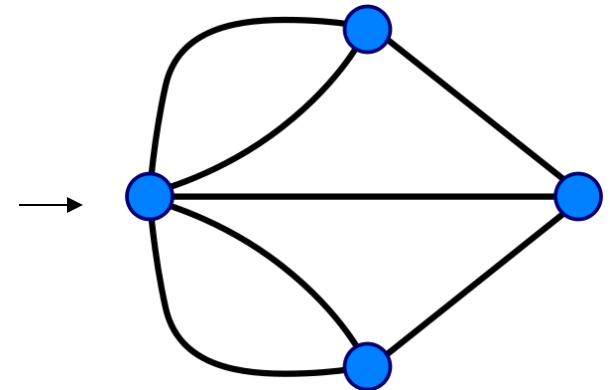
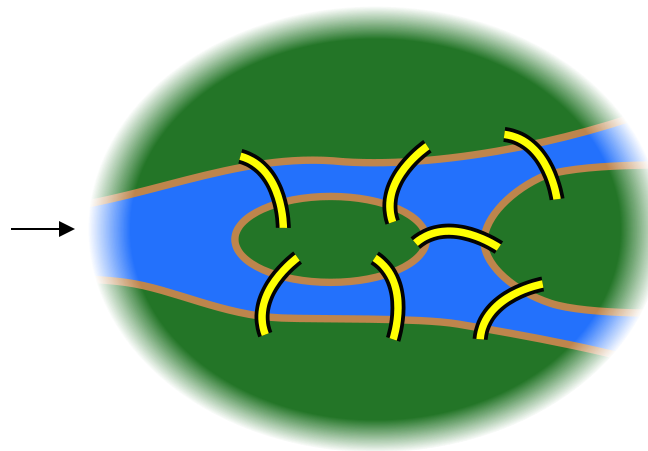
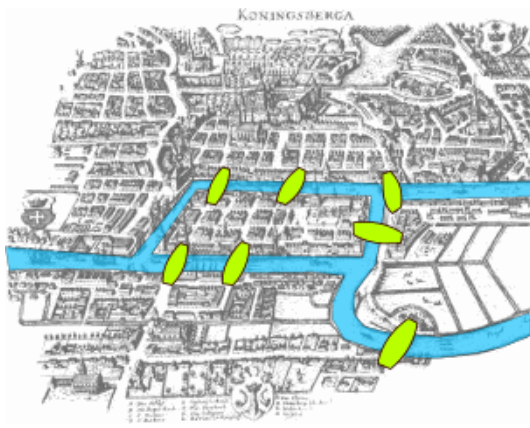
- Seven bridges of Königsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Graphs and networks
- Graph coloring

# Seven Bridges of Königsberg



Is it possible to walk with a route that crosses each bridge exactly once?

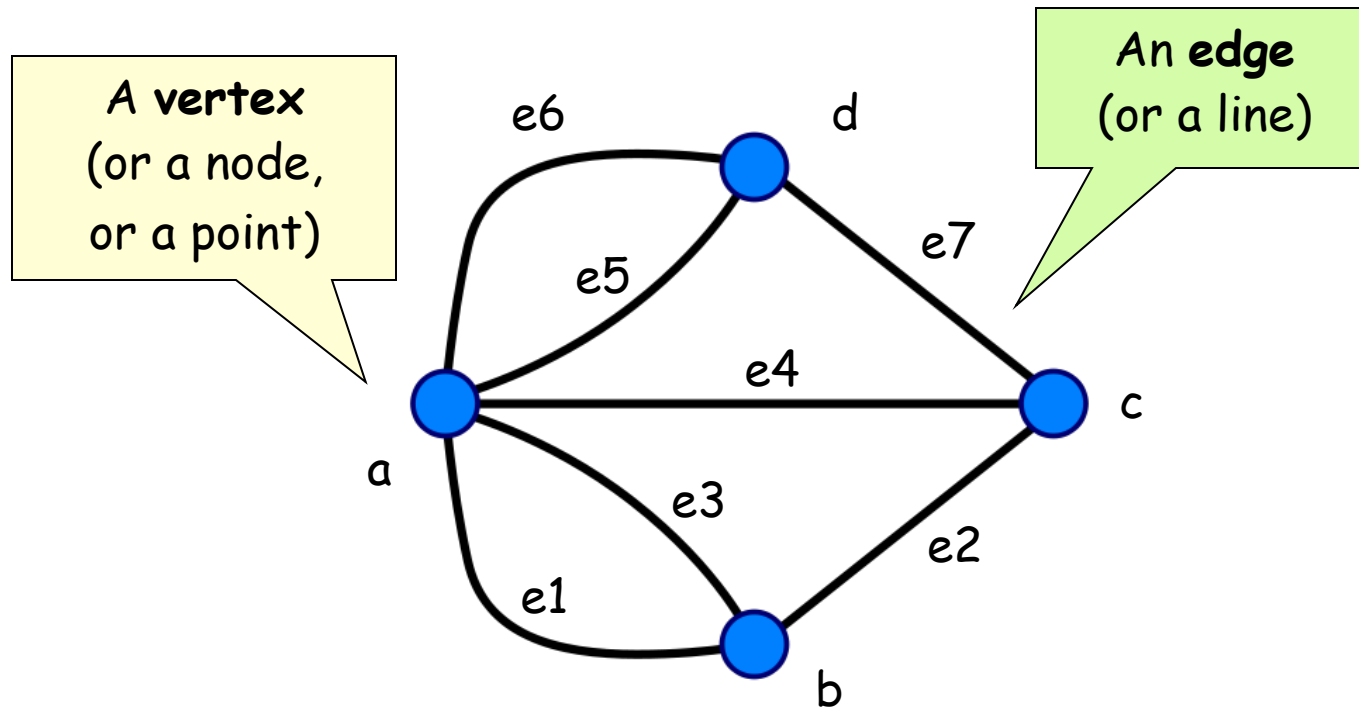
# Seven Bridges of Königsberg



Forget unimportant details.

Forget even more.

# A Graph

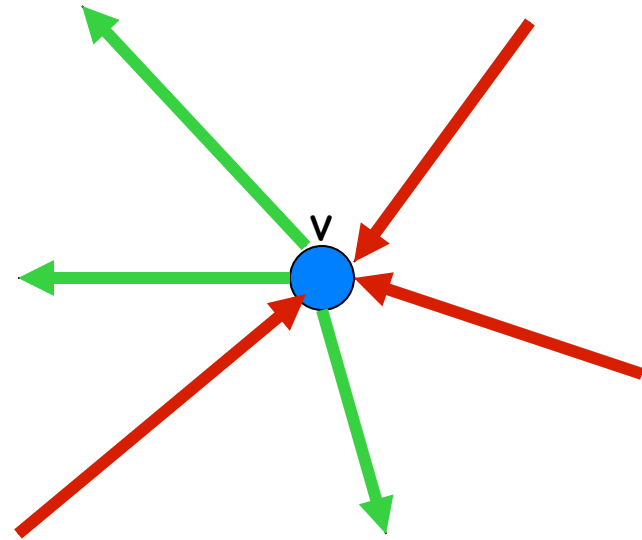
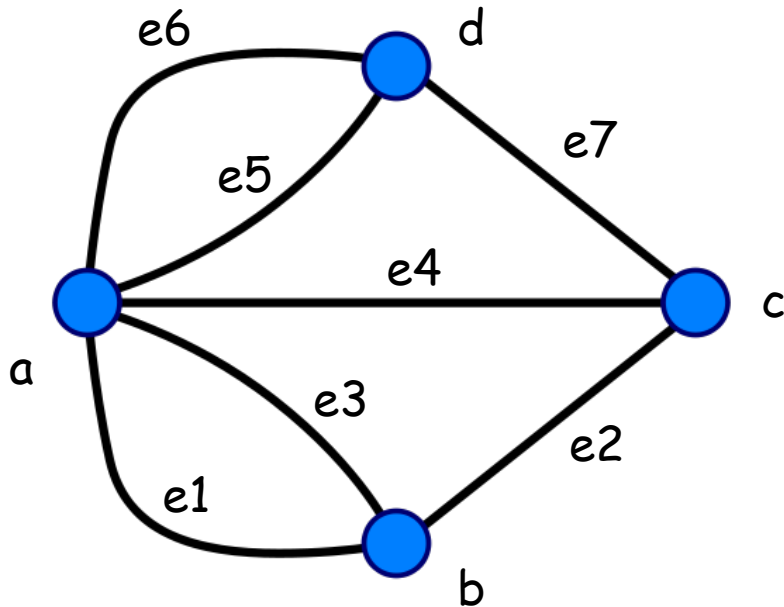


So, what is the “Seven Bridges of Königsberg” problem now?

To find a walk that visits each edge exactly once.

# Euler's Solution

**Question:** Is it possible to find a walk that visits each edge exactly once.

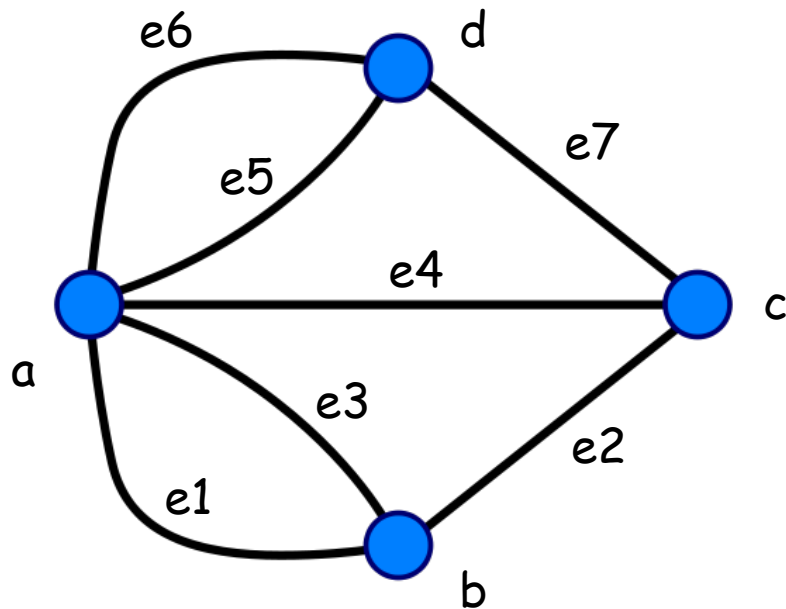


Suppose there is such a walk, there is a starting point and an endpoint point.

For every "intermediate" point  $v$ , there must be the same number of incoming and outgoing edges, and so  $v$  must have an **even number of edges**.

# Euler's Solution

**Question:** Is it possible to find a walk that visits each edge exactly once.



So, at most **two** vertices can have odd number of edges.

In this graph, every vertex has only an odd number of edges, and so there is no walk which visits each edge exactly one.

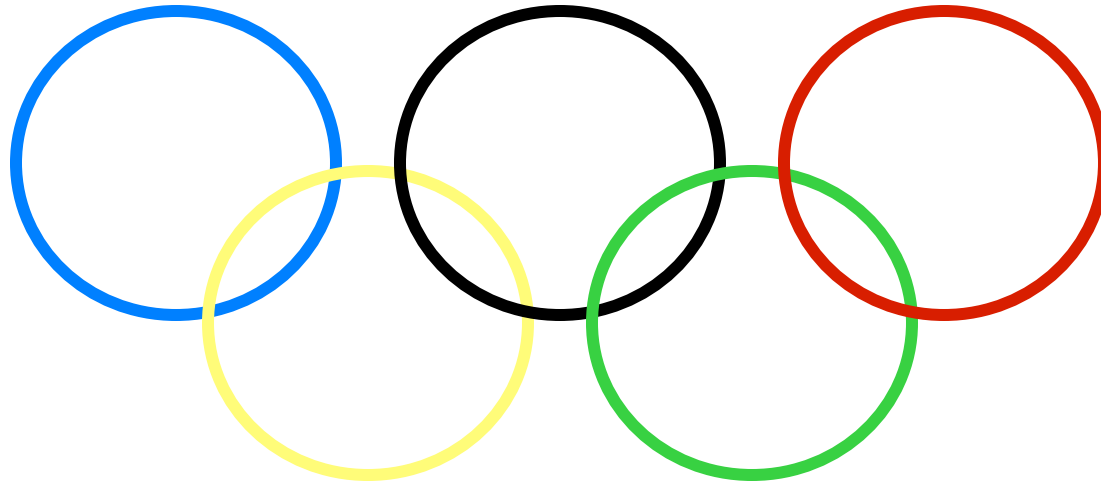
Suppose there is such a walk, there is a starting point and an endpoint point.

For every "intermediate" point  $v$ , there must be the same number of incoming and outgoing edges, and so  $v$  must have an **even number of edges**.

# Euler's Solution

So Euler showed that the “**Seven Bridges of Königsberg**” is unsolvable.

When is it possible to have a walk that visits every edge exactly once?



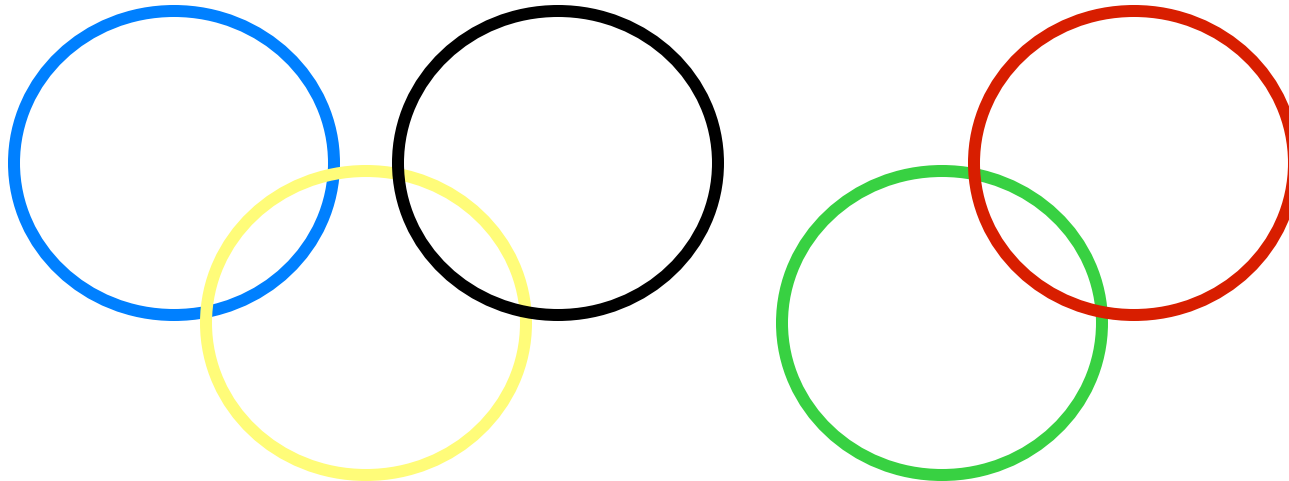
Is it always possible to find such a walk if there is **at most two** vertices with odd number of edges?



# Euler's Solution

So Euler showed that the “**Seven Bridges of Königsberg**” is unsolvable.

When is it possible to have a walk that visits every edge exactly once?



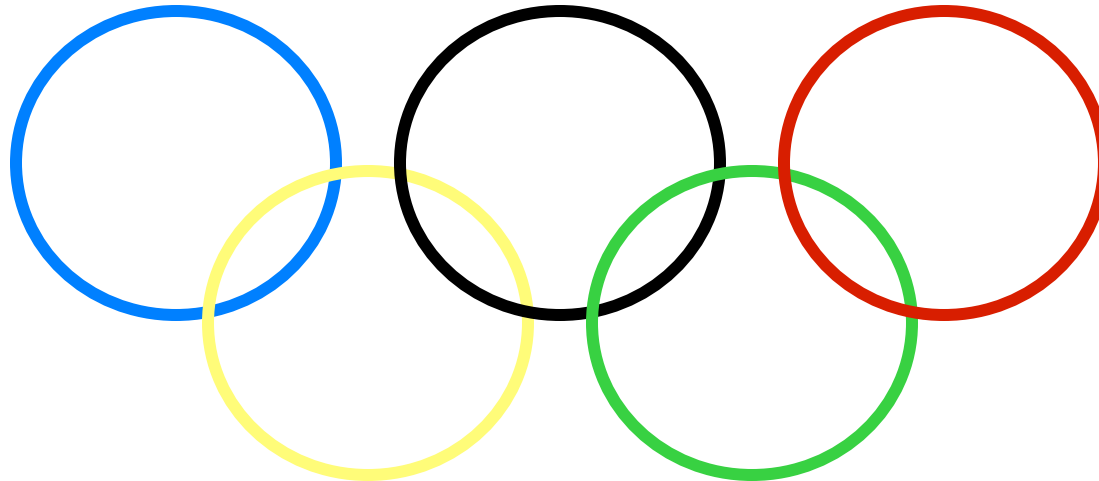
Is it always possible to find such a walk if there is **at most two** vertices with odd number of edges?

NO!

# Euler's Solution

So Euler showed that the “Seven Bridges of Königsberg” is unsolvable.

When is it possible to have a walk that visits every edge exactly once?



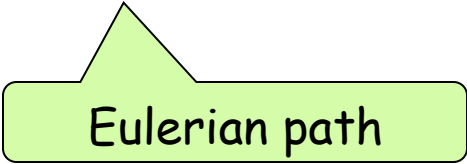
Is it always possible to find such a walk if the graph is “connected” and there are **at most two** vertices with odd number of edges?

YES!

# Euler's Solution

So Euler showed that the “**Seven Bridges of Königsberg**” is unsolvable.

When is it possible to have a walk that visits every edge exactly once?



Eulerian path

**Euler's theorem:** A graph has an Eulerian path if and only if it is “connected” and has at most two vertices with an odd number of edges.

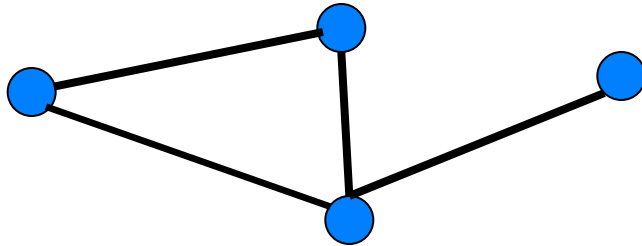
This theorem was proved in 1736,  
and was regarded as the starting point of graph theory.

# This Lecture

- Seven bridges of Königsberg
- **Graphs, degrees**
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Graphs and networks
- Graph coloring

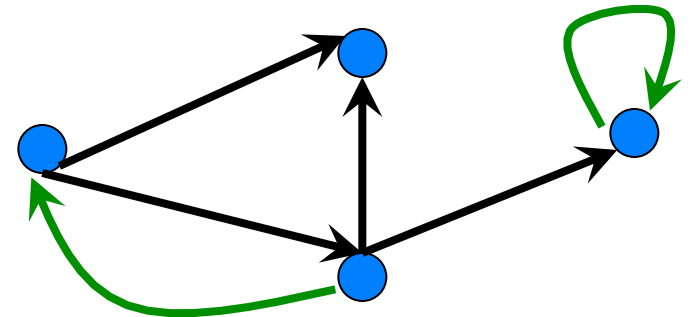
# Types of Graphs

Simple Graph



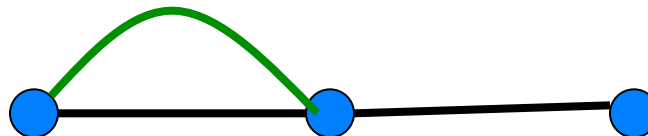
Most of the problems in this course.

Directed Graph



Will not see in this course

Multi-Graph



Eulerian path problem

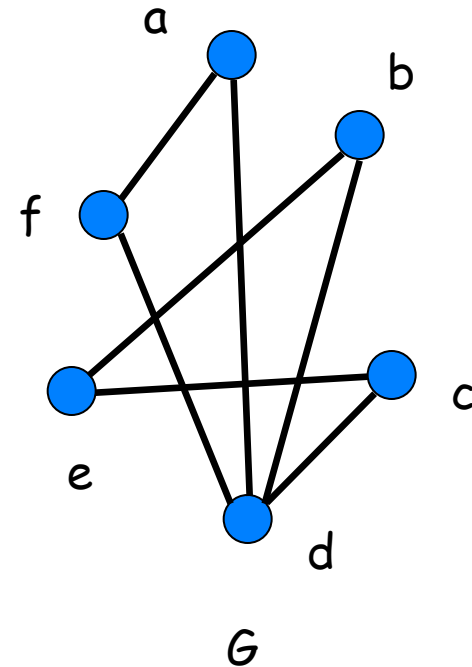
# Simple Graphs

A graph  $G=(V,E)$  consists of:

A set of vertices,  $V$

A set of *undirected* edges,  $E$

- $V(G) = \{a,b,c,d,e,f\}$
- $E(G) = \{ad,af,bd,be,cd,ce,df\}$



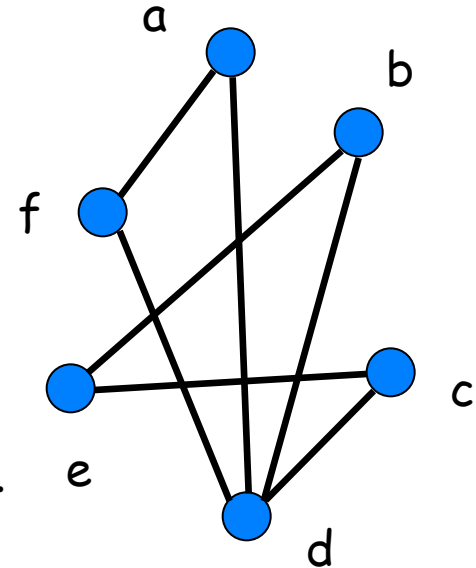
Two vertices  $u,v$  are **adjacent** (**neighbours**) if the edge  $uv$  is present.

# Vertex Degrees

An edge  $uv$  is **incident** on the vertex  $u$  and the vertex  $v$ .

The **neighbour set**  $N(v)$  of a vertex  $v$  is the set of vertices adjacent to it.

e.g.  $N(a) = \{d, f\}$ ,  $N(d) = \{a, b, c, f\}$ ,  $N(e) = \{b, c\}$ .



**degree** of a vertex = # of **incident** edges

e.g.  $\deg(d) = 4$ ,  $\deg(a) = \deg(b) = \deg(c) = \deg(e) = \deg(f) = 2$ .

the degree of a vertex  $v$  = the number of neighbours of  $v$ ?

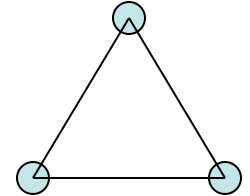
For multigraphs, **NO**.

For simple graphs, **YES**.

# Degree Sequence

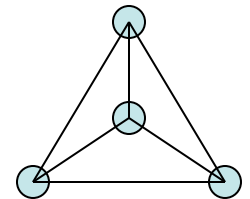
Is there a graph with degree sequence  $(2,2,2)$ ?

YES.



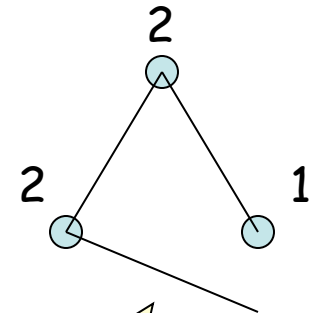
Is there a graph with degree sequence  $(3,3,3,3)$ ?

YES.



Is there a graph with degree sequence  $(2,2,1)$ ?

NO.



Is there a graph with degree sequence  $(2,2,2,2,1)$ ?

NO.

What's wrong with these sequences?

Where to go?



# Handshaking Lemma

For any graph, sum of degrees = twice # edges

Lemma.

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

Corollary.

1. Sum of degree is an even number.
2. Number of odd degree vertices is even.

Examples.

$2+2+1 = \text{odd}$ , so impossible.

$2+2+2+2+1 = \text{odd}$ , so impossible.

# Handshaking Lemma

Lemma.

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

Proof. Each edge contributes 2 to the sum on the right. Q.E.D.

**Question.** Given a degree sequence, if the sum of degree is even, is it true that there is a graph with such a degree sequence?

For simple graphs, **NO**, consider the degree sequence (3,3,3,1).

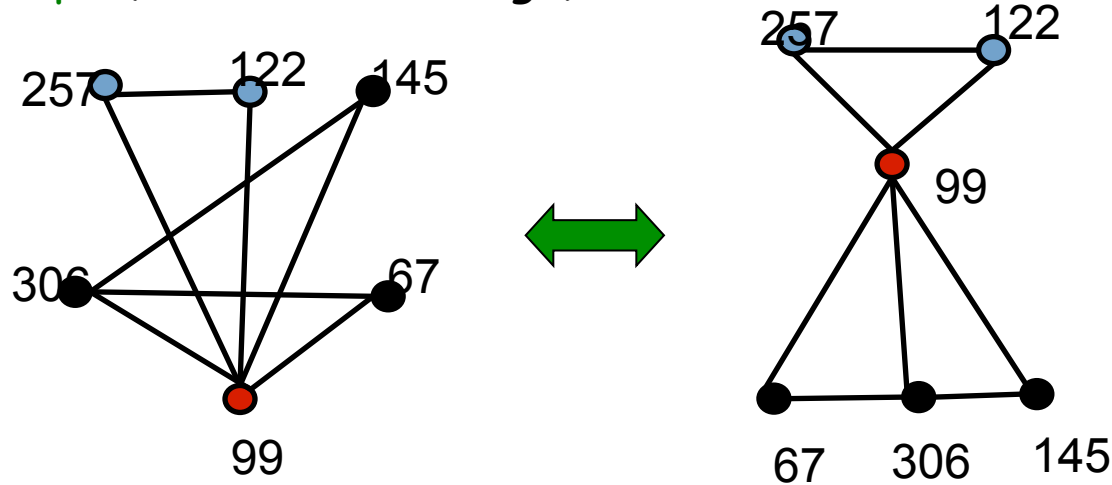
For multigraphs (with self loops), **YES!**

# This Lecture

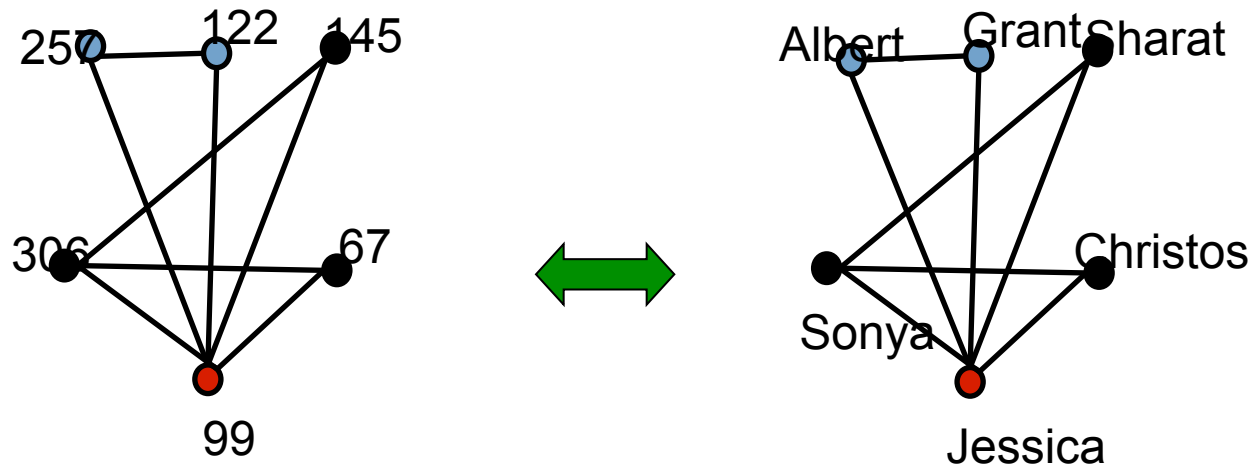
- Seven bridges of Königsberg
- Graphs, degrees
- **Isomorphism**
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Graphs and networks
- Graph coloring

# Same Graphs?

Same graph (different *drawings*)



Same graph (different *labels*)



# Graph Isomorphism

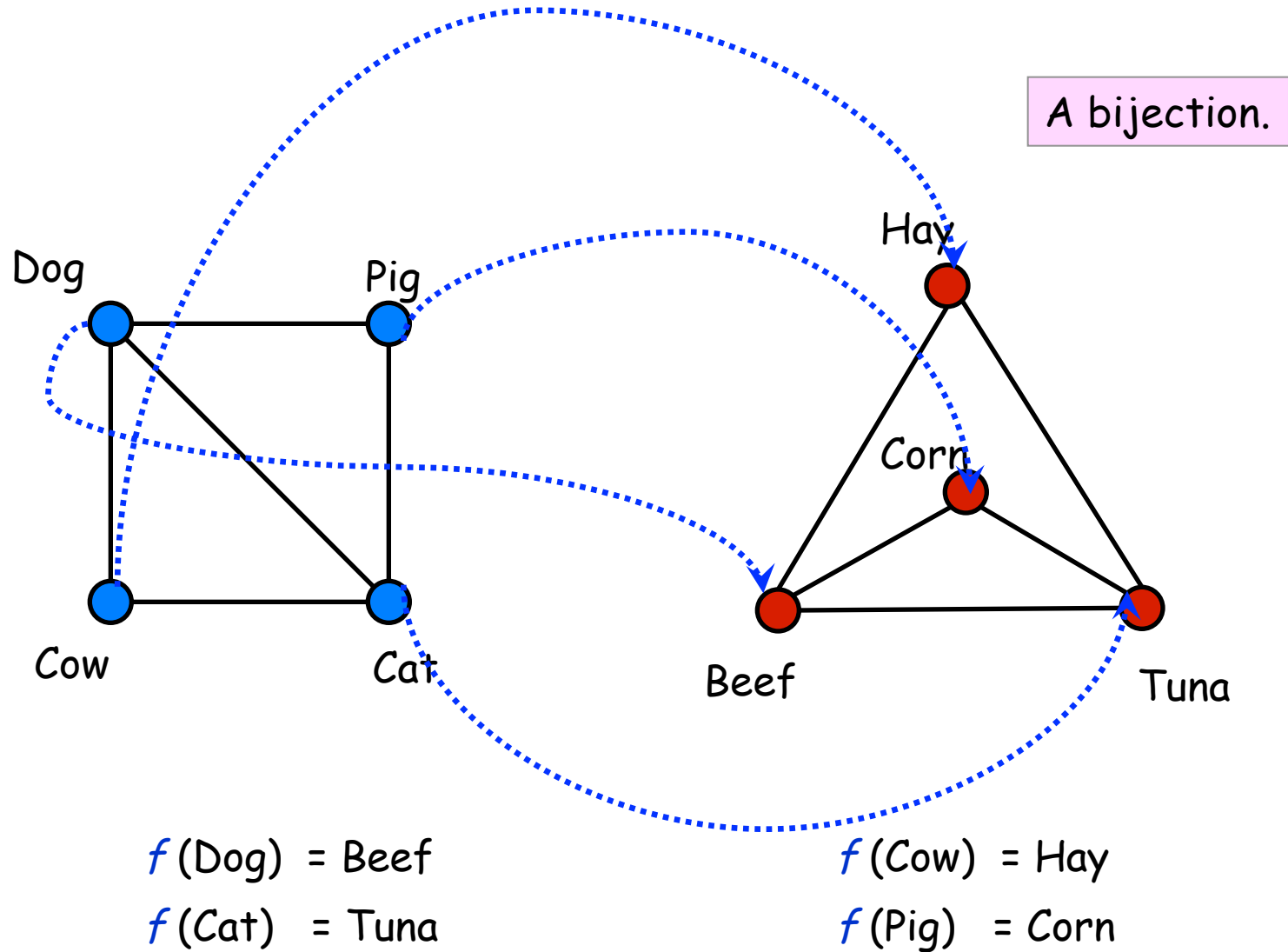
All that matters is the *connections*.

Graphs with the same connections are *isomorphic*.

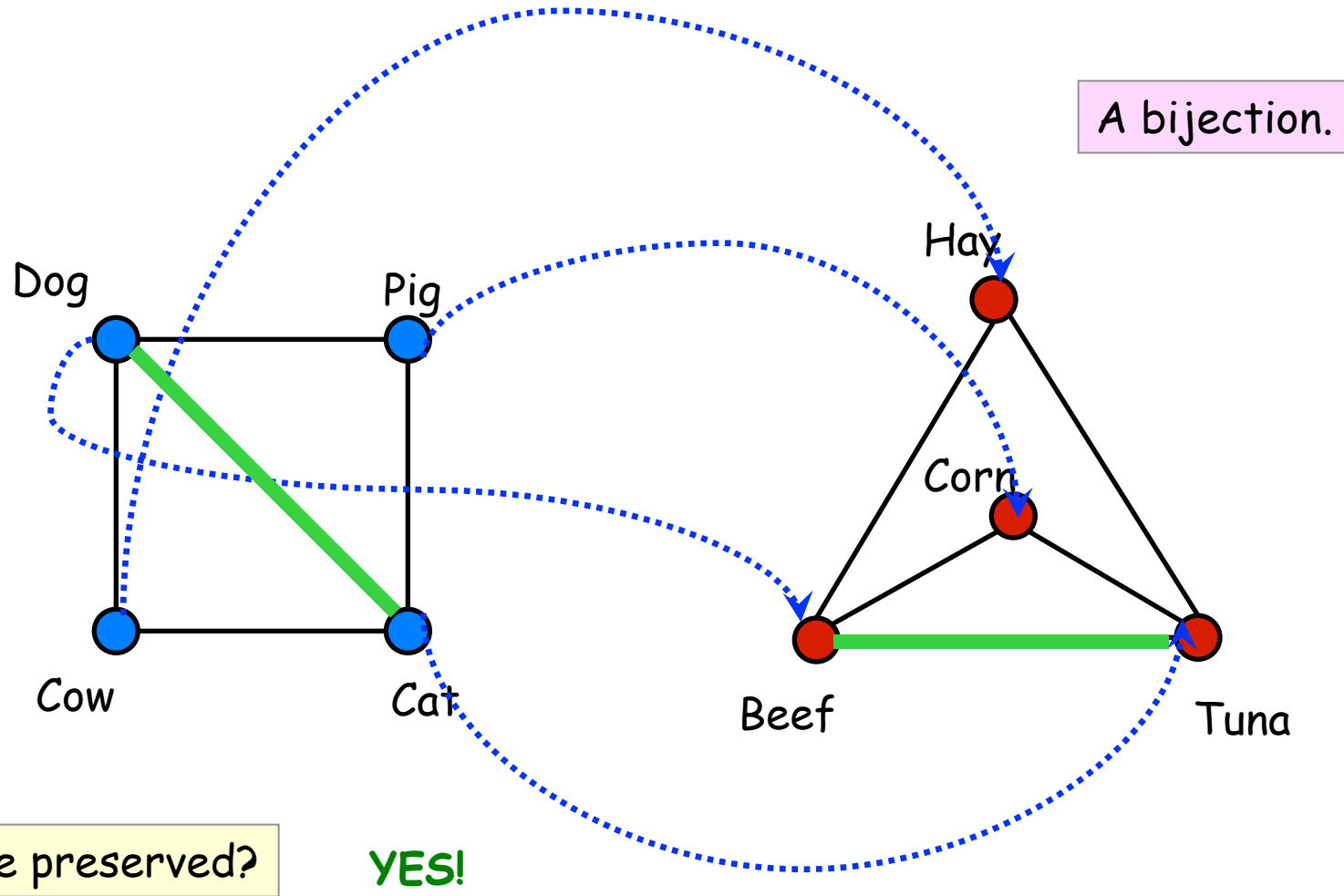
Informally, two graphs are isomorphic if they are the same after *renaming*.

Graph isomorphism has applications like checking fingerprint, testing molecules...

# Are These Isomorphic?

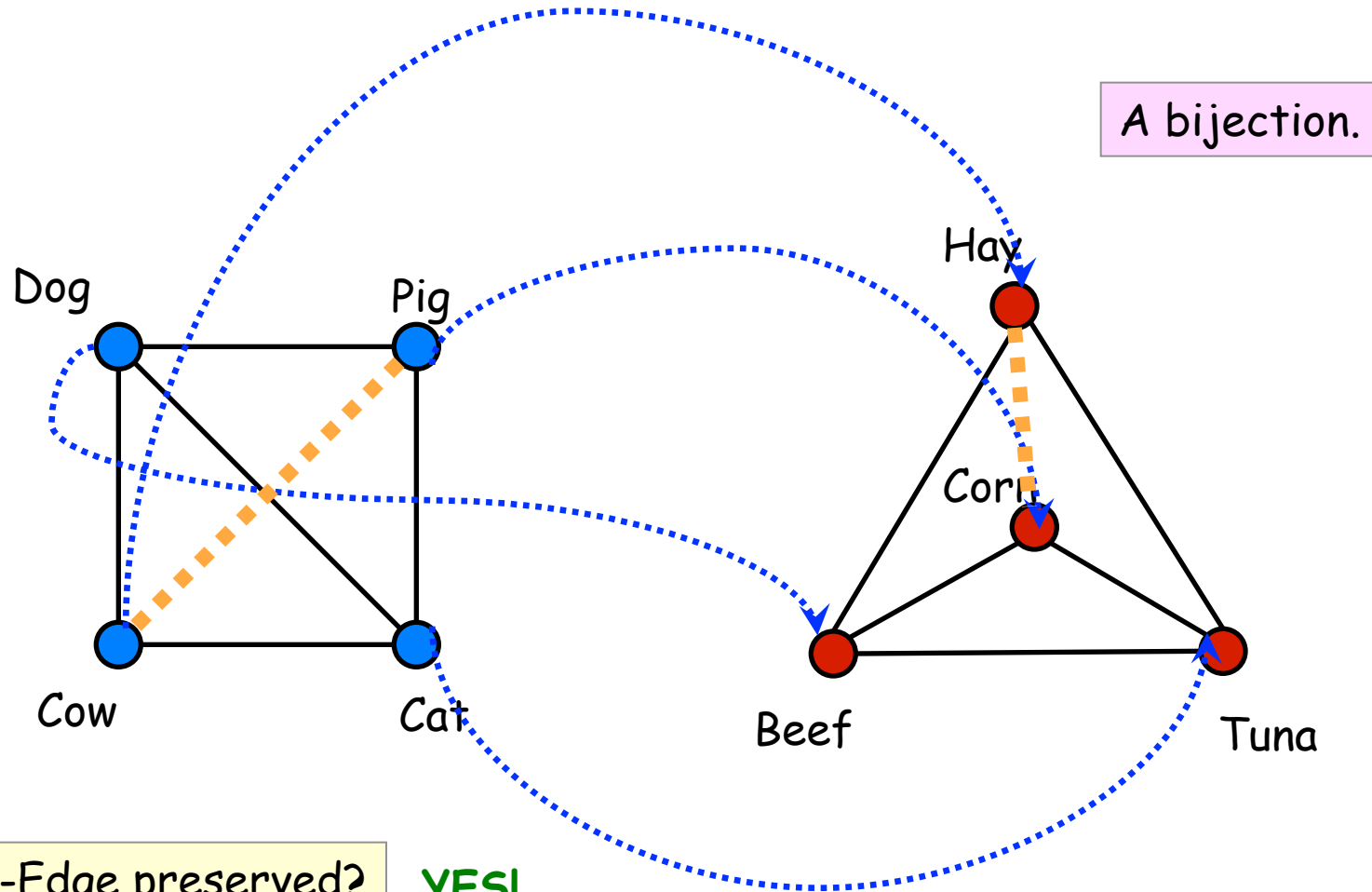


# Are These Isomorphic?



If there is an edge in the original graph, there is an edge after the mapping.

# Are These Isomorphic?



If there is **no** edge in the original graph, there is **no** edge after the mapping.



# Graph Isomorphism

$G_1$  *isomorphic* to  $G_2$  means there is an *edge-preserving vertex matching*.

bijection  $f: V_1 \rightarrow V_2$   
 $u-v$  in  $E_1$  iff  $f(u)-f(v)$  in  $E_2$

$uv$  is an edge in  $G_1$

$f(u)f(v)$  is an edge in  $G_2$

- If  $G_1$  and  $G_2$  are isomorphic, do they have the same number of vertices? YES
- If  $G_1$  and  $G_2$  are isomorphic, do they have the same number of edges? YES
- If  $G_1$  and  $G_2$  are isomorphic, do they have the same degree sequence? YES
- If  $G_1$  and  $G_2$  have the same degree sequence, are they isomorphic? NO

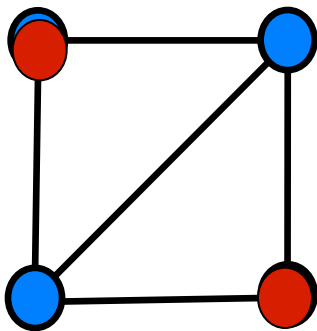
# Checking Graph Isomorphism

How to show two graphs are isomorphic?

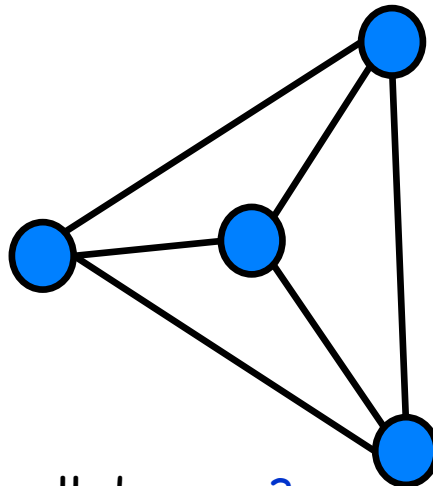
Find a mapping and show that it is edge-preserving.

How to show two graphs are non-isomorphic?

Find some **isomorphic-preserving properties** which is satisfied in one graph but not the other.



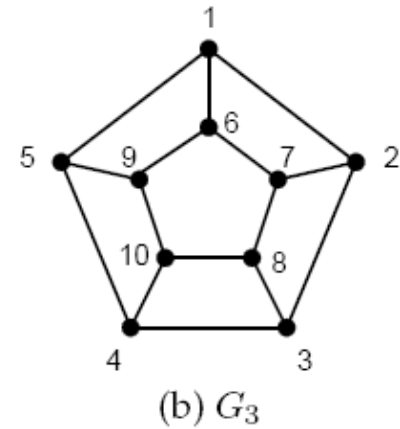
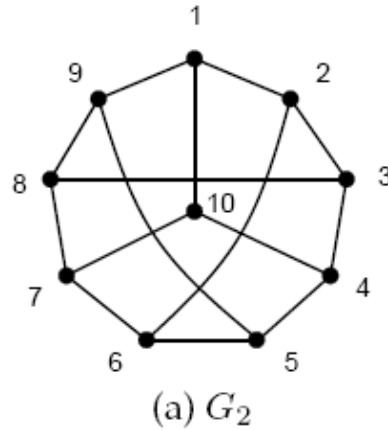
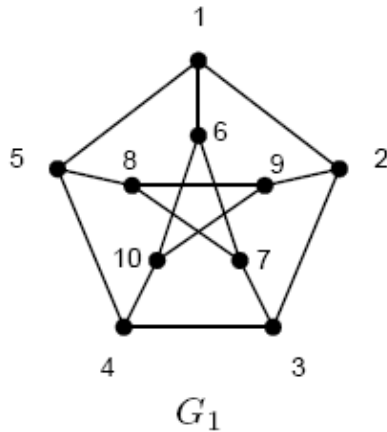
degree 2



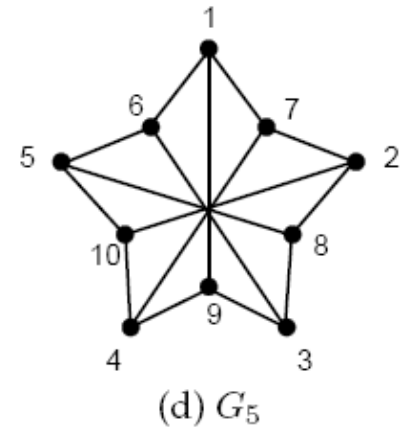
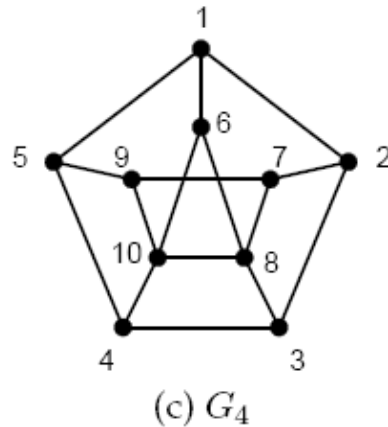
all degree 3

Non-isomorphic

# Exercise



Which is isomorphic to  $G_1$ ?



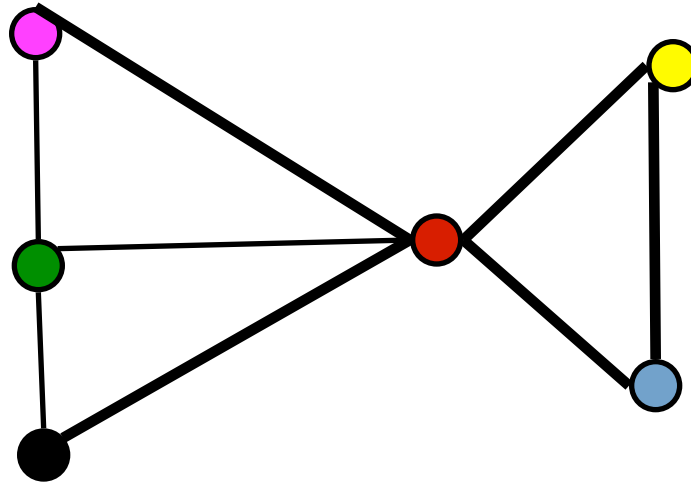
***Testing graph isomorphism is not easy -***

No known general method to test graph isomorphism  
much more efficient than checking all possibilities.

# This Lecture

- Seven bridges of Königsberg
- Graphs, degrees
- Isomorphism
- **Path, cycle, connectedness**
- Tree
- Eulerian cycle
- Graphs and networks
- Graph coloring

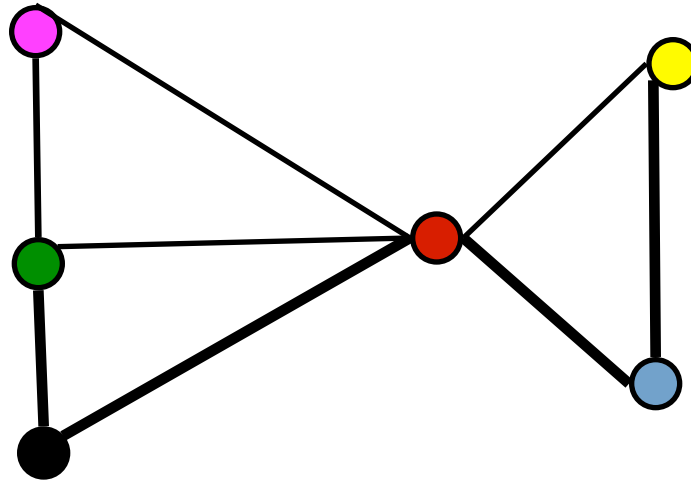
# Paths



*Path*: sequence of *adjacent* vertices

( ● ● ● ● ● )

# Simple Paths



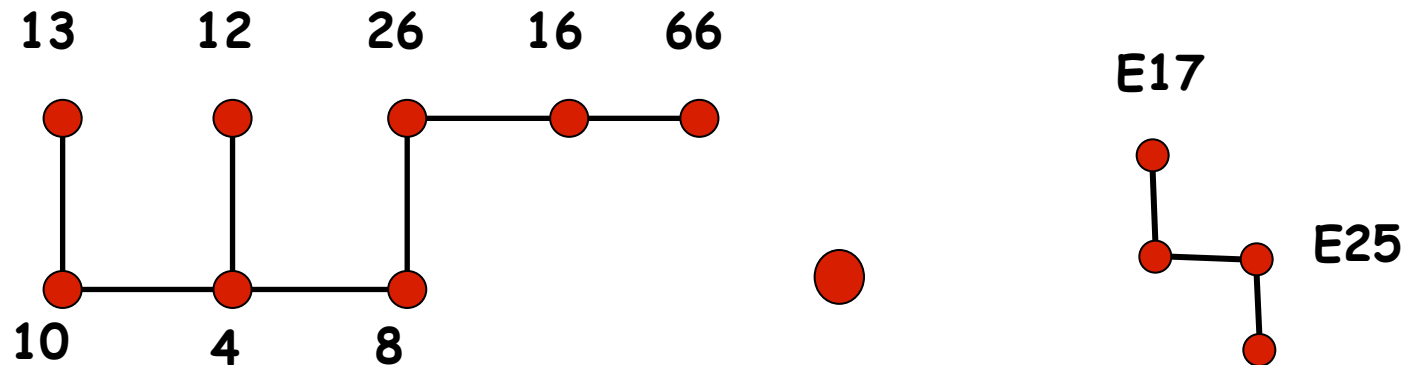
*Simple Path*: all vertices different

(      )

# Connectedness

- ❖ Vertices  $v$ ,  $w$  are *connected* if and only if there is a path starting at  $v$  and ending at  $w$ .
- ❖ A *graph* is *connected* iff every pair of vertices are connected.

Every graph consists of separate connected pieces called *connected components*

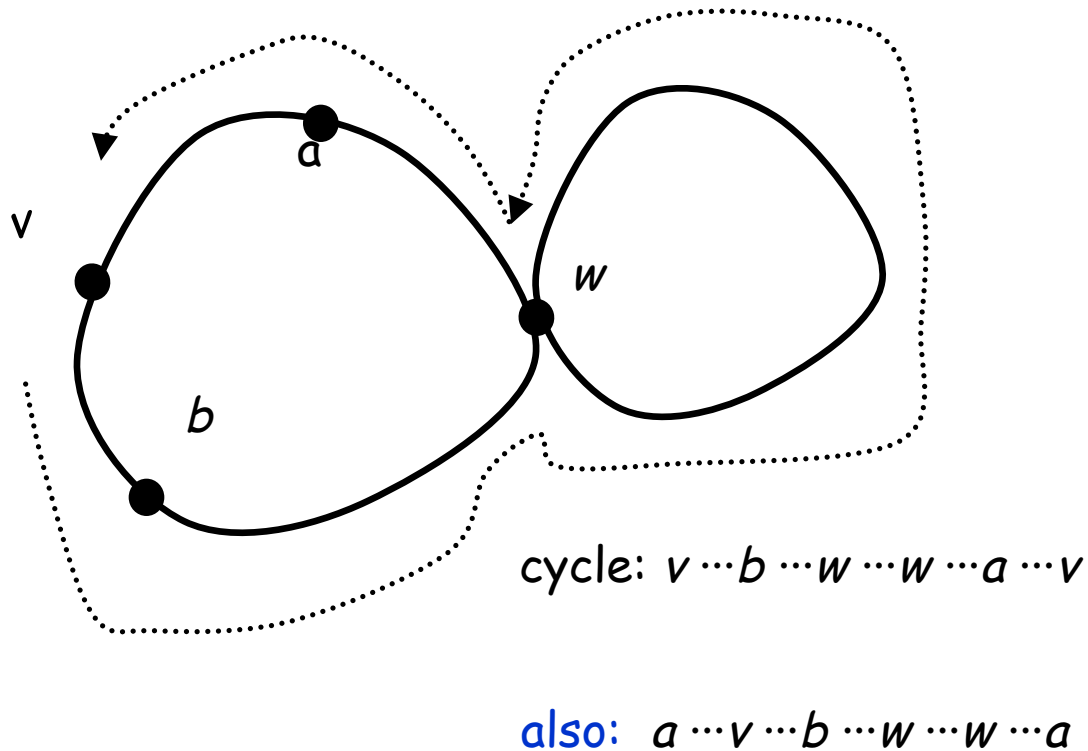


3 connected components

So a graph is *connected* if and only if it has only 1 connected component.

# Cycles

A *cycle* is a path that begins and ends with same vertex.

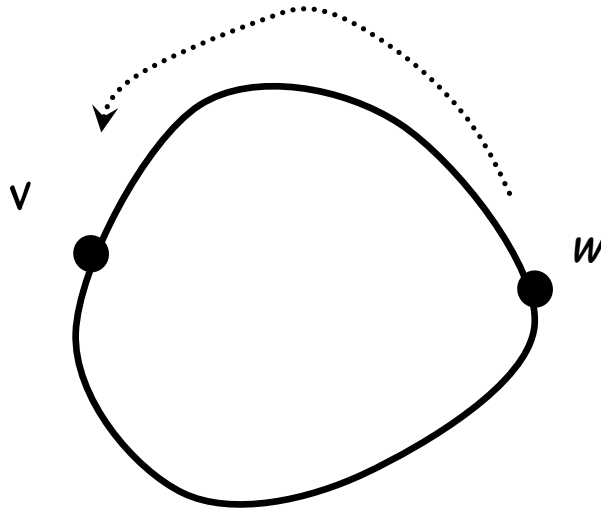




# Simple Cycles

A simple *cycle* is a cycle that doesn't cross itself

In a simple cycle, every vertex is of degree exactly 2.



cycle:  $v \cdots w \cdots v$

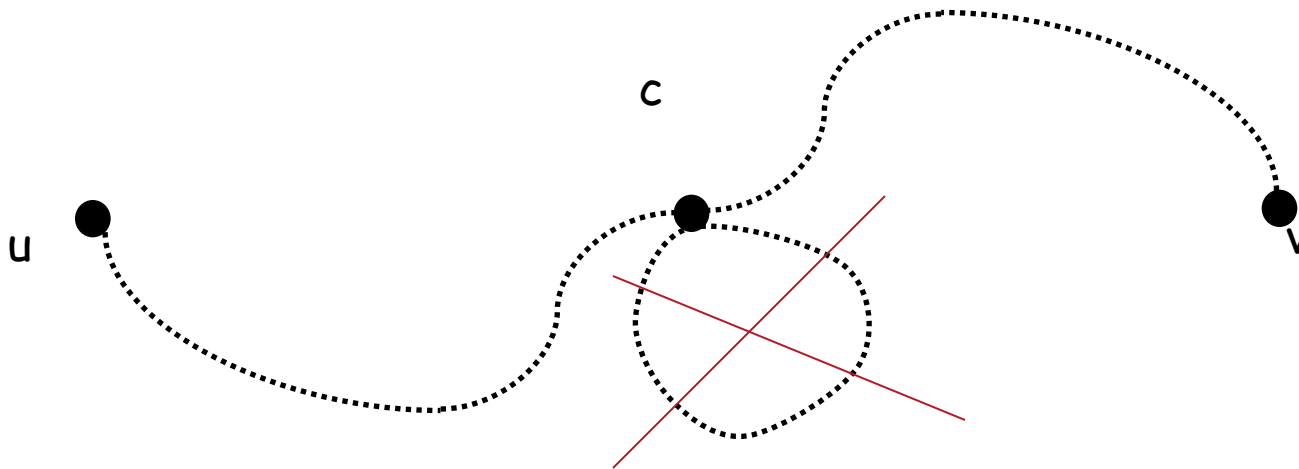
also:  $w \cdots v \cdots w$

# Shortest Paths

A path between  $u$  and  $v$  is a *shortest path* if among all  $u$ - $v$  paths it uses the minimum number of edges.

Is a shortest path between two vertices always simple?

Idea: remove the cycle will make the path shorter.

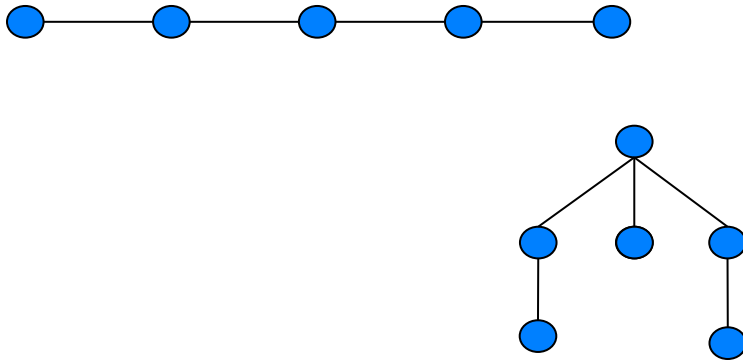


# This Lecture

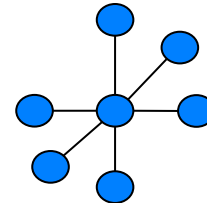
- Seven bridges of Königsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- **Tree**
- Eulerian cycle
- Graphs and networks
- Graph coloring

# Tree

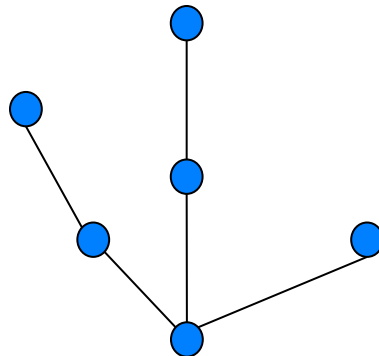
Graphs with no cycles?



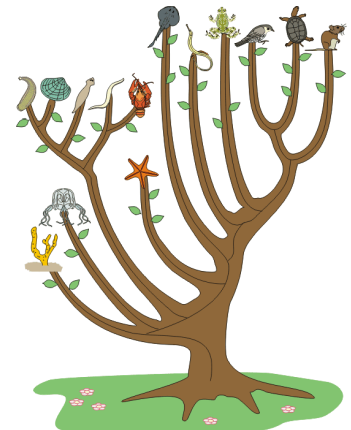
A forest.



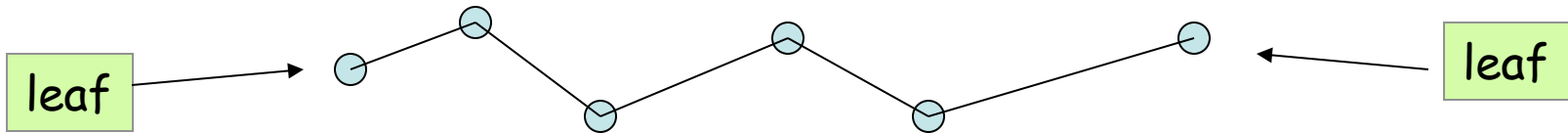
Connected graphs with no cycles?



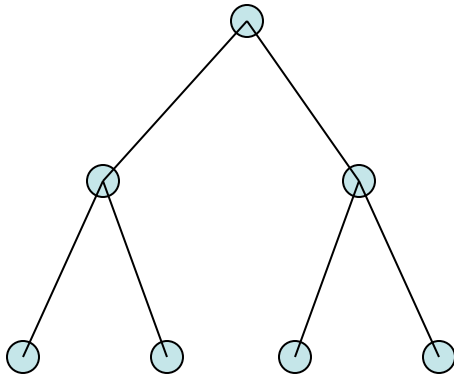
A tree.



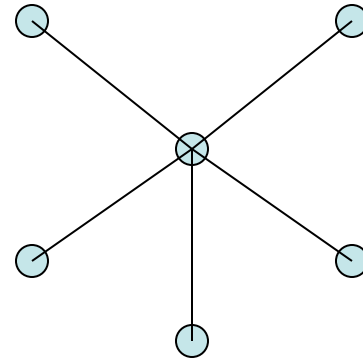
# More Trees



A leaf is a vertex of degree 1.



More leaves.



Even more leaves.

# Tree Characterization by Path

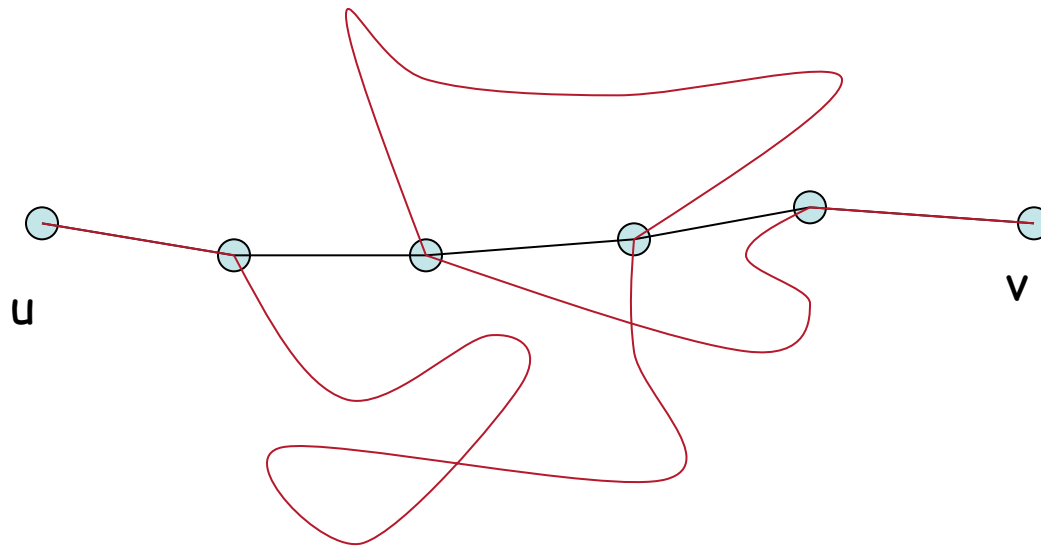
**Definition.** A tree is a connected graph with no cycles.

Can there be no path between  $u$  and  $v$ ?

NO

Can there be more than one simple path between  $u$  and  $v$ ?

NO



This will create cycles.

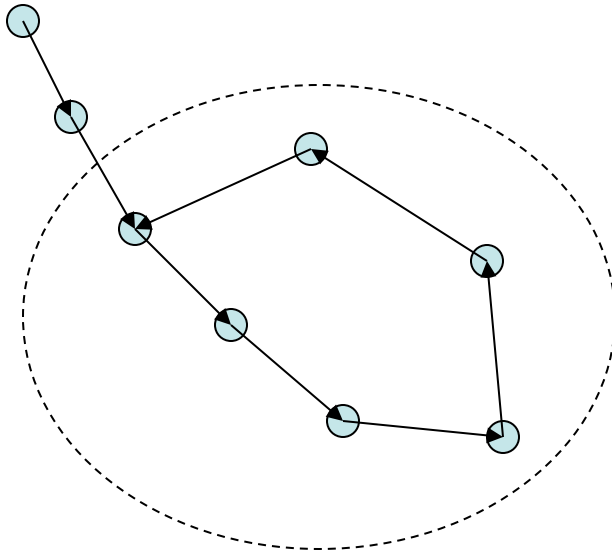
**Claim.** In a tree, there is a unique simple path between every pair of vertices.

# Tree Characterization by Number of Edges

**Definition.** A tree is a connected graph with no cycles.

Can a tree have no leaves? **NO**

Then every vertex has degree at least 2.



Go to unvisited edges as long as possible.

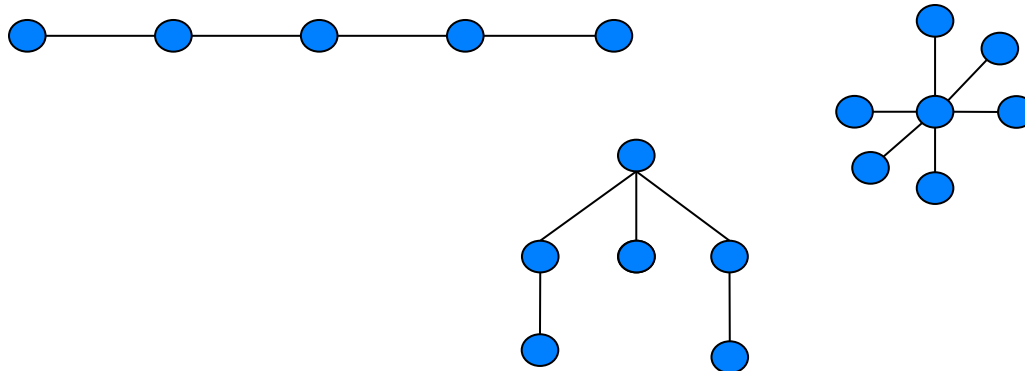
Cannot get stuck,  
unless there is a cycle.

# Tree Characterization by Number of Edges

**Definition.** A tree is a connected graph with no cycles.

Can a tree have no leaves? **NO**

How many edges does a tree have?  $n-1$ ?



We usually use  $n$  to denote the number of vertices,  
and use  $m$  to denote the number of edges in a graph.

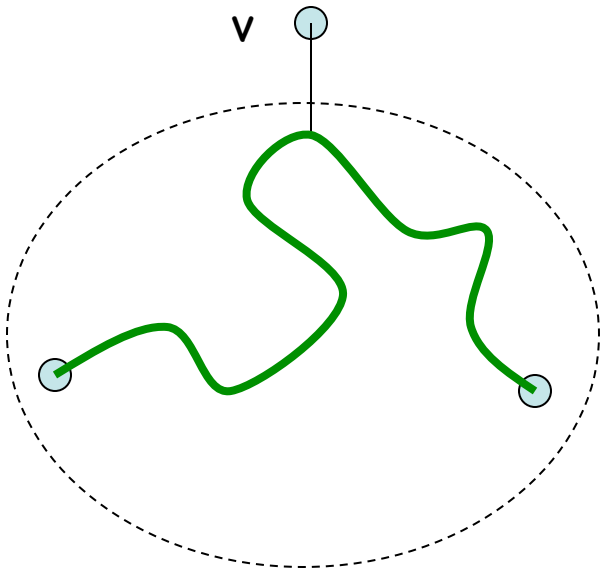


# Tree Characterization by Number of Edges

**Definition.** A tree is a connected graph with no cycles.

Can a tree have no leaves? **NO**

How many edges does a tree have?  $n-1$ ?



Look at a leaf  $v$ .

Is  $T-v$  a tree? **YES**

1. Can  $T-v$  have a cycle? **NO**
2. Is  $T-v$  connected? **YES**

By induction,  $T-v$  has  $(n-1)-1=n-2$  edges.

So  $T$  has  $n-1$  edges.

# Tree Characterizations

**Definition.** A tree is a connected graph with no cycles.

## Characterization by paths:

A graph is a tree if and only if  
there is a unique simple path between every pair of vertices.

## Characterization by number of edges:

A graph is a tree if and only if it is connected and has  $n-1$  edges.

# This Lecture

- Seven bridges of Königsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- **Eulerian cycle**
- Graphs and networks
- Graph coloring

# Eulerian Graphs

**Euler's theorem:** A graph has an Eulerian path if and only if it is connected and has at most two vertices with an odd number of edges.

Can a graph have only 1  
odd degree vertex?

Odd degree vertices.

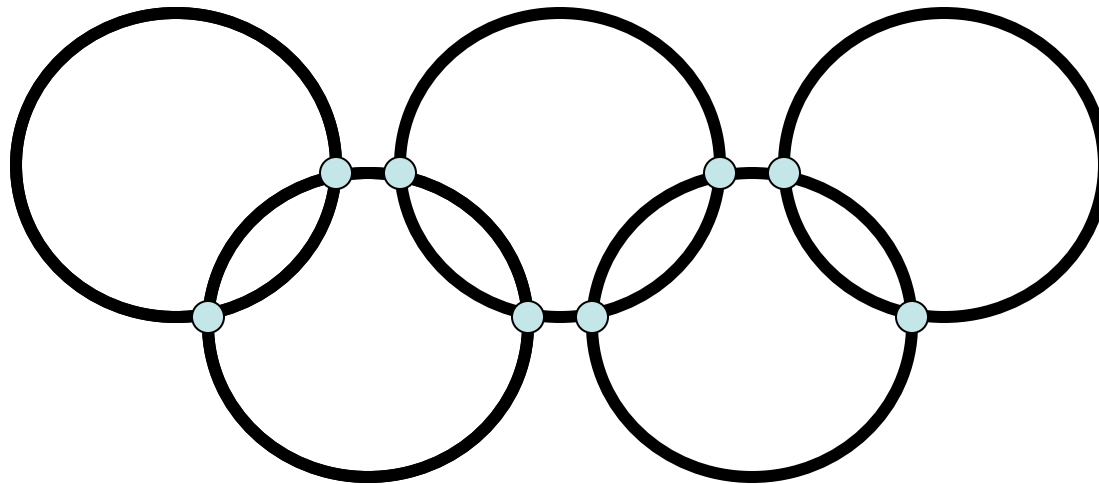
**Euler's theorem:** A connected graph has an Eulerian path if and only if it has zero or two vertices with odd degrees.

Proof by induction.

# Eulerian Cycle

**Euler's theorem:** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.

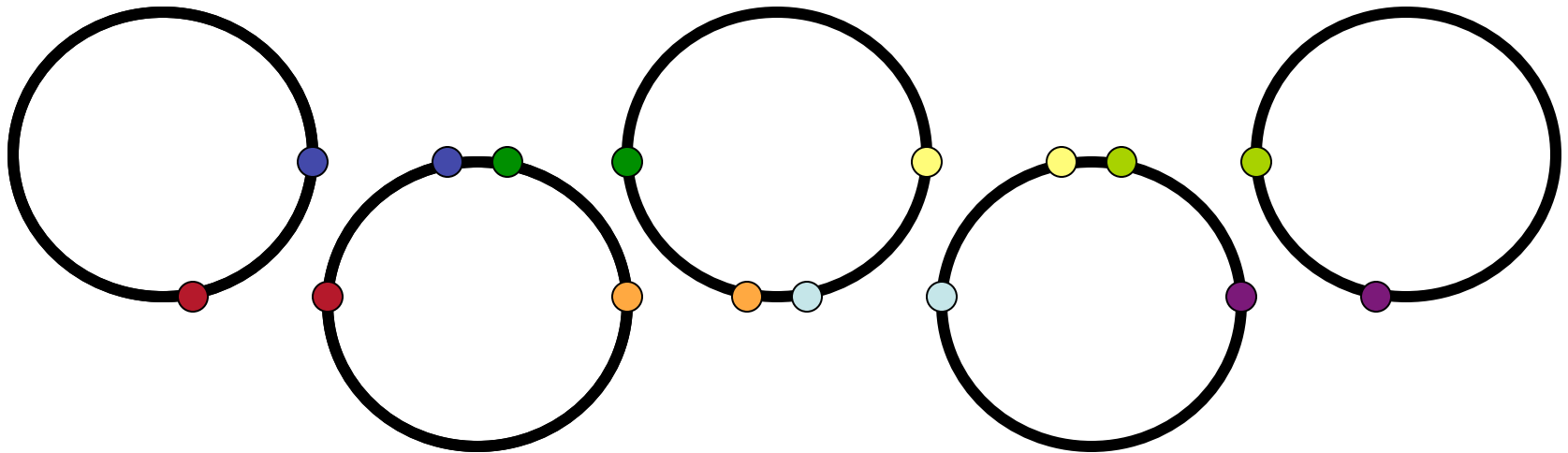
First we find an Eulerian cycle in the below example.



# Eulerian Cycle

**Euler's theorem:** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.

Note that the edges can be partitioned into five simple cycles.

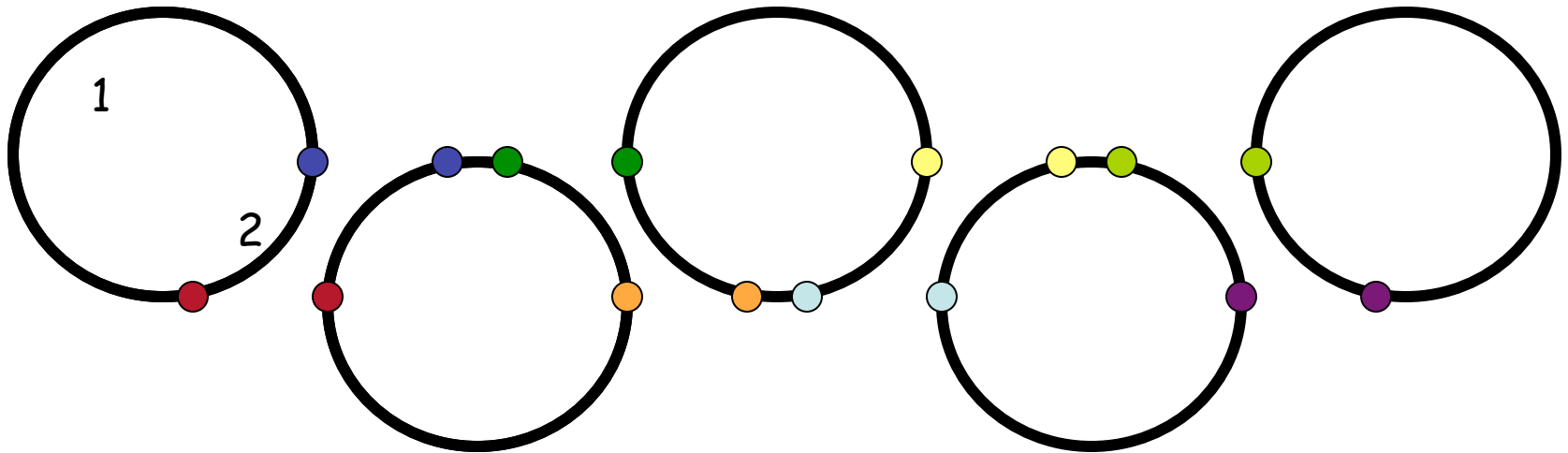


Vertices of the same color represent the same vertices.

# Eulerian Cycle

**Euler's theorem:** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.

The idea is that we can construct an Eulerian cycle by adding cycle one by one.

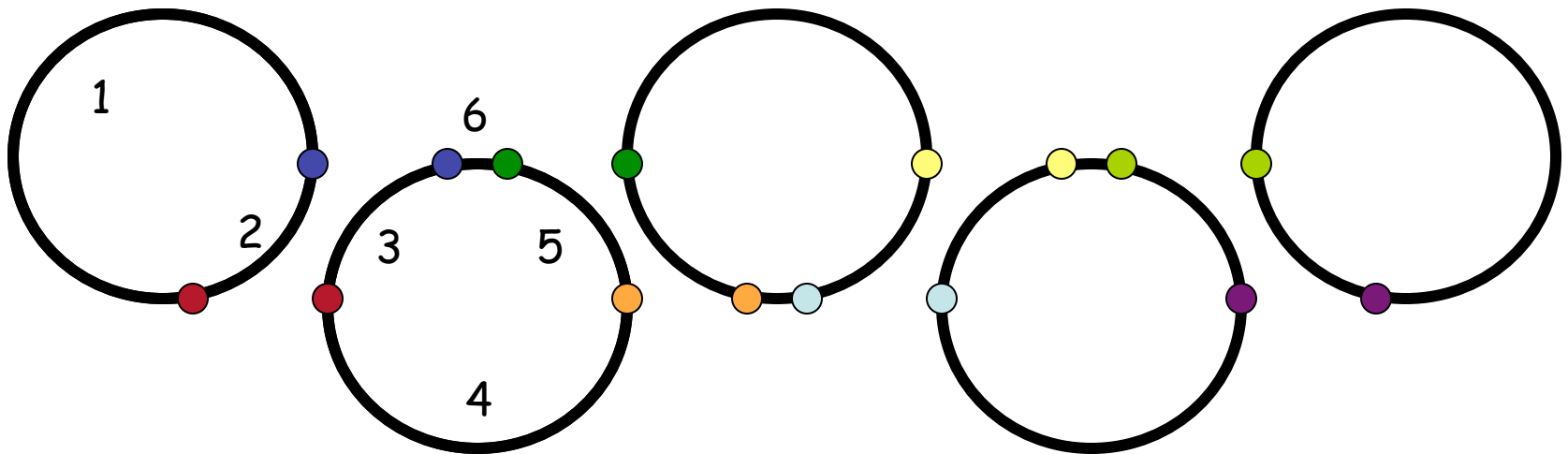


First traverse the first cycle.

# Eulerian Cycle

**Euler's theorem:** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.

The idea is that we can construct an Eulerian cycle by adding cycle one by one.



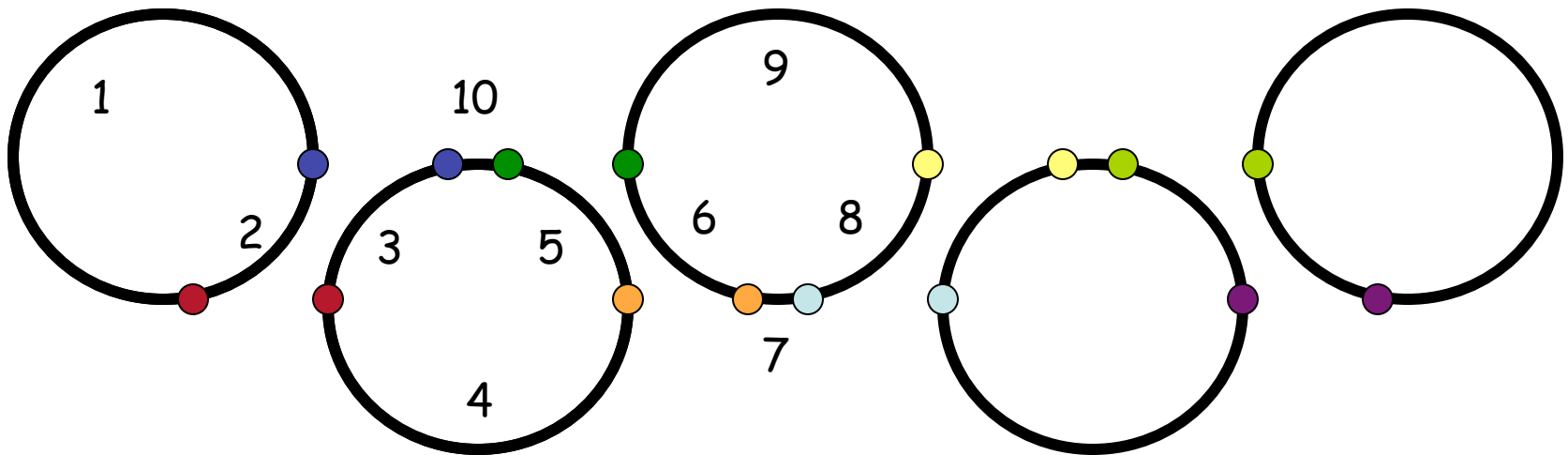
Then traverse the second cycle.



# Eulerian Cycle

**Euler's theorem:** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.

How to deal with the third cycle?

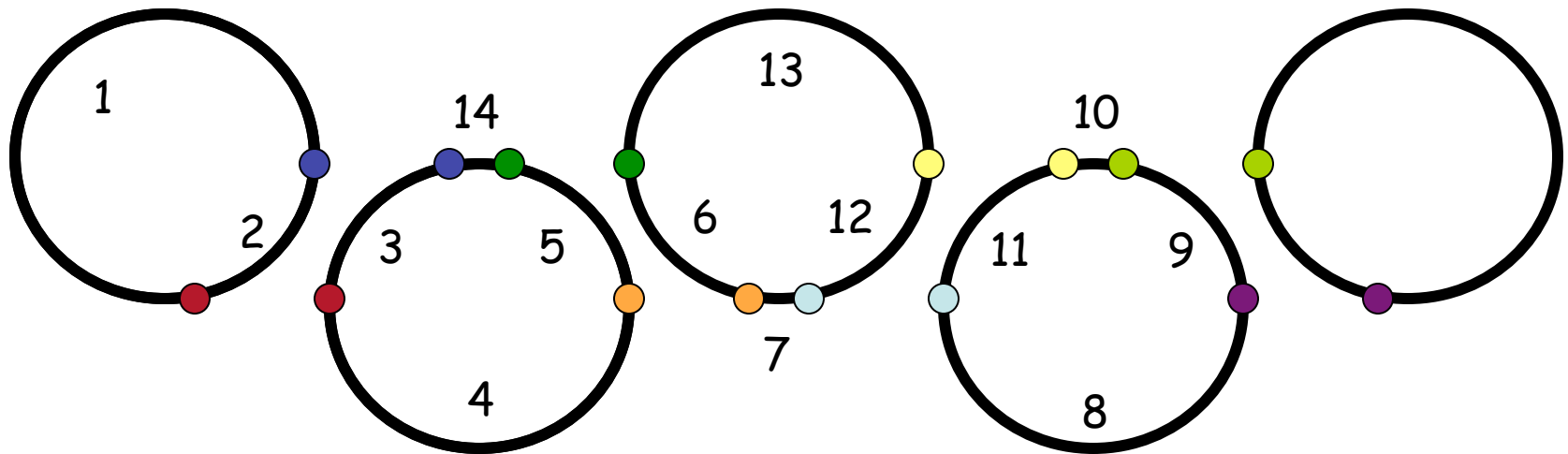


We can "detour" to the third cycle before finishing the second cycle.

# Eulerian Cycle

**Euler's theorem:** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.

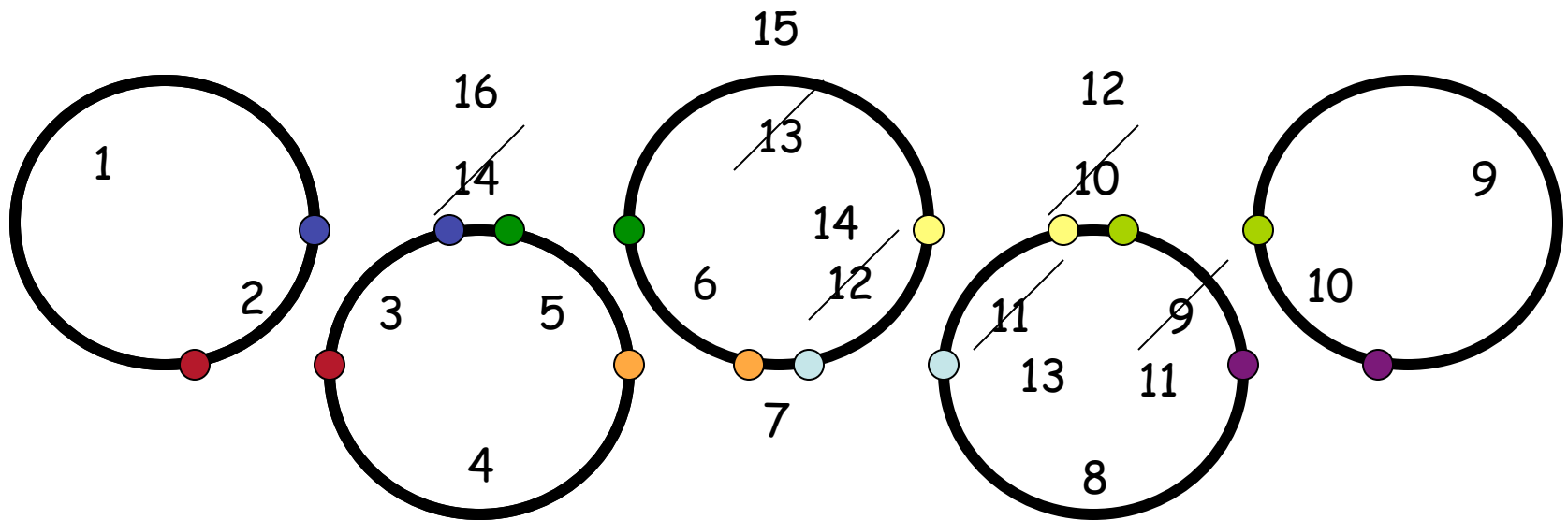
We use the same idea to deal with the fourth cycle



We can "detour" to the fourth cycle at an "intersection point".

# Eulerian Cycle

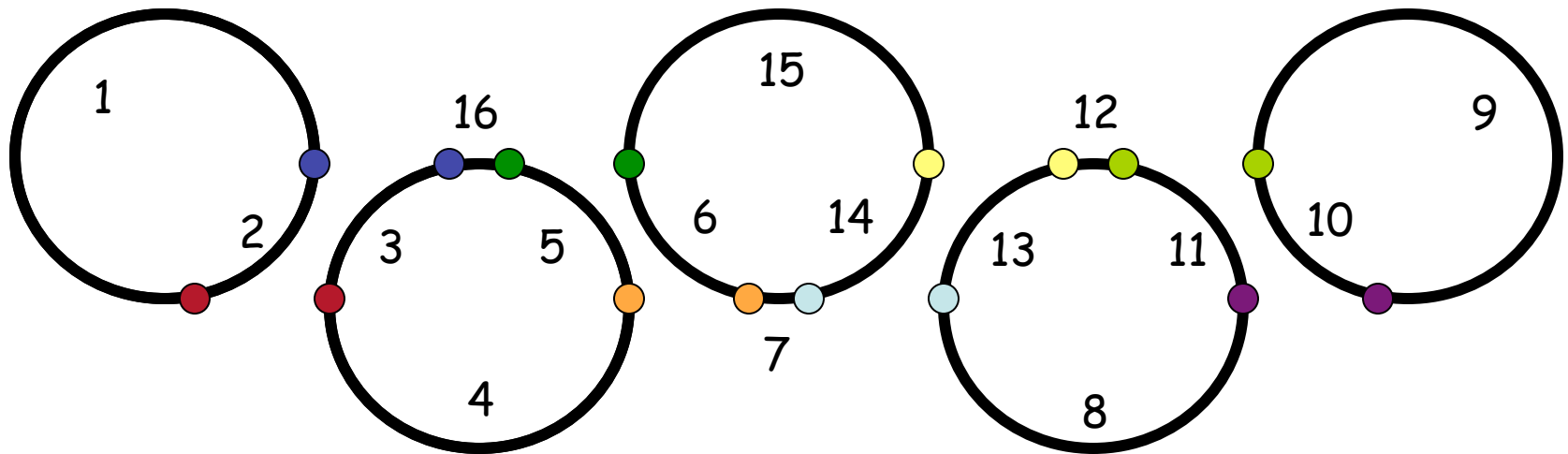
**Euler's theorem:** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.



We can "insert" the fifth cycle at an "intersection point".

# Eulerian Cycle

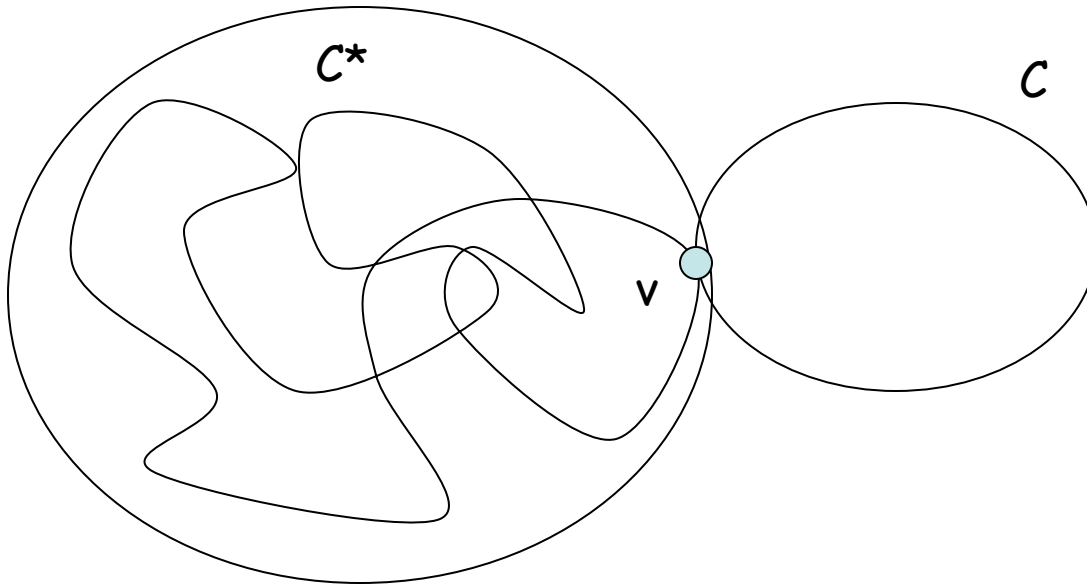
**Euler's theorem:** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.



So we have an Eulerian cycle of this example

## Idea

In general, if we have a “partial” Eulerian cycle  $C^*$ ,  
and it intersects with a cycle  $C$  on a vertex  $v$ ,  
then we can extend the Eulerian cycle  $C^*$  to include  $C$ .



First follow  $C^*$  until we visit  $v$ , then follow  $C$  until we go back to  $v$ ,  
and then follow  $C^*$  from  $v$  to the end.

# Proof

We have informally proved the following claim in the previous slides.

**Claim 1.** If the edges of a connected graph can be partitioned into simple cycles, then we can construct an Eulerian cycle.

**Euler's theorem:** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.

We can prove Euler's theorem if we can prove the following claim.

**Claim 2.** If every vertex is of even degree, then the edges can be partitioned into simple cycles.

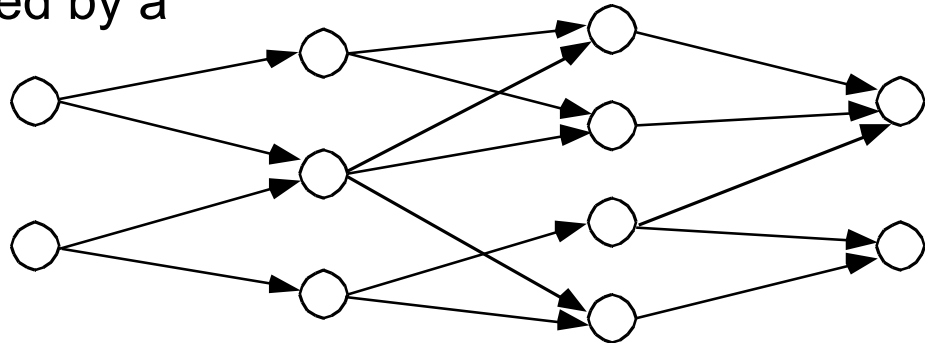
Proof is not considered here.

# This Lecture

- Seven bridges of Königsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- **Graphs and networks**
- Graph coloring

# Graphs and Networks

• Many problems can be represented by a graphical network representation.



• Examples:

- Distribution problems
- Routing problems
- Maximum flow problems
- Designing computer / phone / road networks
- Equipment replacement
- And of course the Internet

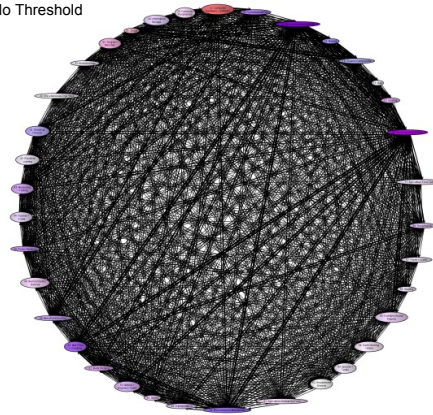
Aside: finding the right problem representation is one of the key issues.



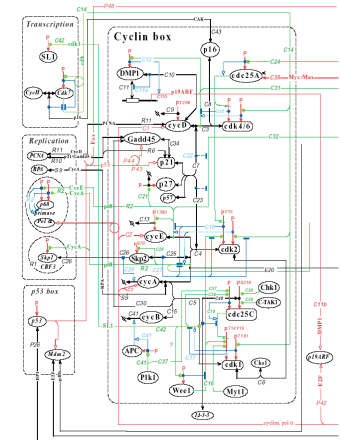
# New Science of Networks

Networks are pervasive

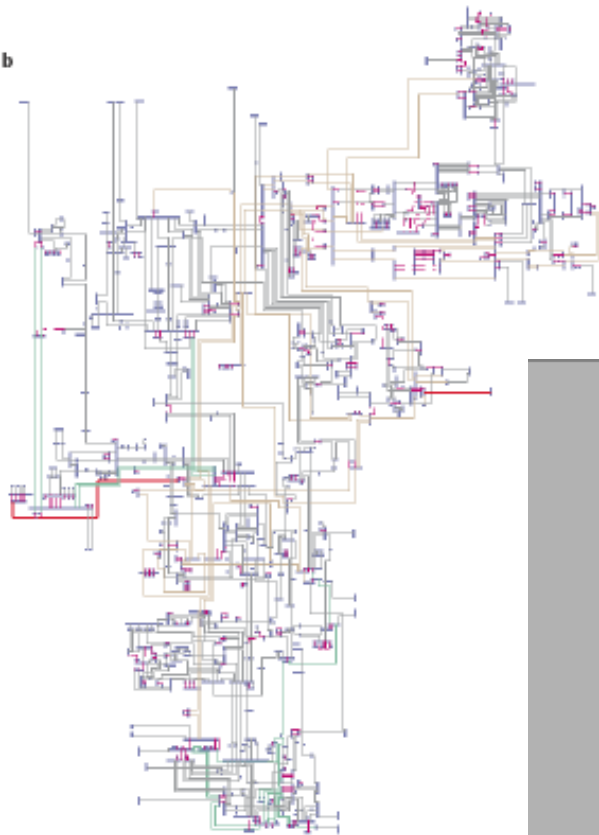
Sub-Category Graph  
No Threshold



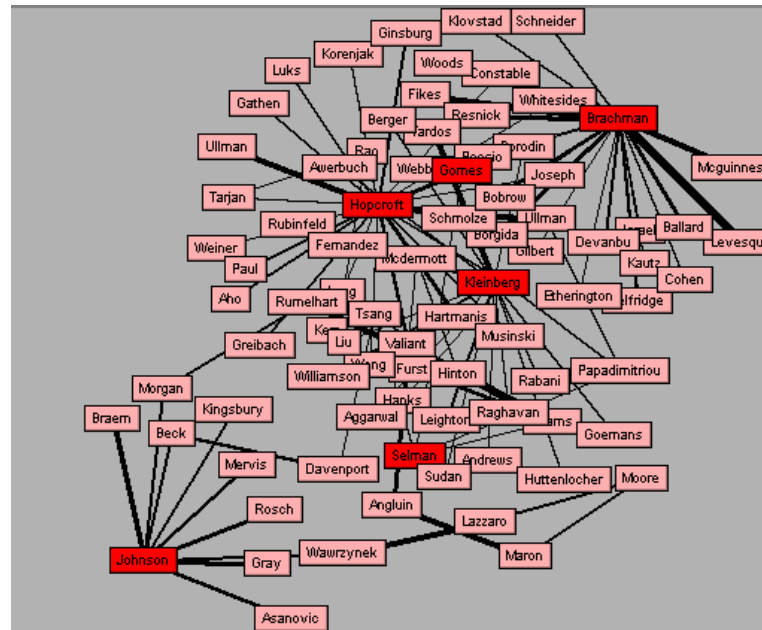
Utility Patent network  
1972-1999  
(3 Million patents)  
Gomes, Hopcroft, Lesser, Selman



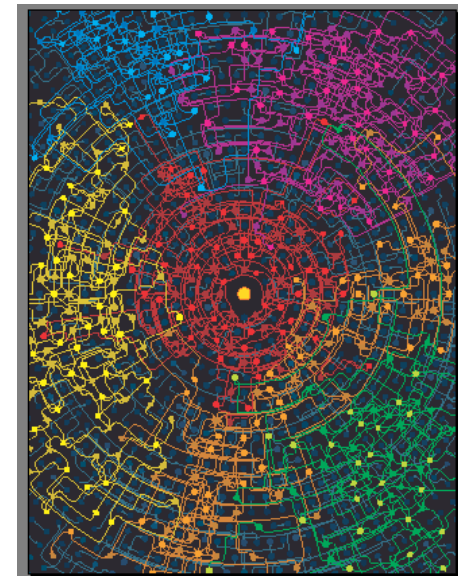
Neural network of the  
nematode worm *C. elegans*  
(Strogatz, Watts)



NYS Electric  
Power Grid  
(Thorp, Strogatz, Watts)



Network of computer scientists  
ReferralWeb System  
(Kautz and Selman)



Cybercommunities57  
(Automatically discovered)  
Kleinberg et al

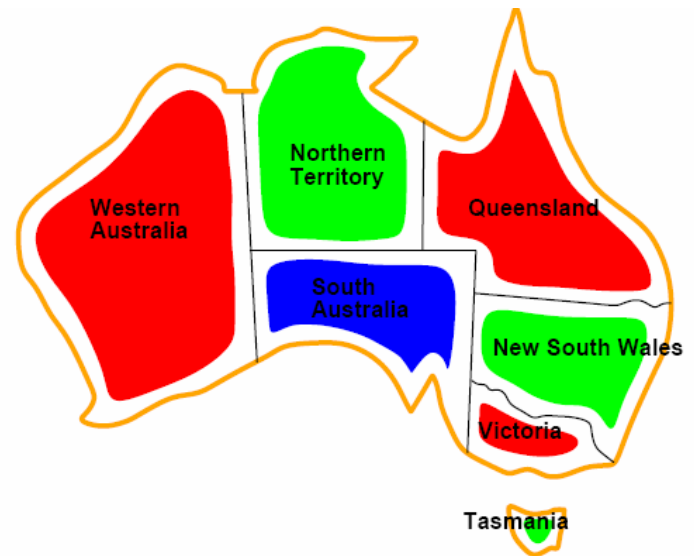
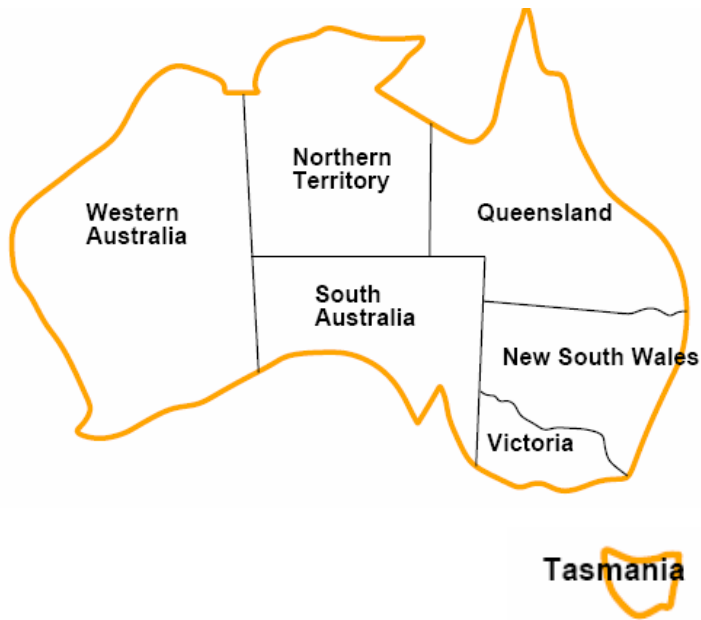
# Applications of Networks

| <b>Applications</b>                     | <b>Physical analog<br/>of nodes</b>                                      | <b>Physical analog<br/>of arcs</b>                     | <b>Flow</b>                                       |
|---|--|--|---|
| <b>Communication<br/>systems</b>        | phone exchanges,<br>computers,<br>transmission<br>facilities, satellites | Cables, fiber optic<br>links, microwave<br>relay links | Voice messages,<br>Data,<br>Video transmissions   |
| <b>Hydraulic systems</b>                | Pumping stations<br>Reservoirs, Lakes                                    | Pipelines  | Water, Gas, Oil,<br>Hydraulic fluids              |
| <b>Integrated<br/>computer circuits</b> | Gates, registers,<br>processors  | Wires  | Electrical current                                |
| <b>Mechanical systems</b>               | Joints   | Rods, Beams,<br>Springs                                | Heat, Energy                                      |
| <b>Transportation<br/>systems</b>       | Intersections,<br>Airports,<br>Rail yards                                | Highways,<br>Airline routes<br>Railbeds                | Passengers,<br>freight,<br>vehicles,<br>operators |

# This Lecture

- Seven bridges of Königsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Graphs and networks
- **Graph coloring**

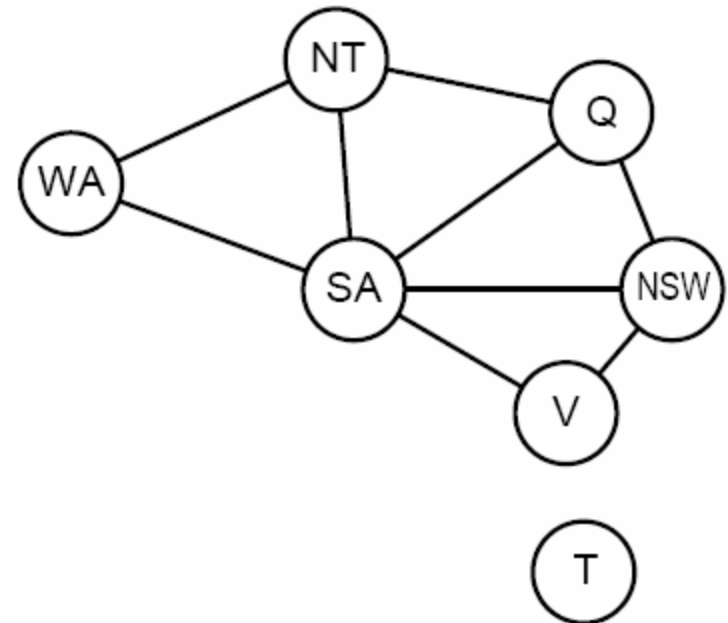
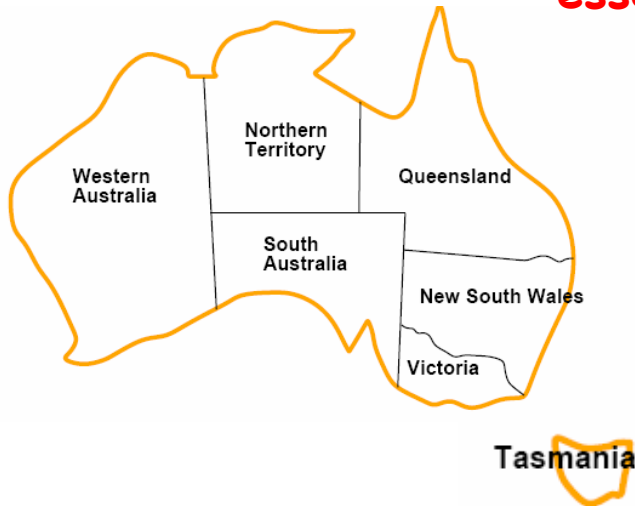
# Example: Coloring a Map



How to color this map so that no two adjacent regions have the same color?

# Graph representation

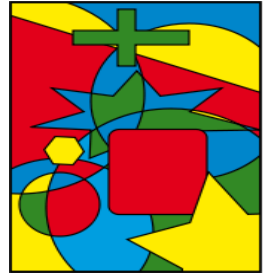
Abstract the  
essential info:



Coloring the nodes of the graph:

What's the minimum number of colors such that any two nodes connected by an edge have different colors?

# Four Color Theorem



Four color map.

- The **chromatic number** of a graph is the **least number of colors** that are required to color a graph.
- **The Four Color Theorem** – *the chromatic number of a planar graph is no greater than four. (quite surprising!)*
- Proof by Appel and Haken 1976;
- careful case analysis performed by computer;
- Proof reduced the **infinitude of possible maps to 1,936 reducible configurations** (later reduced to 1,476) which had to be checked one by one by computer.
- The computer program ran for hundreds of hours. The first significant *computer-assisted* mathematical proof. *Write-up was hundreds of pages including code!*