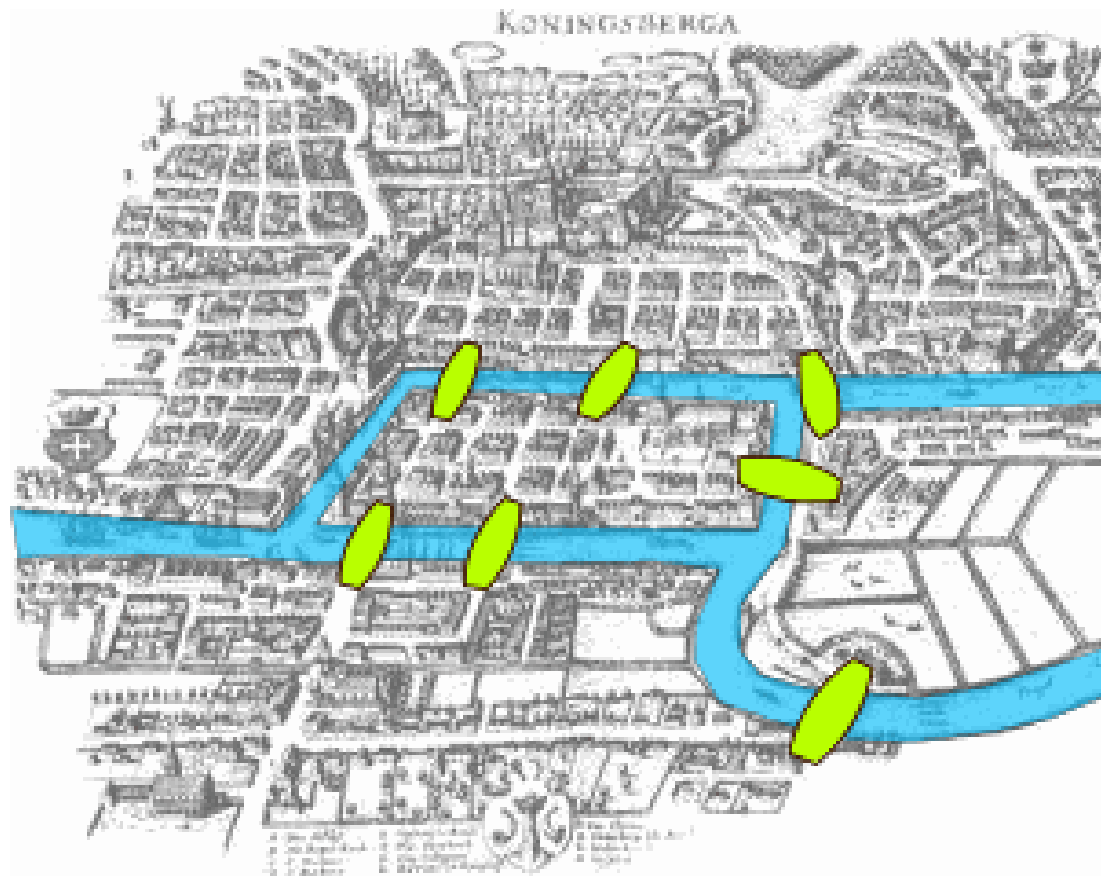


# Introduction to Graphs



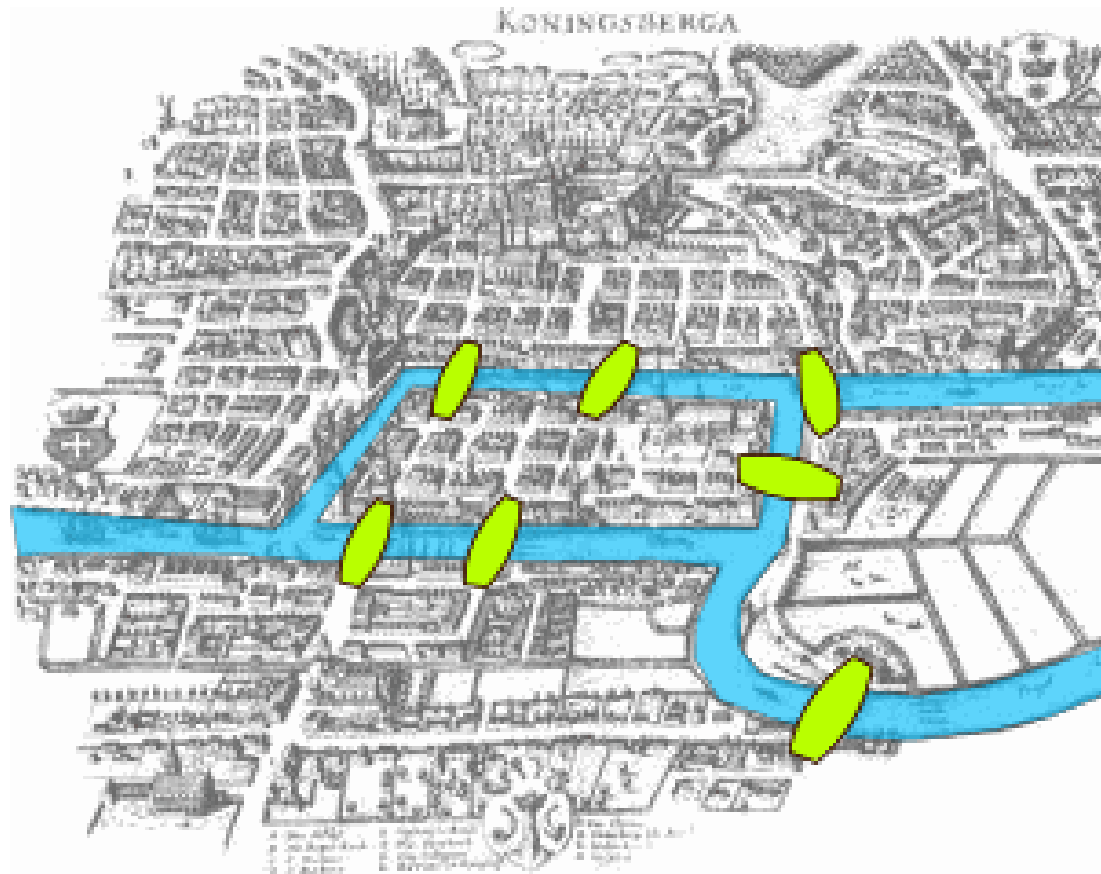
# This Lecture

In this part we will study some basic graph theory.

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

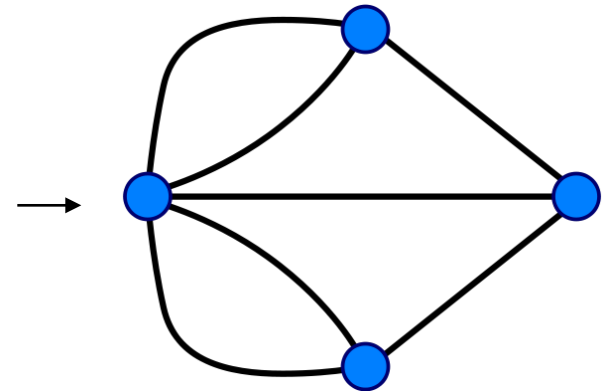
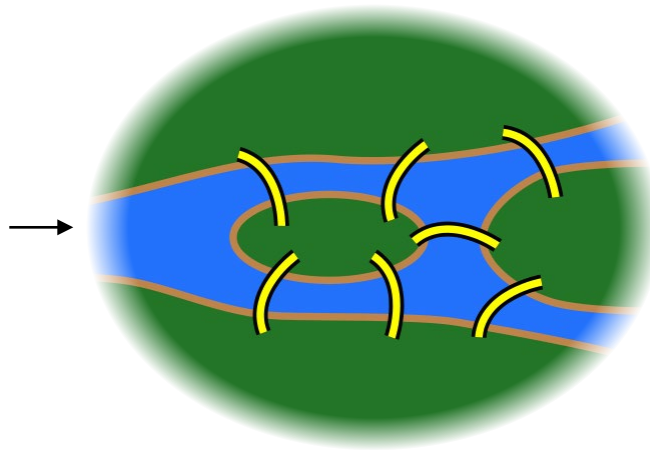
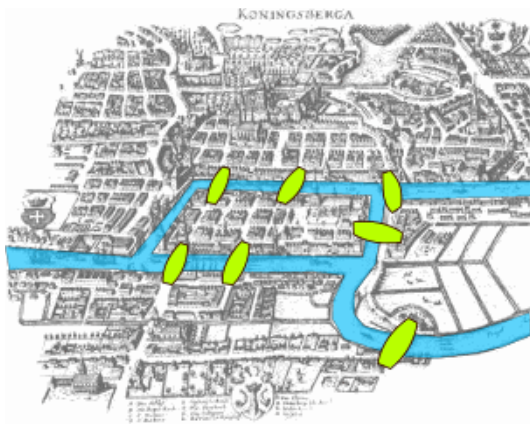
- Seven bridges of Königsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Directed graphs

# Seven Bridges of Königsberg



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

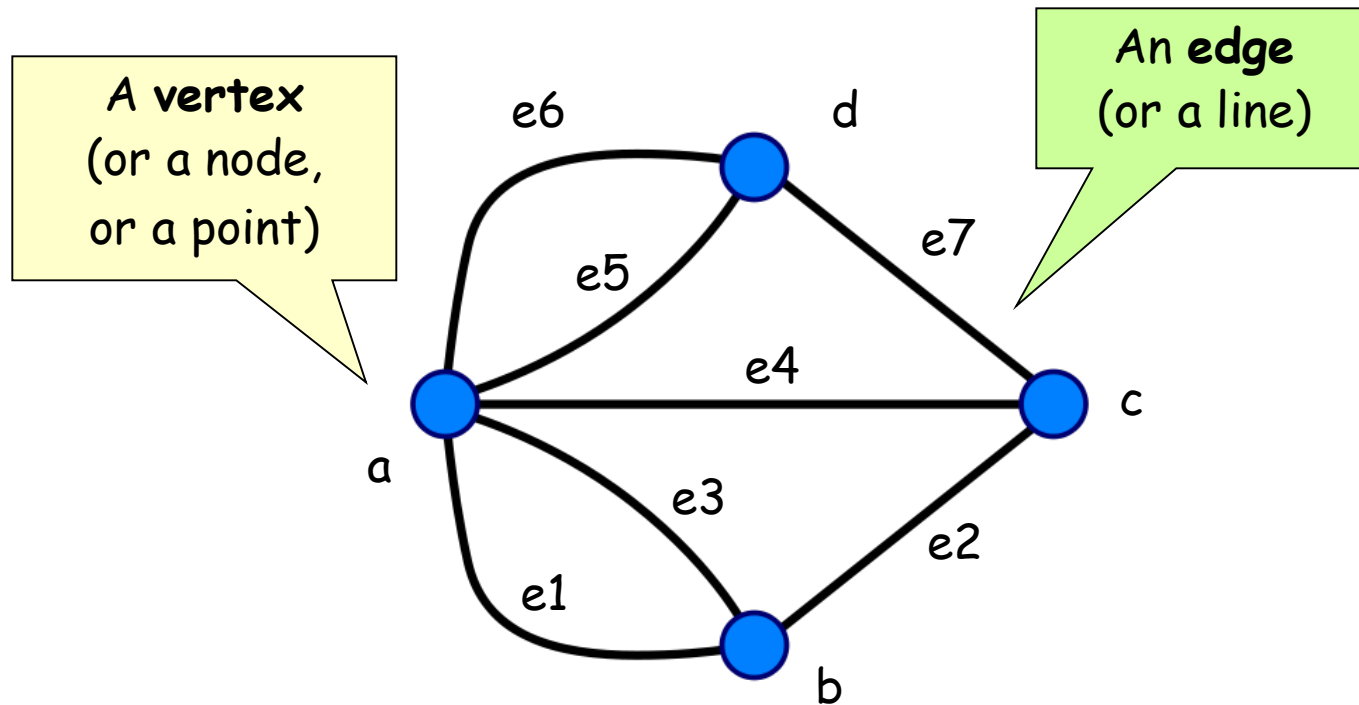
# Seven Bridges of Königsberg



Forget unimportant details.

Forget even more.

# A Graph

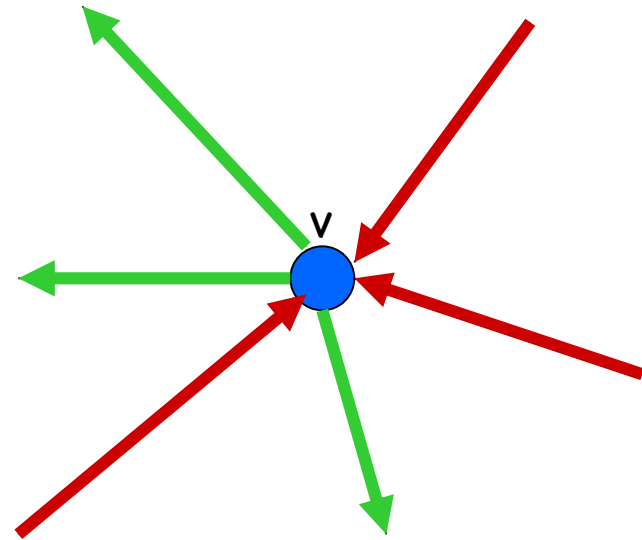
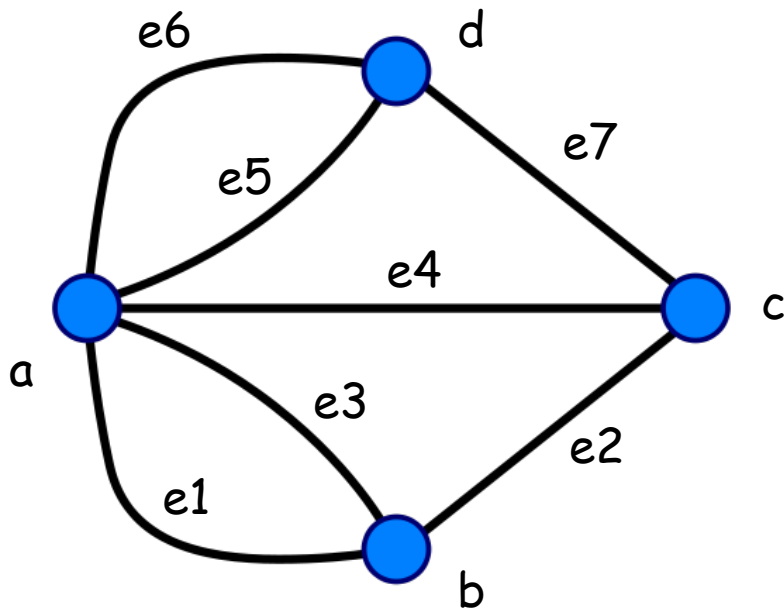


So, what is the “Seven Bridges of Königsberg” problem now?

To find a walk that visits each edge exactly once.

# Euler's (*pronounce oiler*) Solution

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

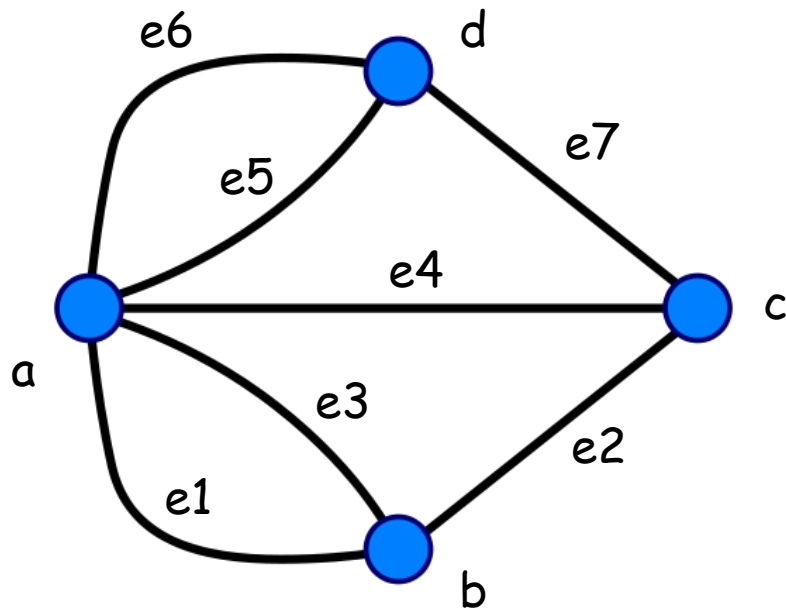


Suppose there is such a walk. Then there is a starting point and an end point.

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

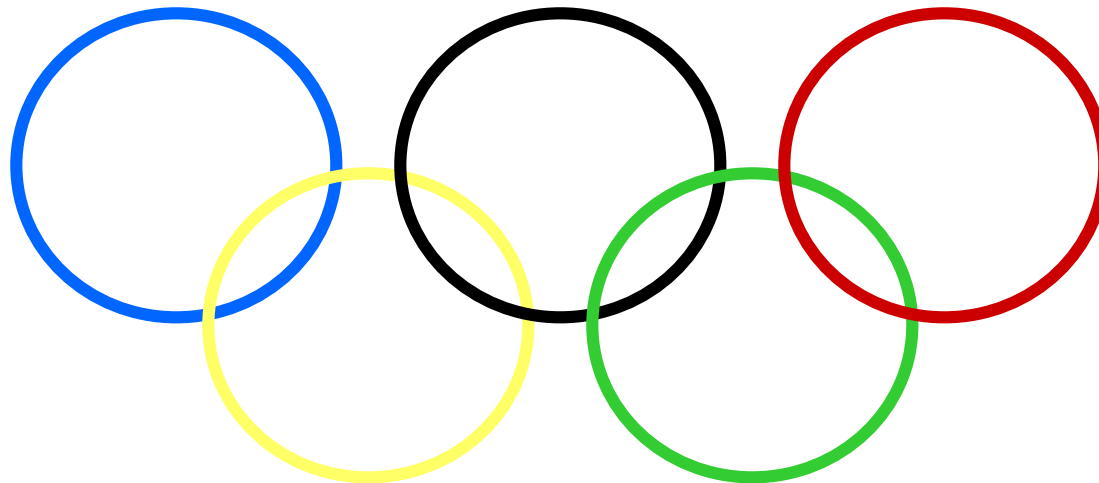
Suppose there is such a walk. Then there is a starting point and an end point.

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



One-line  
drawing  
problem

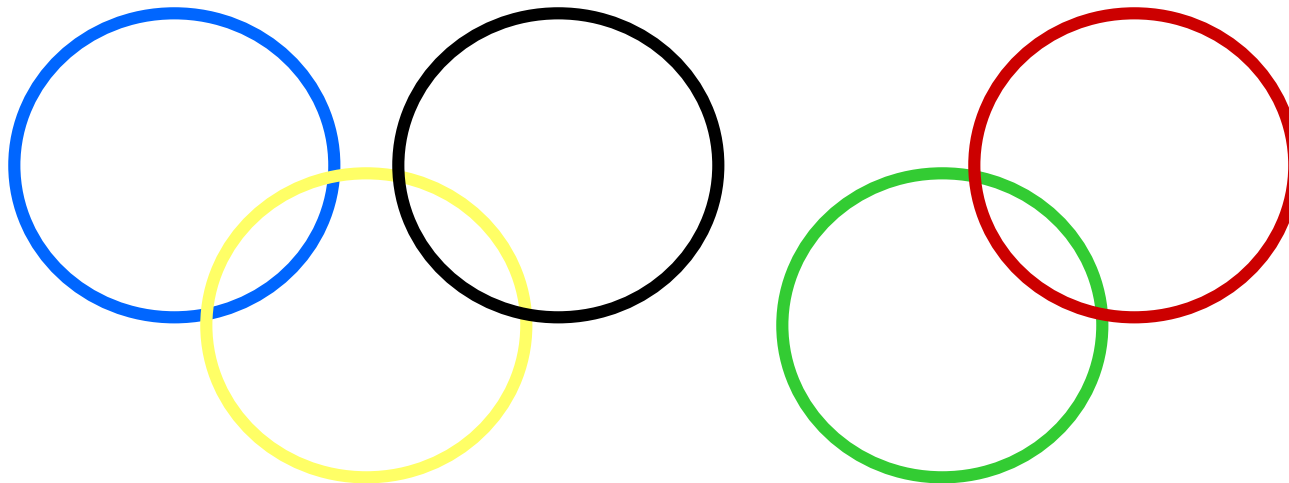
Is it always possible to find such a walk if there are  
**at most two** vertices with odd number of edges?



# 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?



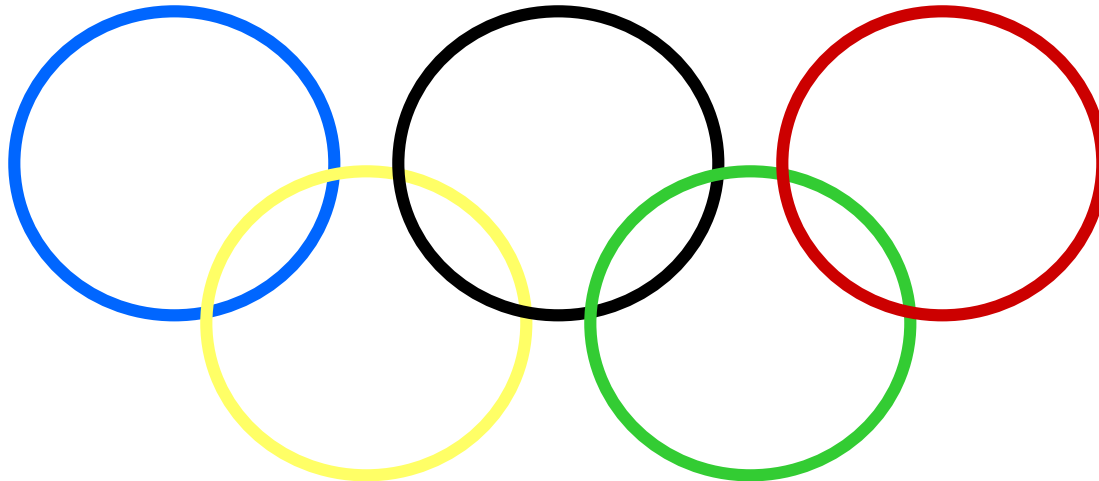
Is it always possible to find such a walk if there are  
**at most two** vertices with odd number of edges?

**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.

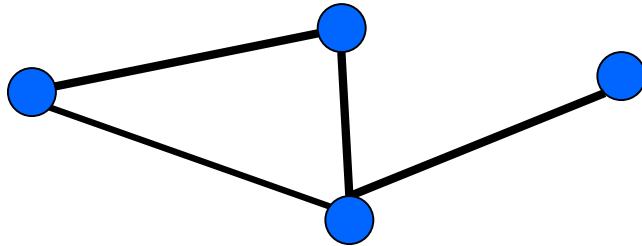
This theorem was proved in 1736,  
and was regarded as the starting point of graph theory.

# This Lecture

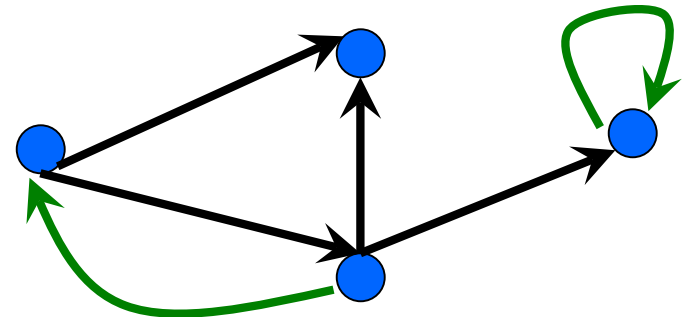
- Seven bridges of Königsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Directed graphs

# Types of Graphs

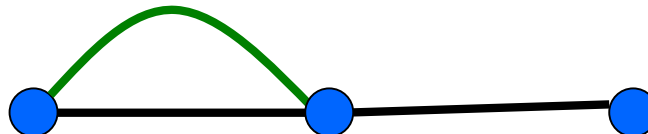
**Simple Graph**  
(no multiedges, no loops,  
no directed edges)



**Directed Graph (digraph)**



**Multi-Graph**



Eulerian path  
problem

Unless otherwise specified, all graphs in this course are simple.

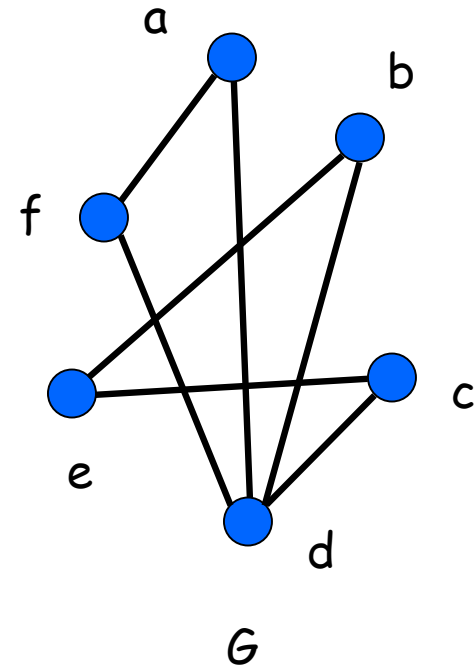
# Simple Graphs

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

A set of vertices,  $V$

A set of *undirected* edges,  $E$

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



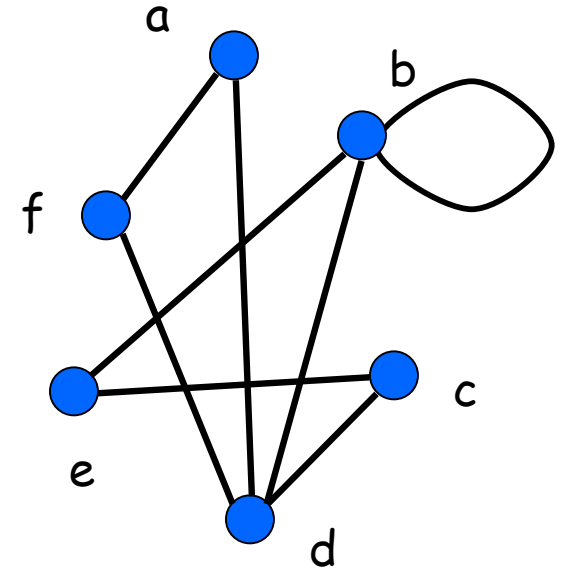
The vertices a,b are **adjacent** (**neighbours**) if the edge ab is present.

# Vertex Degrees

An edge  $uv$  is **incident** with 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 (but loops counted twice.)

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

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

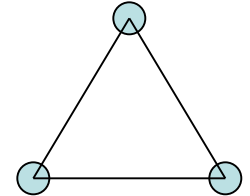
For multigraphs, **NO**.

For simple graphs, **YES**.

# Degree Sequence

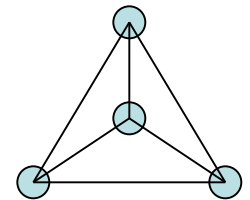
Is there a graph with degree sequence  $(2,2,2)$ ?

YES.



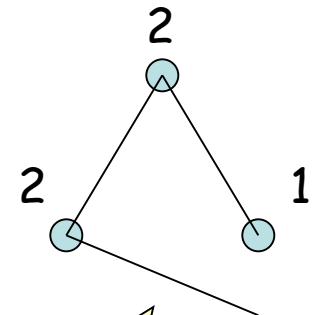
Is there a graph with degree sequence  $(3,3,3,3)$ ?

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 always a graph with such a degree sequence?

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

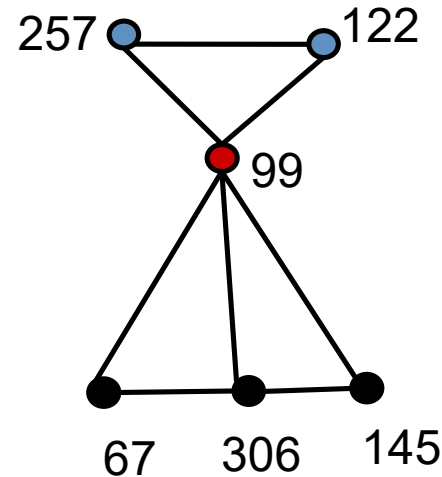
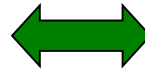
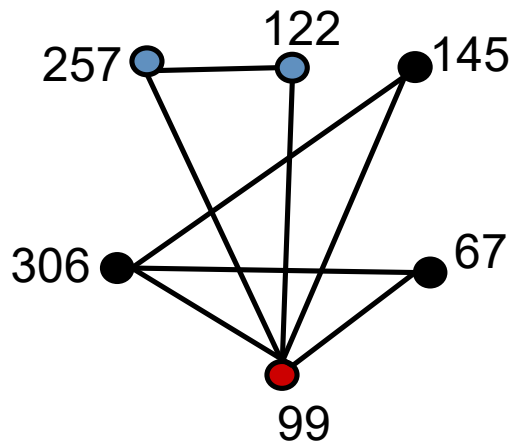
For multigraphs (with self loops), **YES!**

# This Lecture

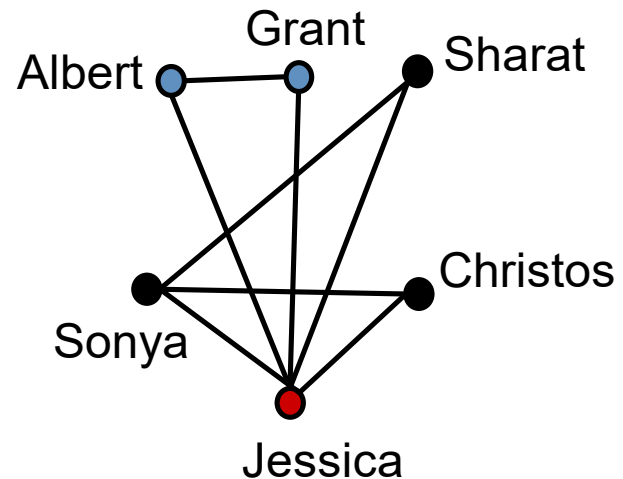
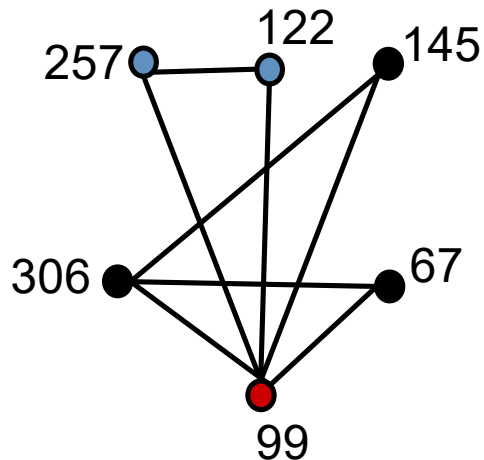
- Seven bridges of Königsberg
- Graphs, degrees
- **Isomorphism**
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Directed graphs

# Same Graphs?

Same graph (different drawings)



Same graph (different labels)



# Graph Isomorphism

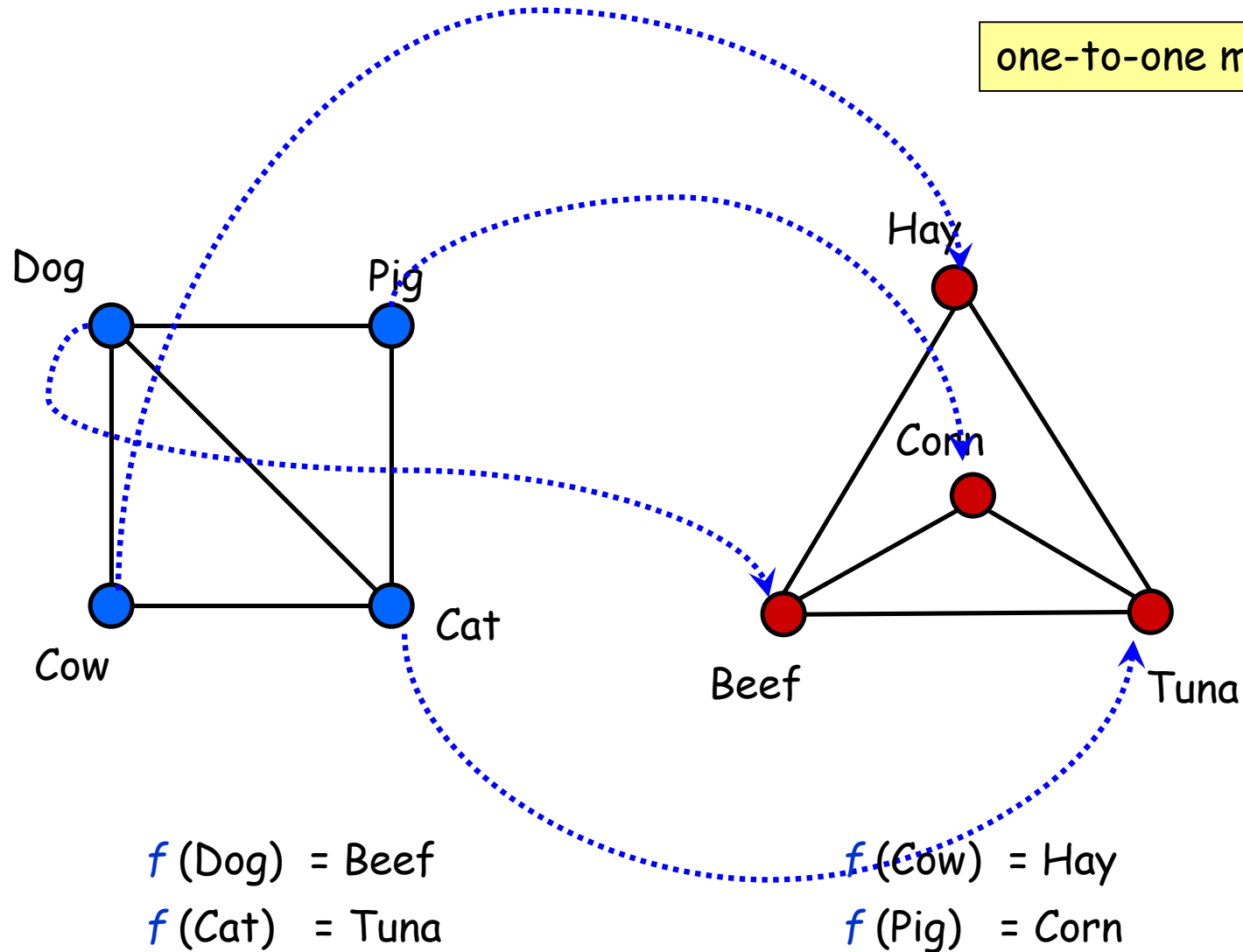
All that matters is the *connections*.

Graphs with the same connections are *isomorphic*.

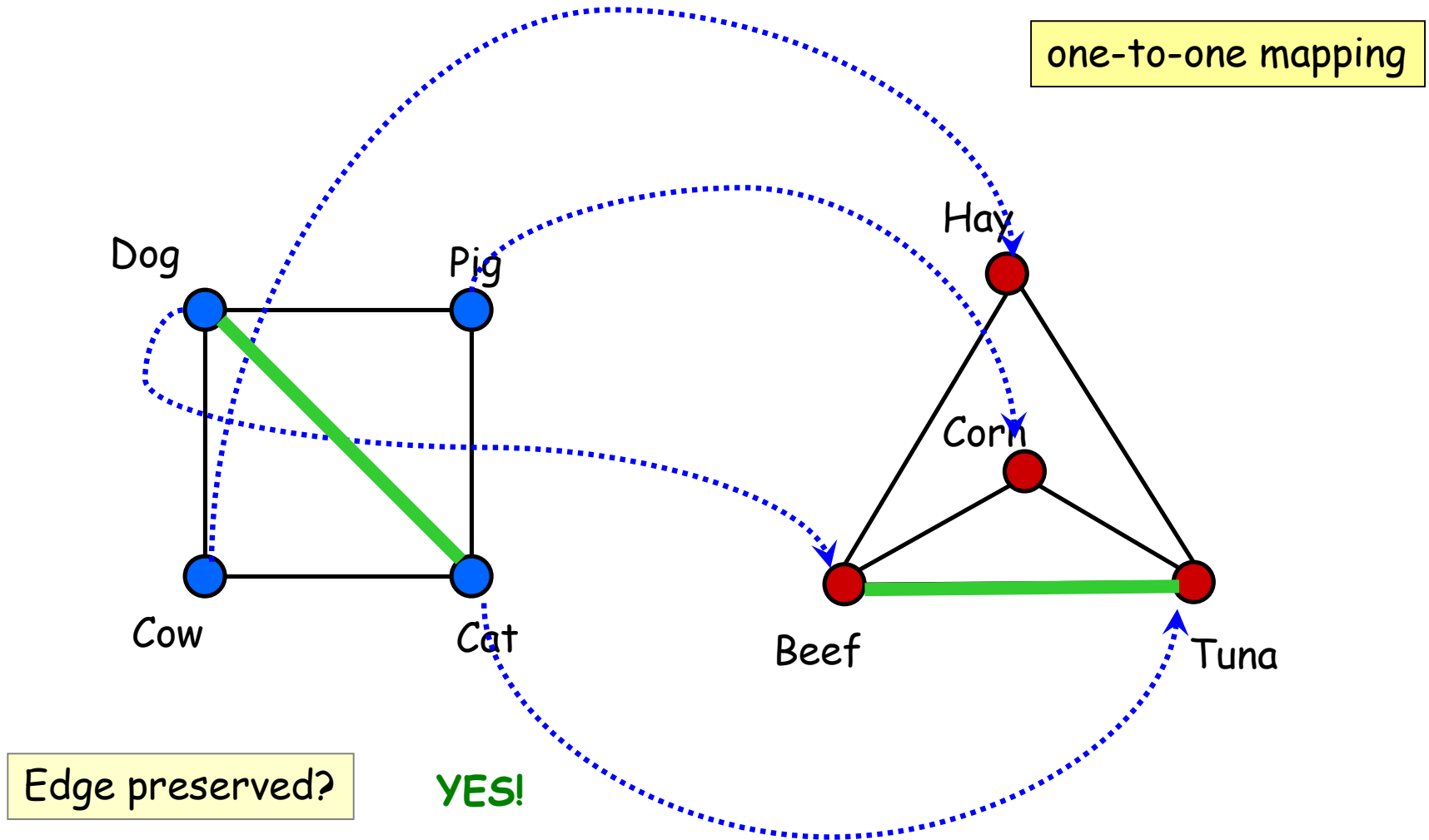
Informally, two graphs are isomorphic if they are the same after *renaming*.

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

# Are These Isomorphic?

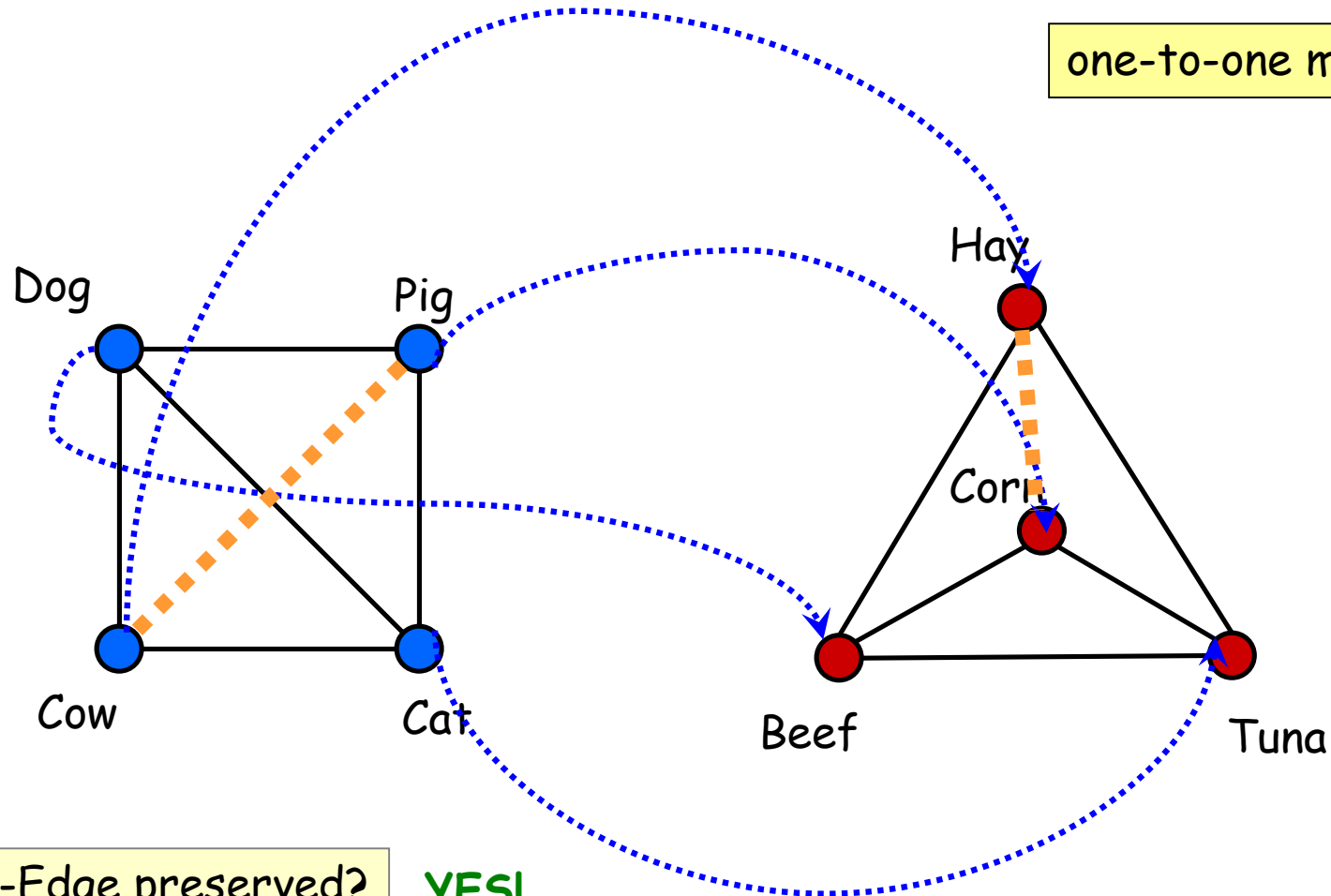


# Are These Isomorphic?



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 a one-to-one mapping (*isomorphism*) of the vertices that is edge-preserving.

$\exists$  one-to-one mapping  $f: V_1 \rightarrow V_2$   
 $u-v$  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

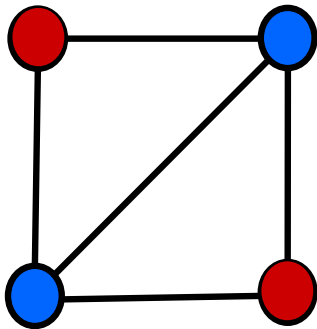
# Checking Graph Isomorphism

How to show two graphs are isomorphic?

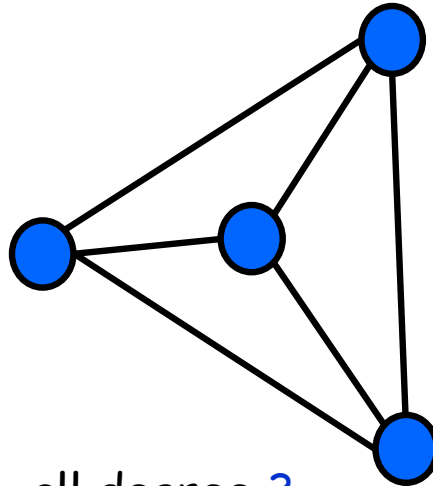
Find a mapping and show that it is edge preserving.

How to show two graphs are not isomorphic?

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



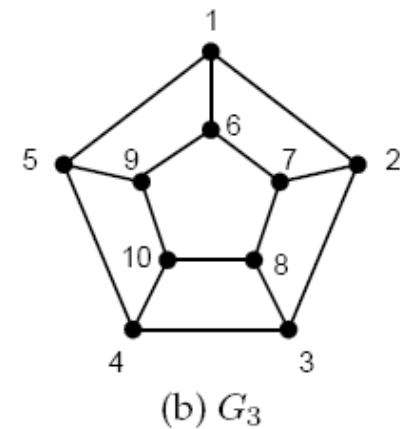
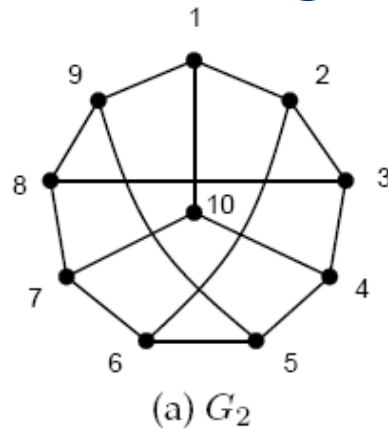
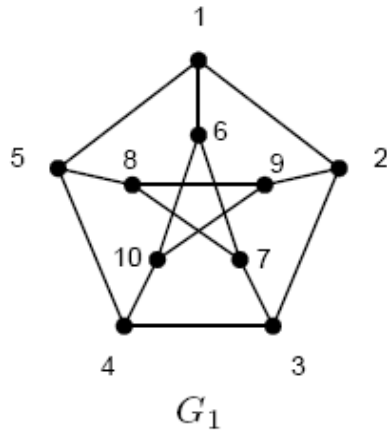
degree 2



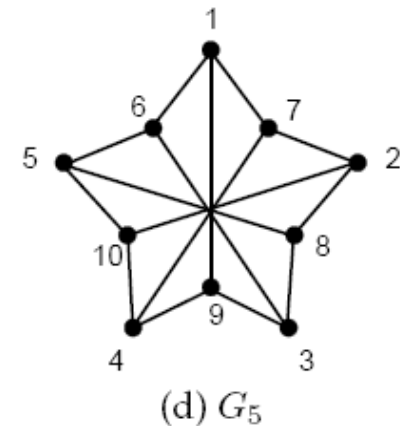
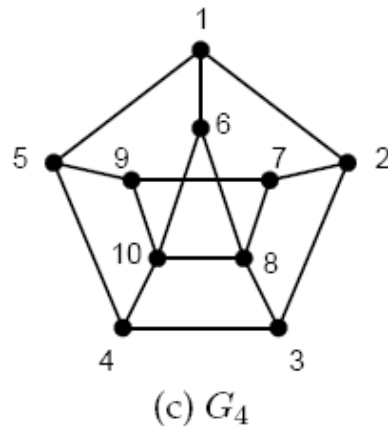
all degree 3

Non-isomorphic

## Challenge



Which is isomorphic to  $G_1$ ?



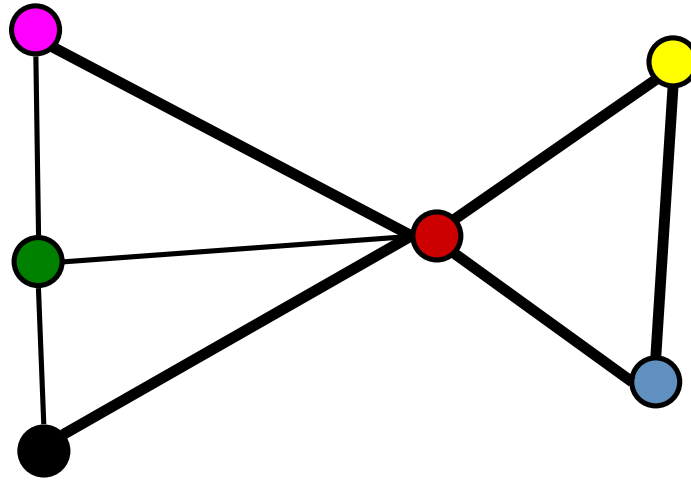
*Testing graph isomorphism is not easy -*

The problem is not known to be solvable in polynomial time nor to be NP-complete (known as NP-intermediate)

# This Lecture

- Seven bridges of Königsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Directed graphs

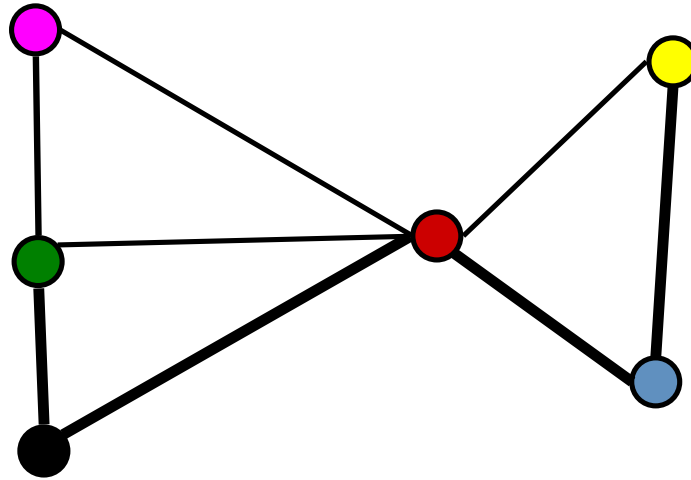
# Paths



*Path*: sequence of adjacent vertices

( ● ● ● ● ● ● )

# Simple Paths



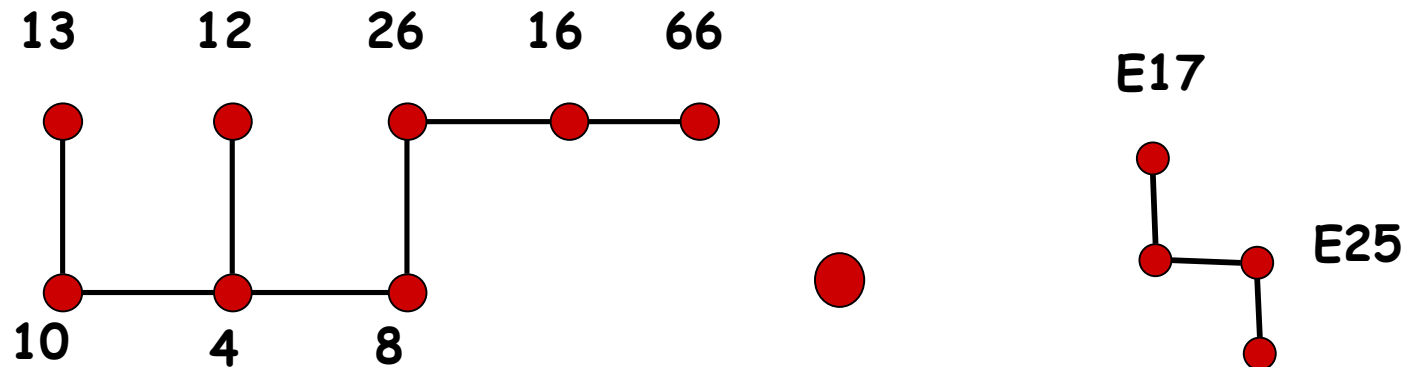
*Simple Path*: all vertices different

( ● ● ● ● ● )

# Connectedness

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

Every graph consists of disjoint connected pieces (called *connected components*)



3 connected components

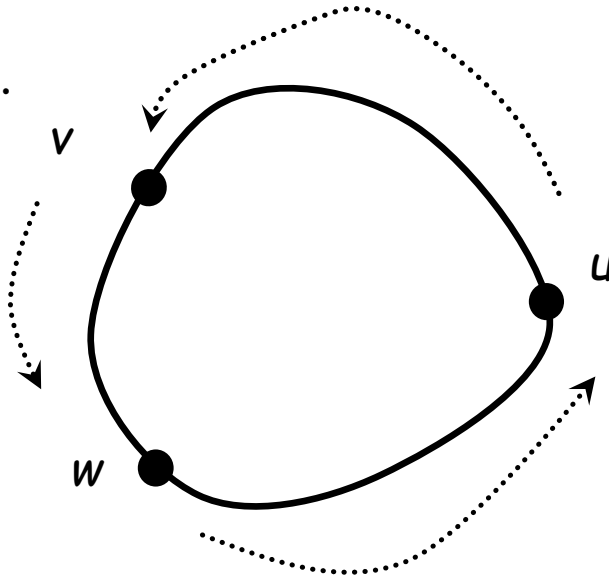
So a graph is *connected* if and only if it has only 1 connected component.

# Simple Cycles

A (simple) **cycle** is a connected graph whose vertices are all of degree 2.

A (simple) **cycle** in a graph is a connected **subgraph** whose vertices are all of degree 2.

For multigraphs, a loop is a cycle.



cycle:  $u \cdots v \cdots w \cdots u$     also:  $v \cdots w \cdots u \cdots v$

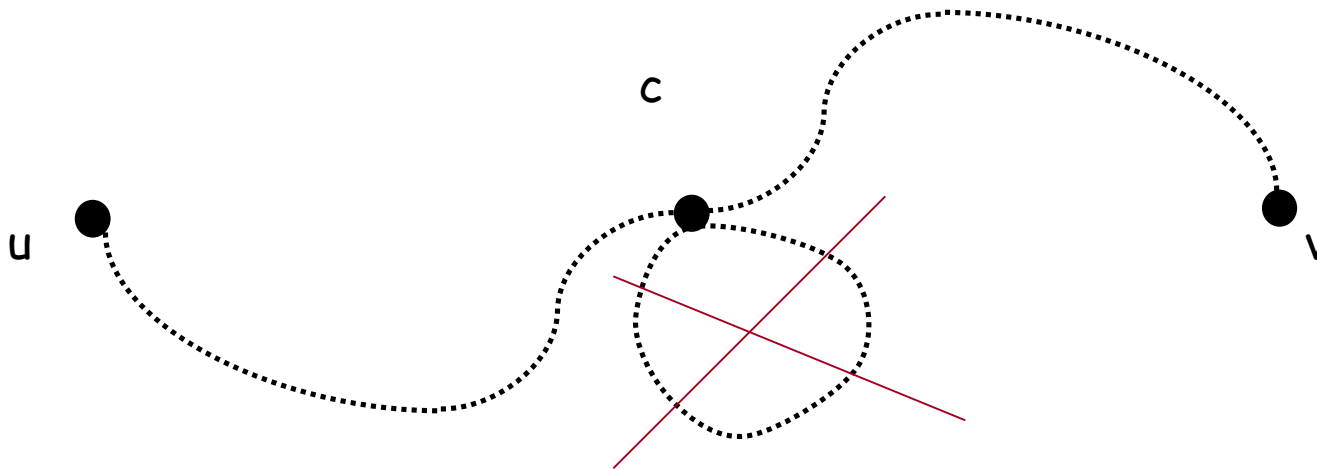


# Shortest Paths

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

Is a shortest path between two vertices always simple?

Idea: remove the cycle will make the path shorter.

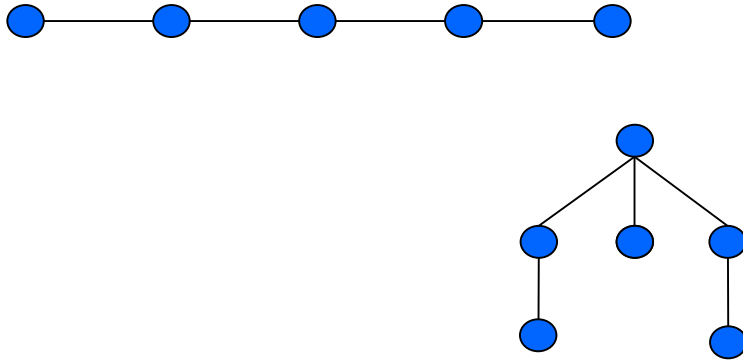


# This Lecture

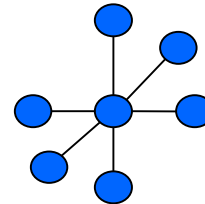
- Seven bridges of Königsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- **Tree**
- Eulerian cycle
- Directed graphs

# Tree

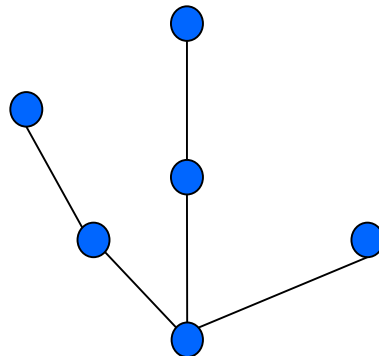
Graphs with no cycles?



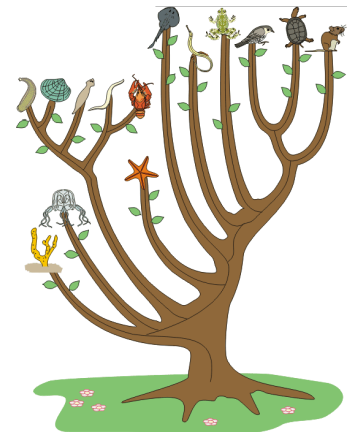
A forest.



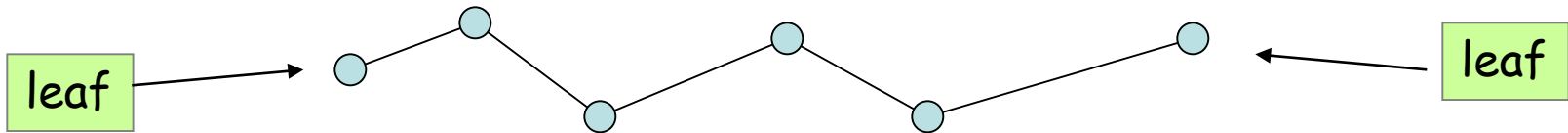
Connected graphs with no cycles?



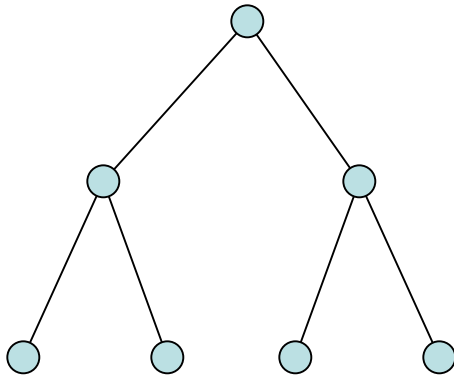
A tree.



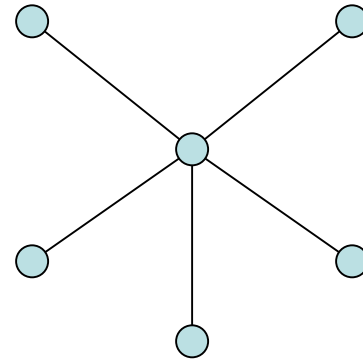
# More Trees



A *leaf* is a vertex of degree 1.



More leaves.



Even more leaves.

# Tree Characterization by Path

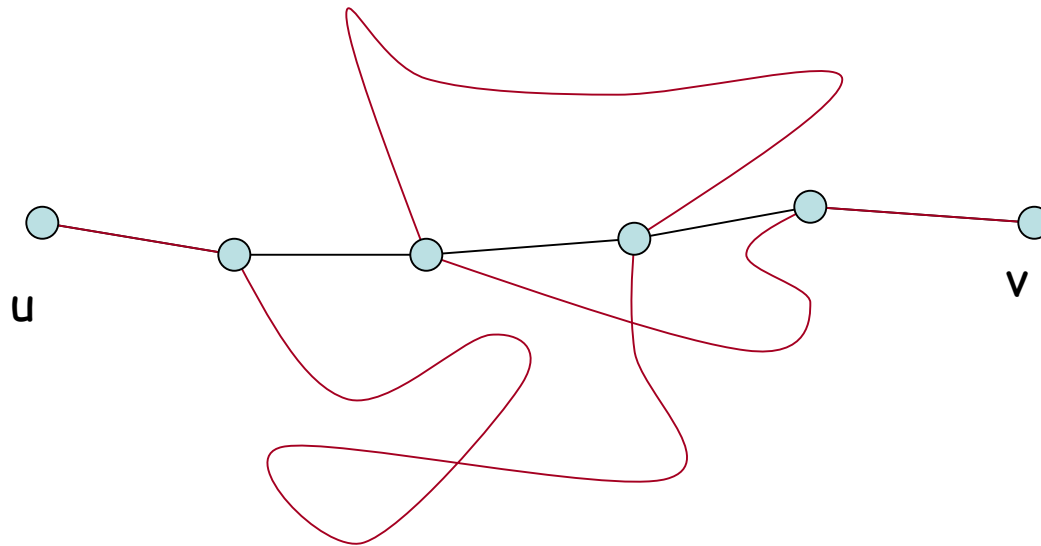
**Definition.** A tree is a connected graph with no cycles.

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.

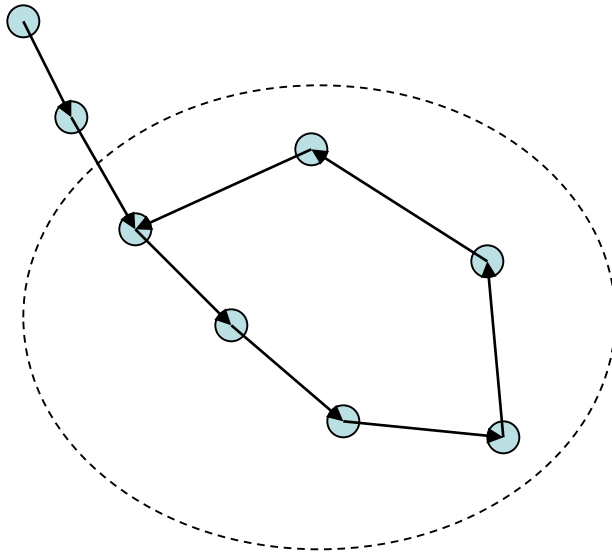
**Claim.** In a tree, there is a unique simple path between every pair of vertices.

# Tree Characterization by Number of Edges

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

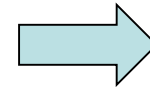
Can a tree have no leaves? **NO**

Then every vertex has degree at least 2.



Go to unvisited vertices as far as possible.

Finitely many  
vertices



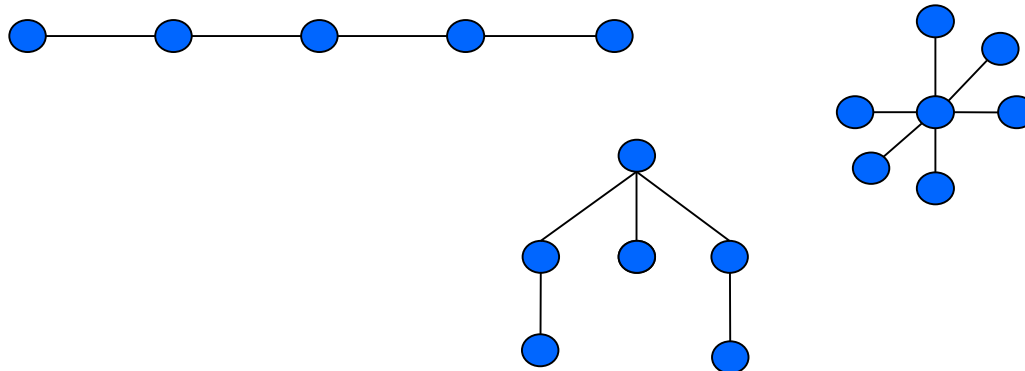
there is a cycle.

# 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?  $m = n - 1$



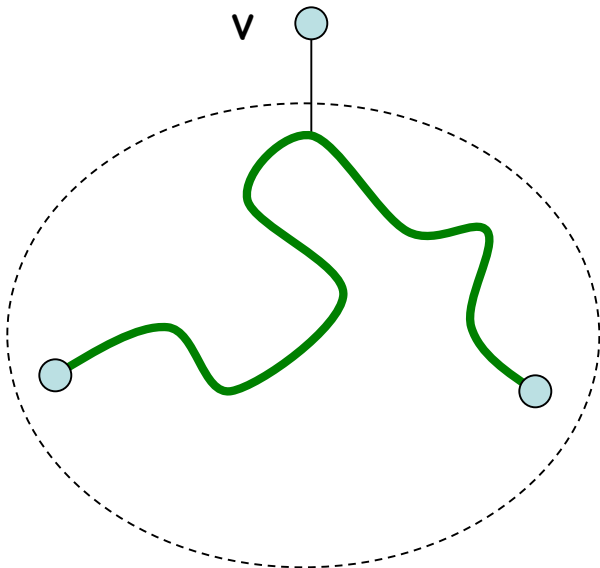
We usually use  $n$  to denote the number of vertices,  
and use  $m$  to denote the number of edges in a graph.

# 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?  $m = n-1$



Look at a leaf  $v$ .

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

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

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.  
(We have proved one direction, how about the converse?)

- Remove edges as long as there is a cycle.
- This will result in a tree with  $n-1$  edges.
- So the original graph has more than  $n-1$  edges, a contradiction.

# This Lecture

- Seven bridges of Königsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Directed graphs

# 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

Here we only discuss simple graphs, but the technique can be easily generalized to multigraphs.

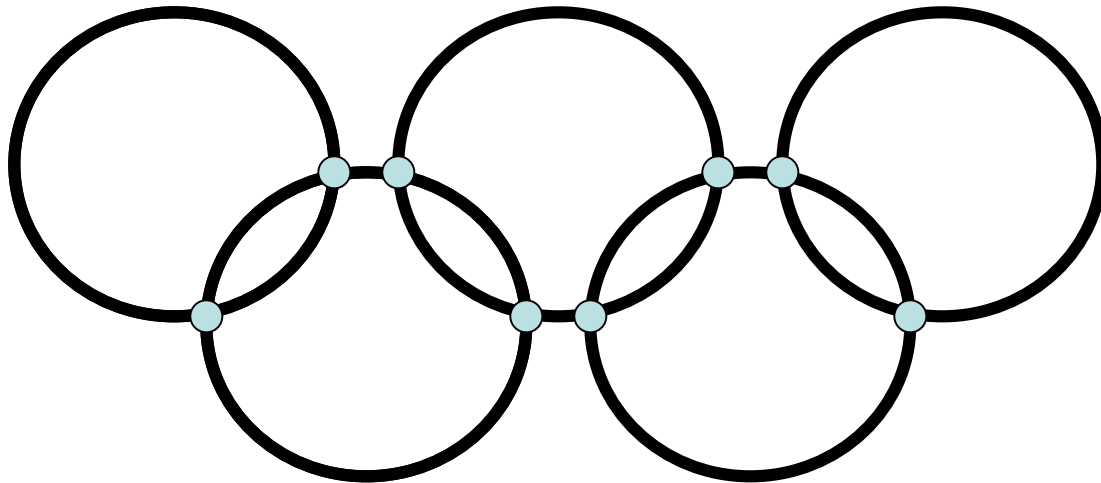
**Euler's theorem.** A connected graph has an Eulerian path if and only if it has zero or two vertices with odd degrees.

Proof by induction. We first consider the case of ***Eulerian cycle*** (a cycle that visits each edge exactly once).

# Eulerian Cycle

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

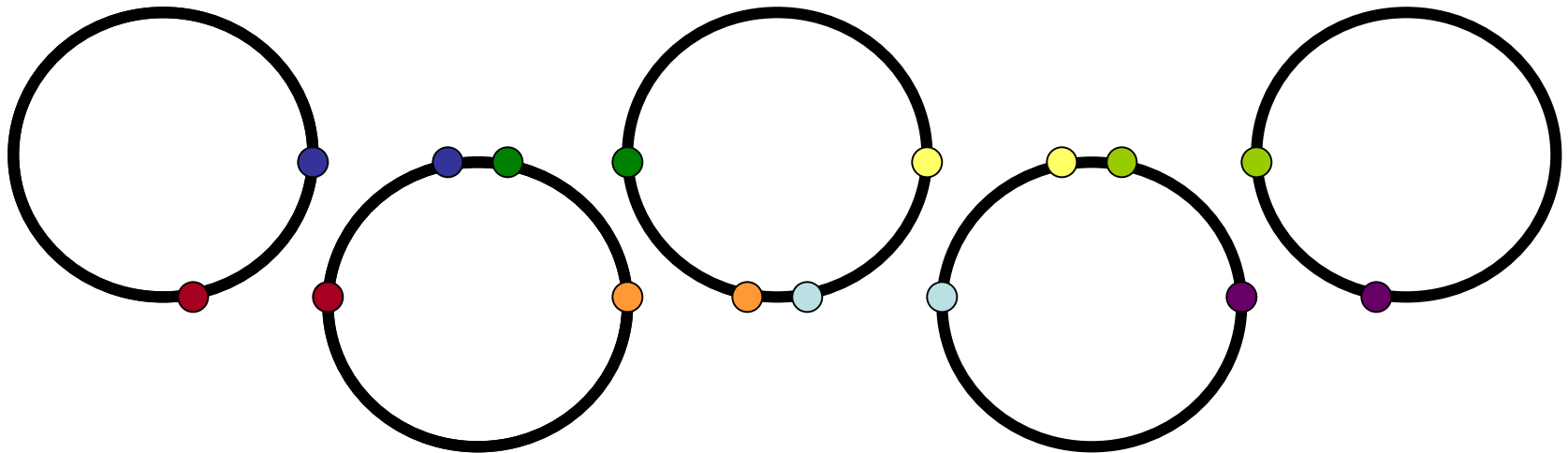
First we find an Eulerian cycle in the example below.



# 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.

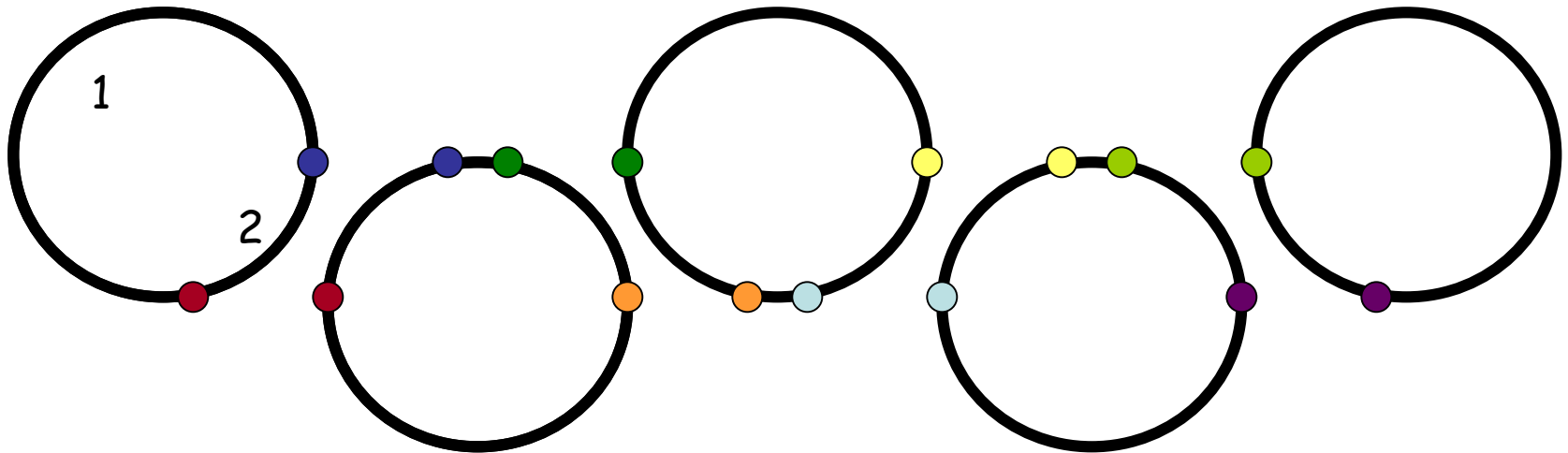


Vertices of the same color represent the same vertices.

# 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 cycles one by one.

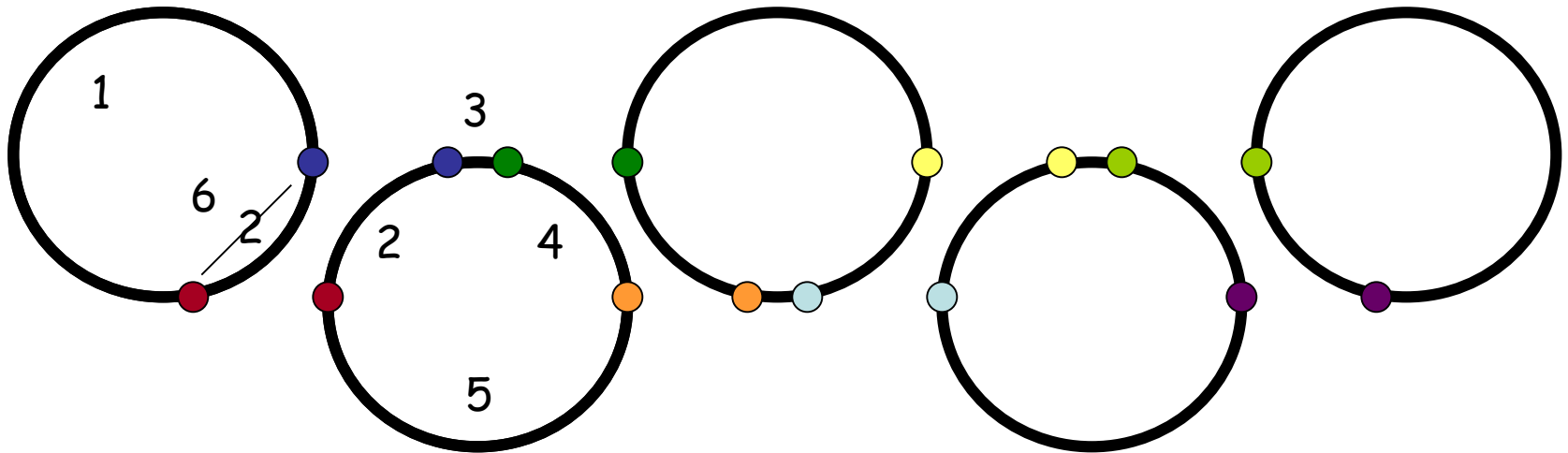


First transverse the first cycle.

# 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 cycles one by one.

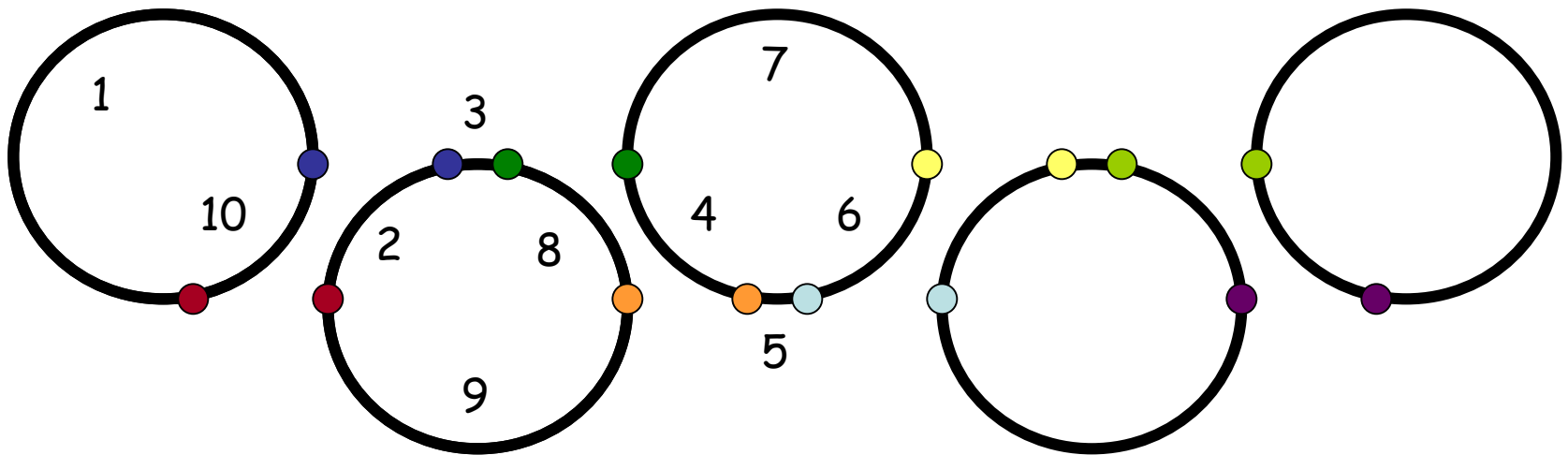


Then transverse the second cycle.

# 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?



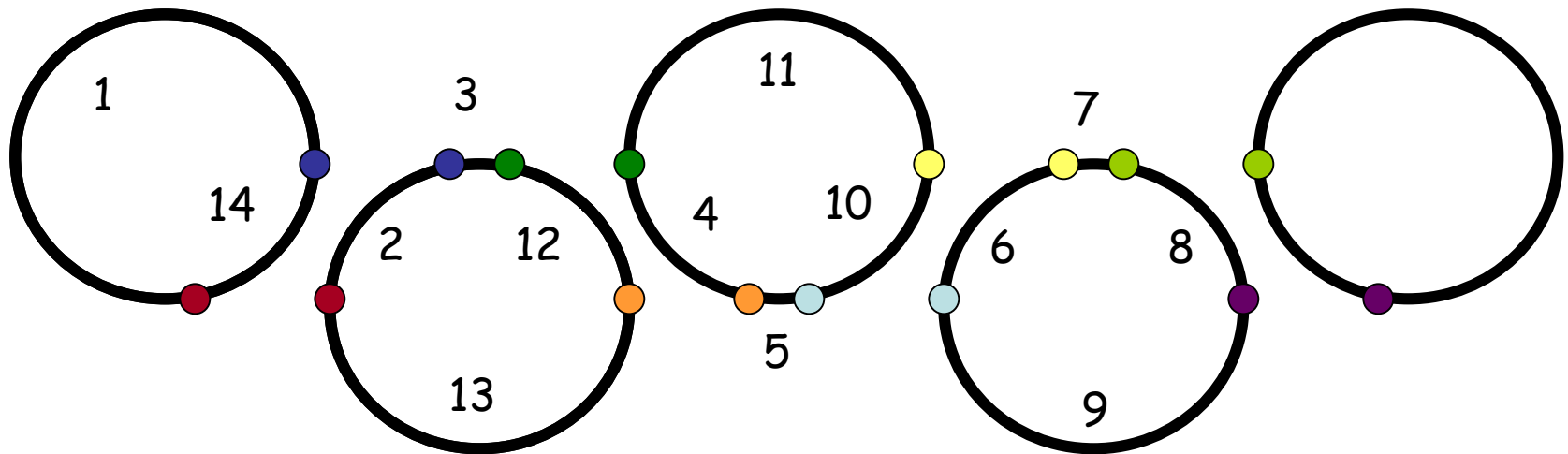
We can "detour" to the third cycle before finishing the second cycle.



# 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.

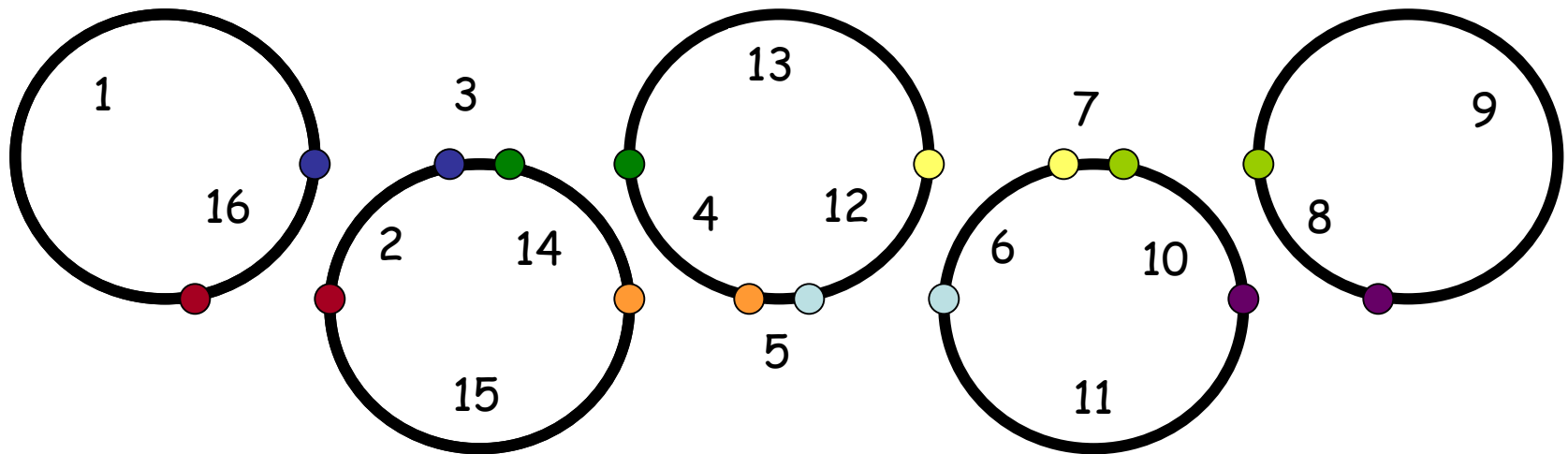


We can "detour" to the fourth cycle at an "intersection point".

# Eulerian Cycle

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

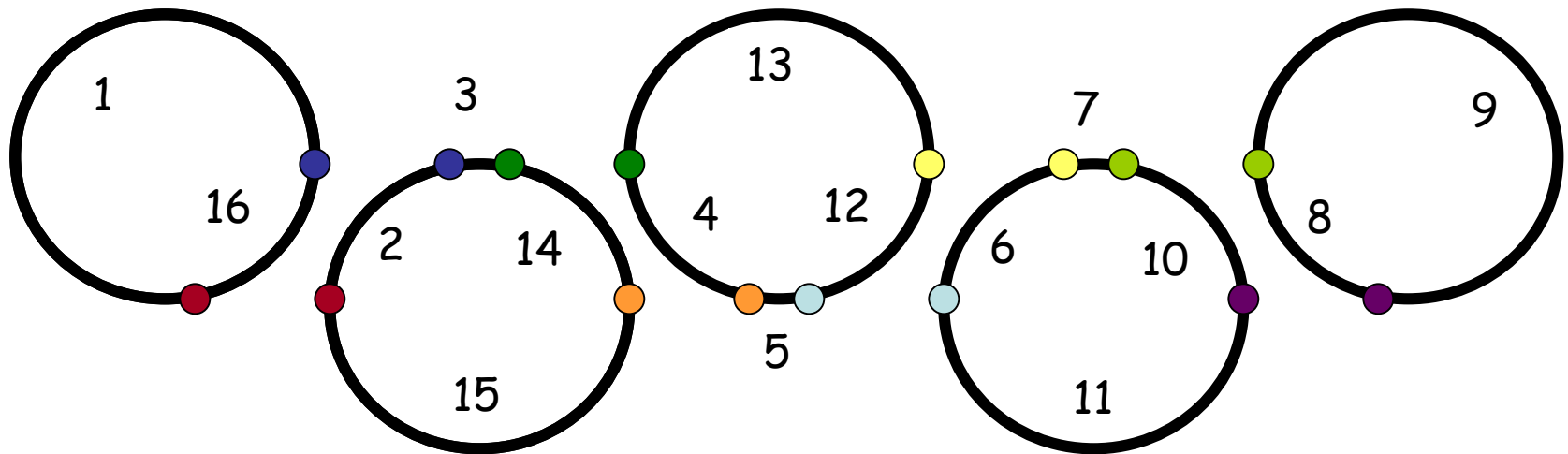
We use the same idea to deal with the fifth cycle.



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

# Eulerian Cycle

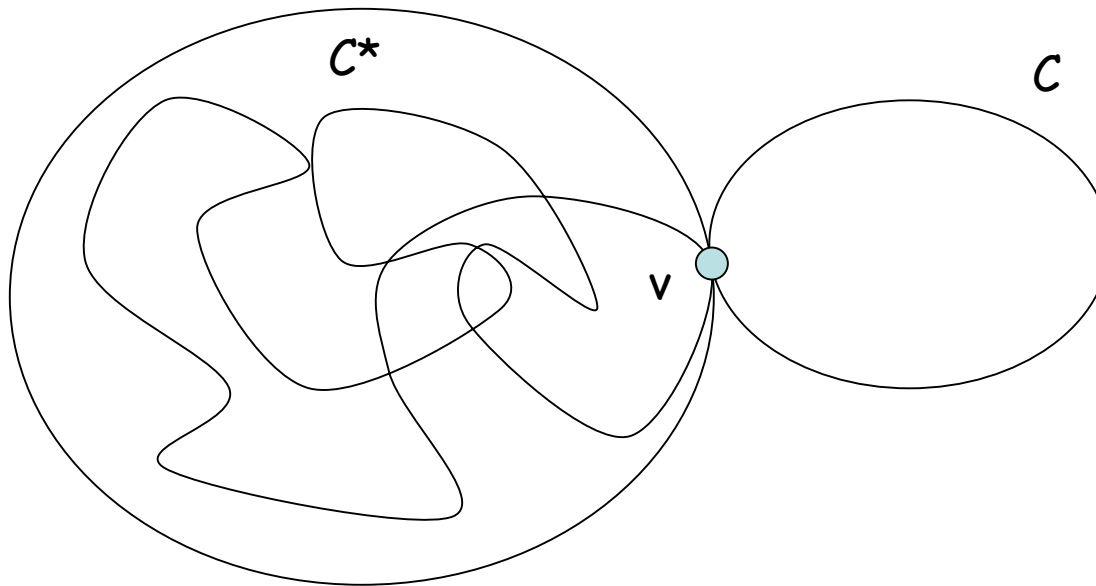
**Euler's theorem.** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.



So we have found an Eulerian cycle in 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 include the cycle  $C$  in the “partial Eulerian cycle”  $C^*$ .



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.

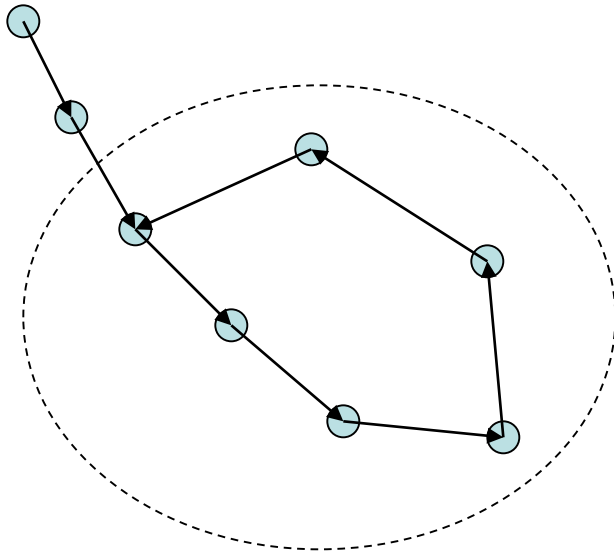
We can prove Euler's theorem if we can prove the following claim.

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

# Partitioned into Simple Cycles

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

First we can find one cycle by the same idea as before.



Start from any vertex.

Go to unvisited edges as far as possible.

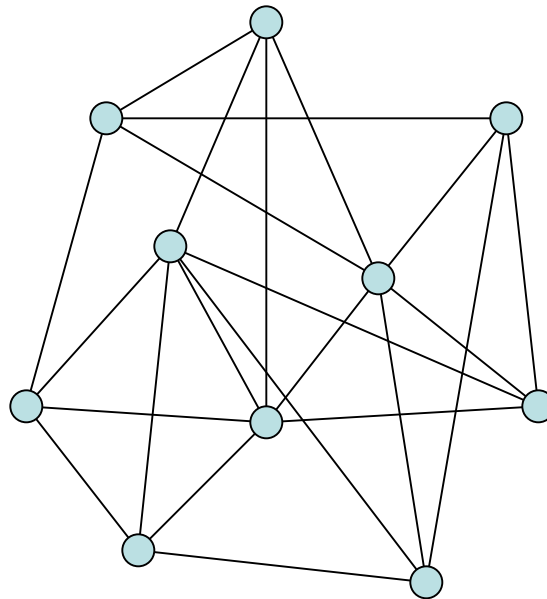
Finitely many  
vertices



there is a cycle.

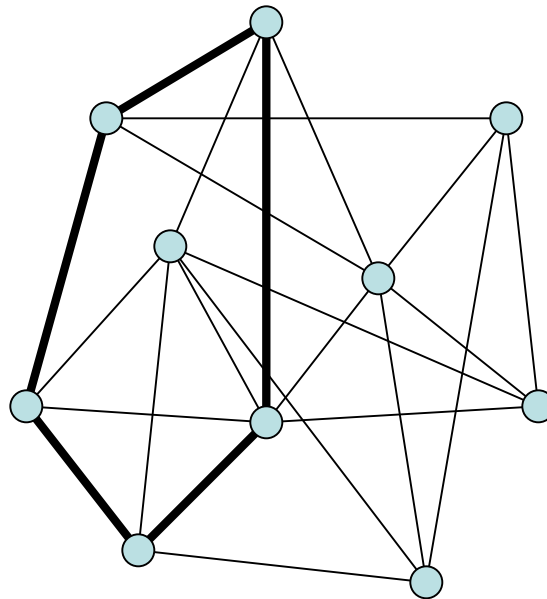
# Partitioned into Simple Cycles

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



# Partitioned into Simple Cycles

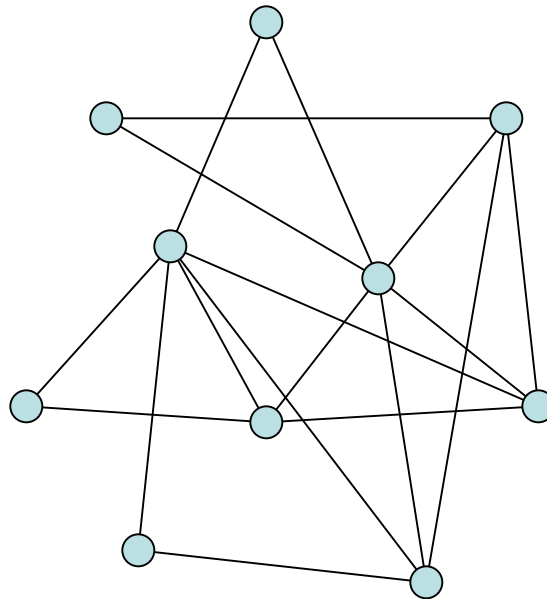
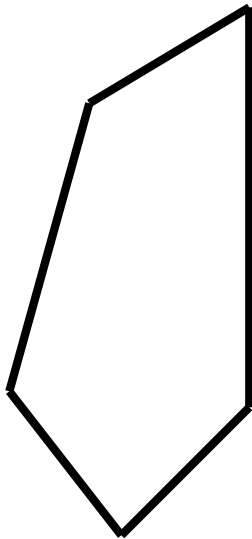
**Claim 2.** If every vertex is of even degree, then the edges can be partitioned into simple cycles.





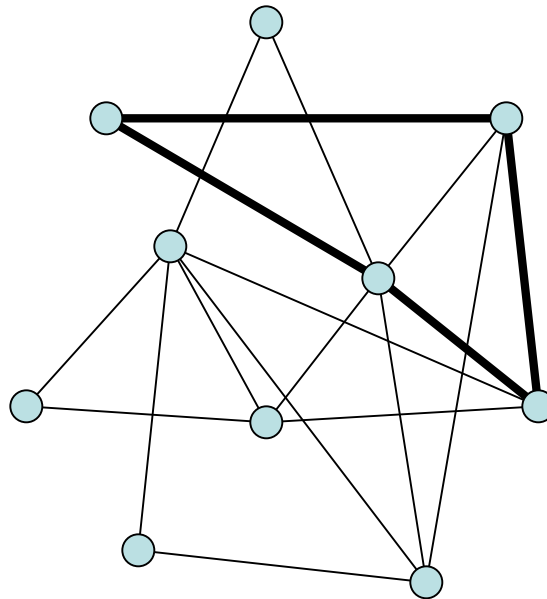
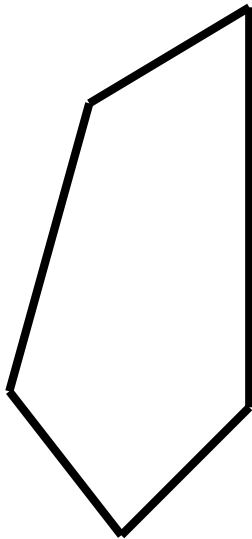
## Partitioned into Simple Cycles

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



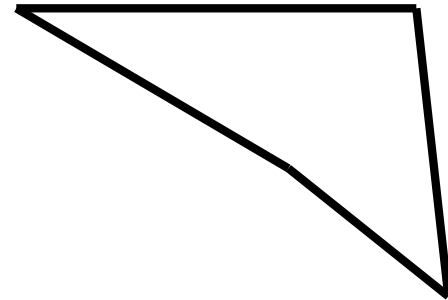
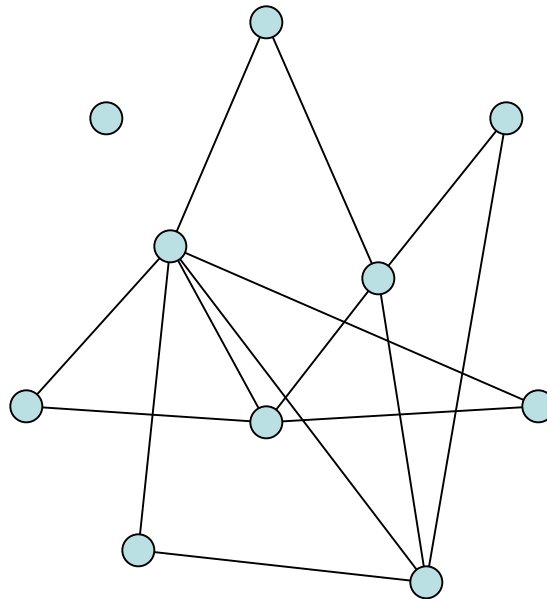
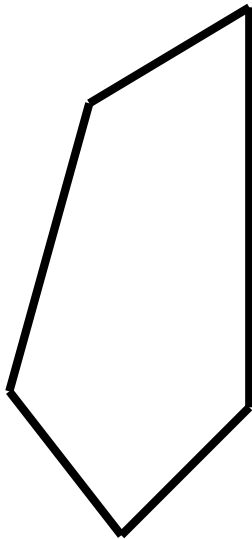
# Partitioned into Simple Cycles

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



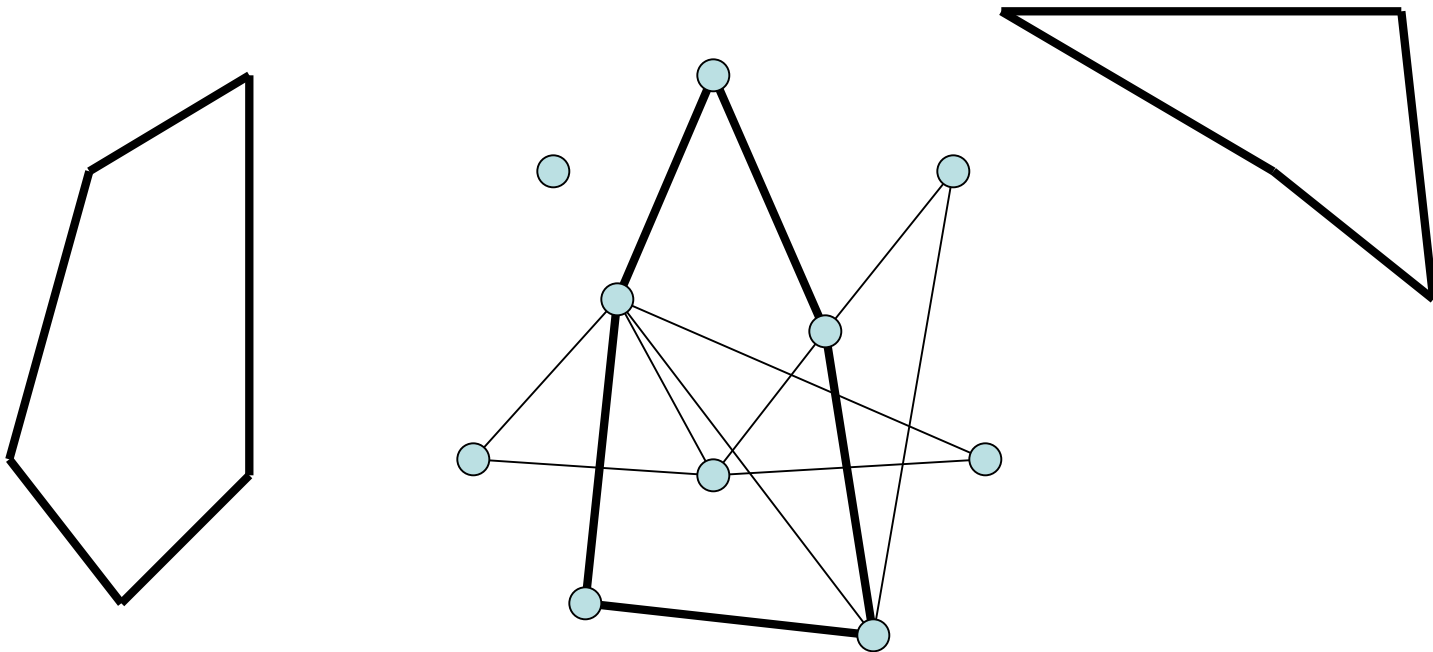
## Partitioned into Simple Cycles

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



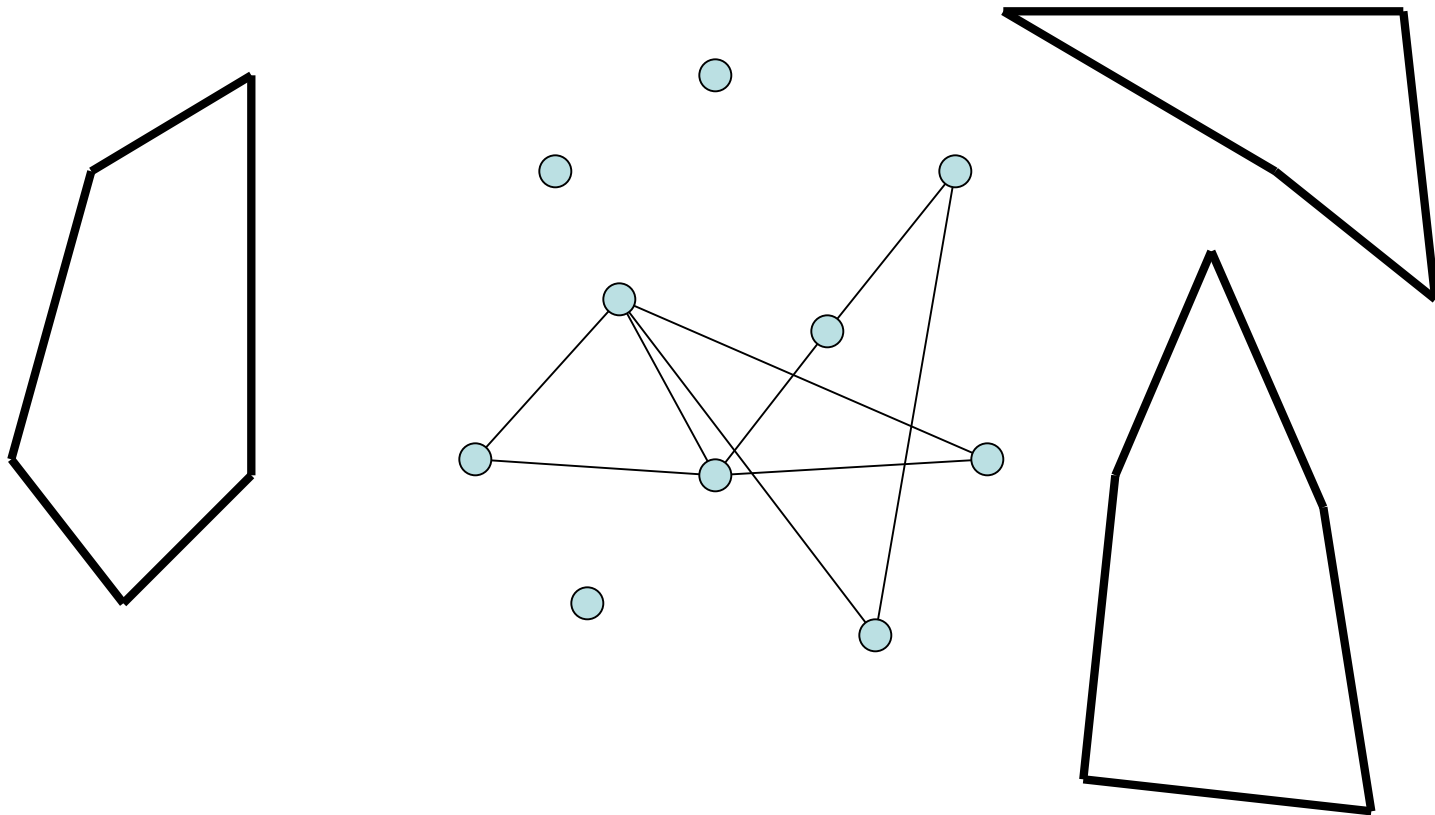
## Partitioned into Simple Cycles

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



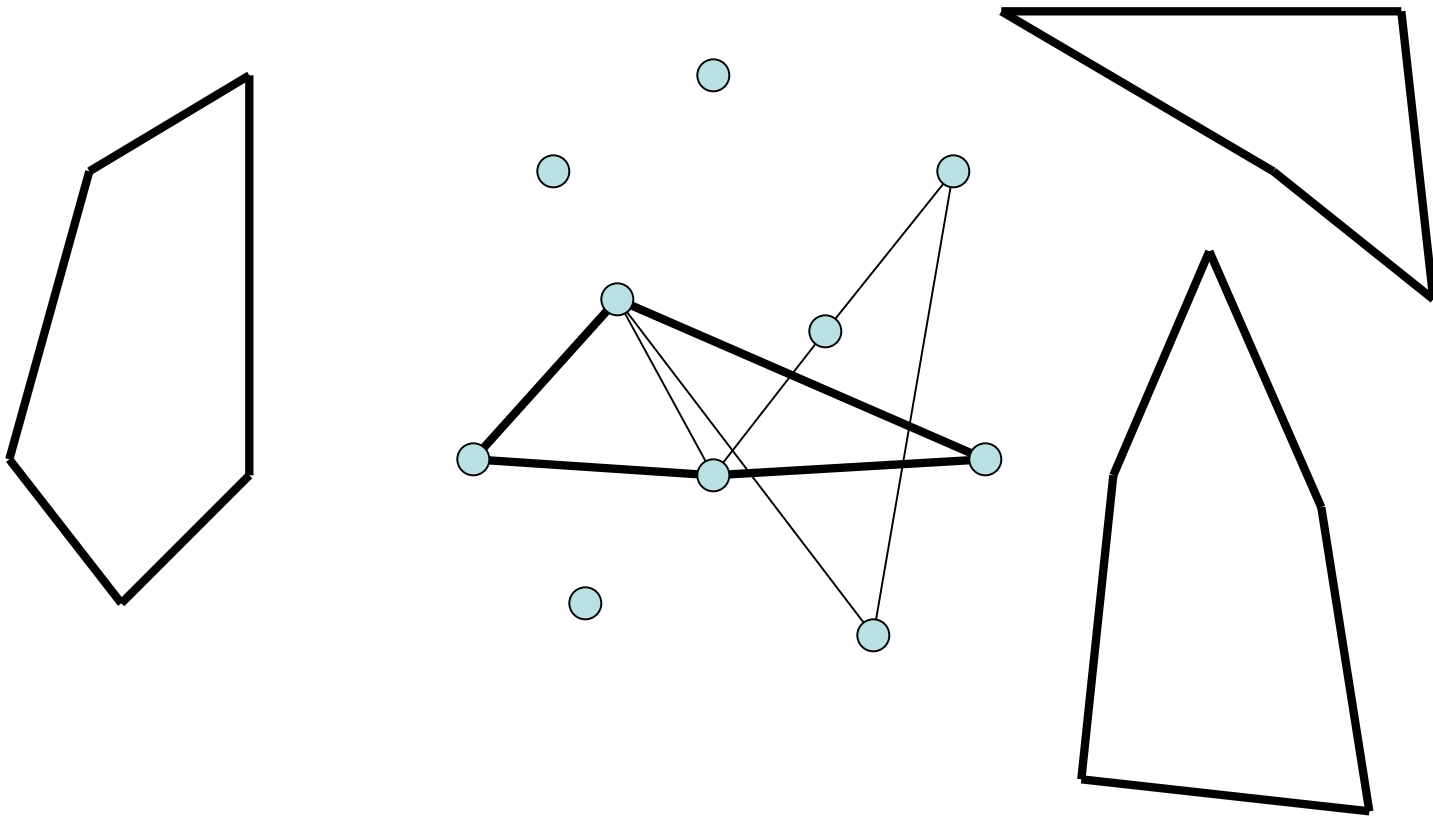
# Partitioned into Simple Cycles

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



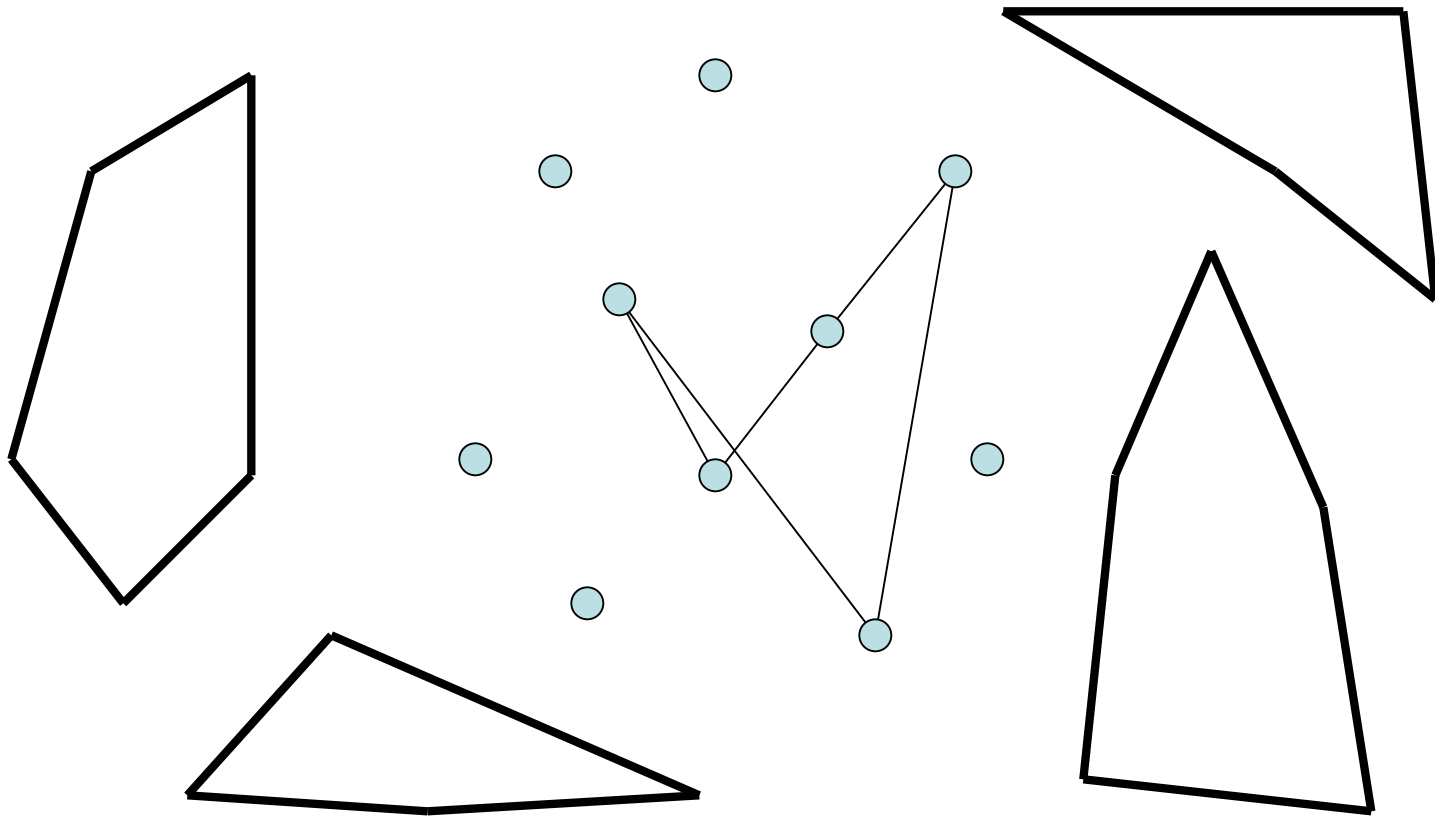
## Partitioned into Simple Cycles

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



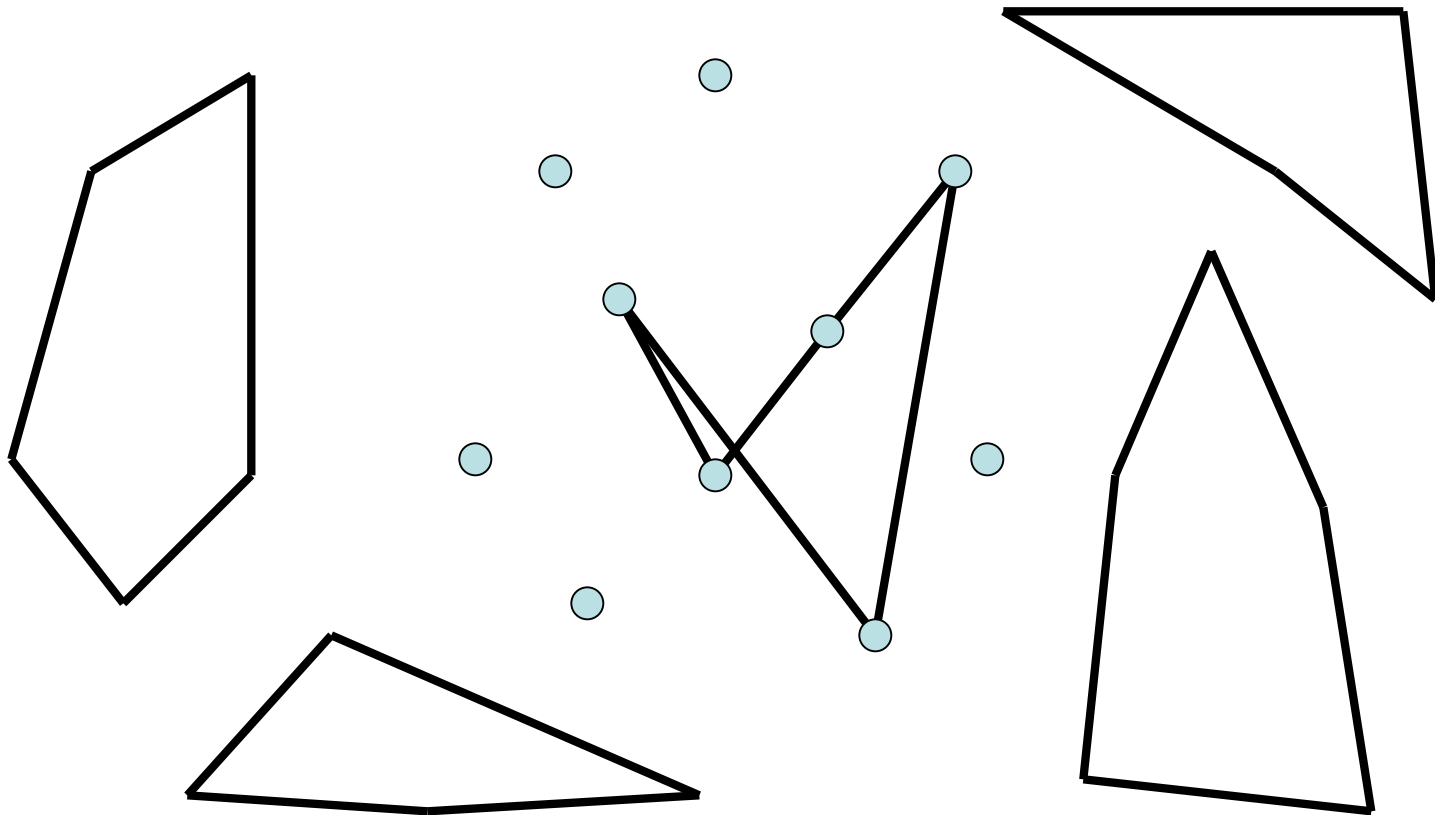
## Partitioned into Simple Cycles

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



## Partitioned into Simple Cycles

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





## Partitioned into Simple Cycles

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

**Proof.** Let  $C$  be a simple cycle.

Remove the edges in  $C$  from the graph  $G$  and call the new graph  $G'$ .

So the degree of each vertex is either unchanged or decreased by two.

So every vertex of the graph  $G'$  is still of even degree.

Note that  $G'$  has fewer edges than  $G$ .

In finite steps,  $G'$  can be partitioned into simple cycles  $C_1, C_2, \dots, C_k$ .

So the original graph  $G$  can be partitioned into simple cycles  $C_1, C_2, \dots, C_k$ .

# 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.

We proved the following claim by induction.

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

So now we have:

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

# Proof

Recall the theorem we wanted to prove..

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

How can we deal with this case?

We can reduce it to the zero vertex case!

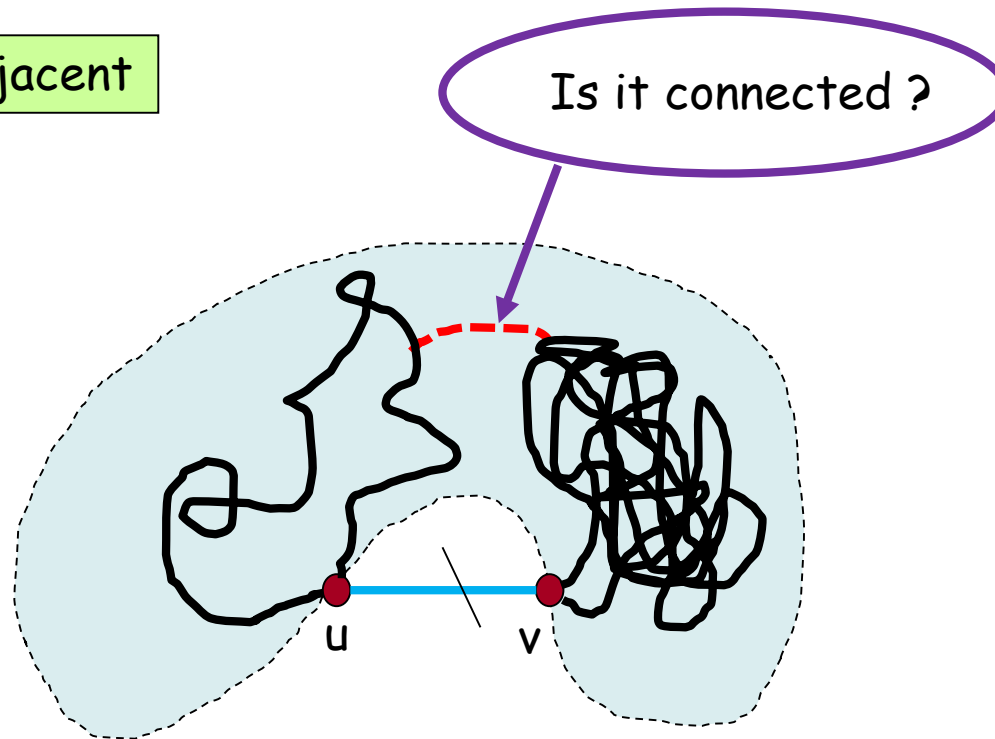
**Euler's theorem.** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.

# Odd-Degree Case is Eulerian

**Claim 3.** If a connected graph has two vertices of odd degree, then it has an Eulerian path.

Assume  $u, v$  are the two vertices with odd degrees.

Case 1:  $u, v$  are adjacent



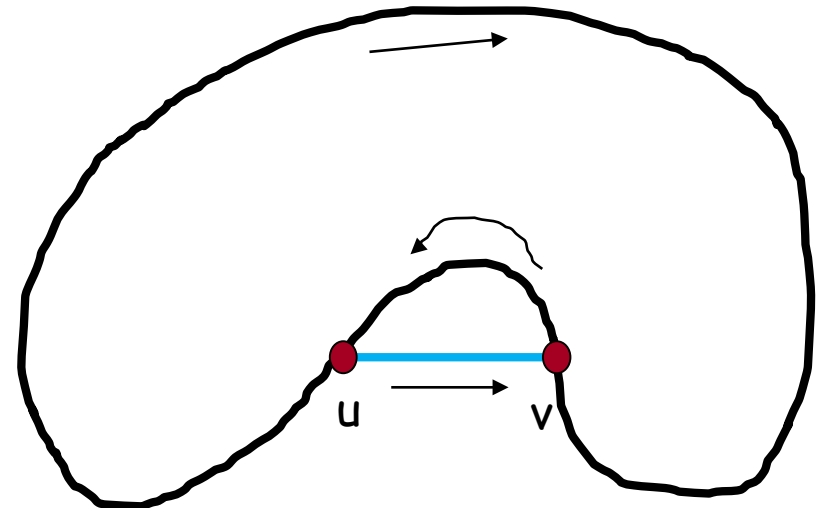
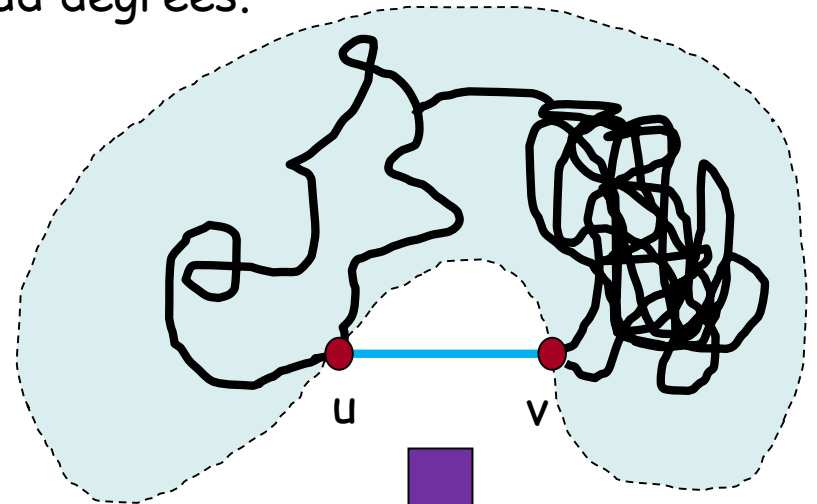
# Odd-Degree Case is Eulerian

Assume  $u, v$  are the two vertices with odd degrees.

Case 1:  $u, v$  are adjacent

Is it connected?

YES!



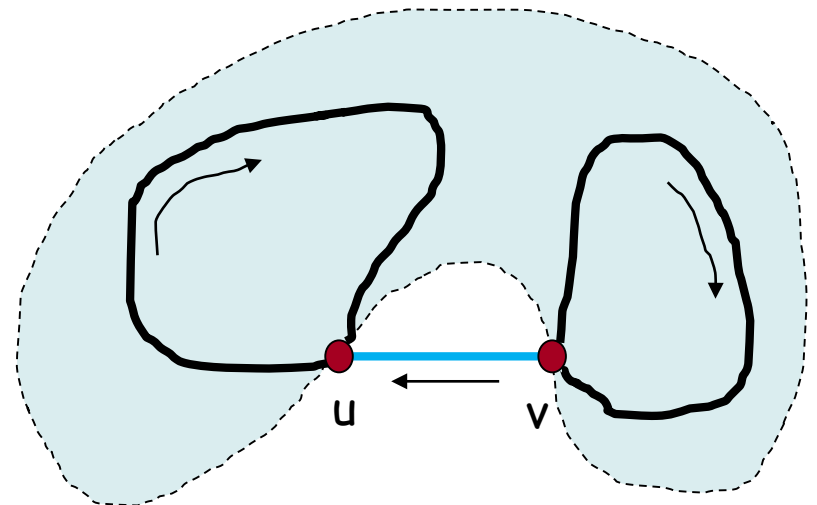
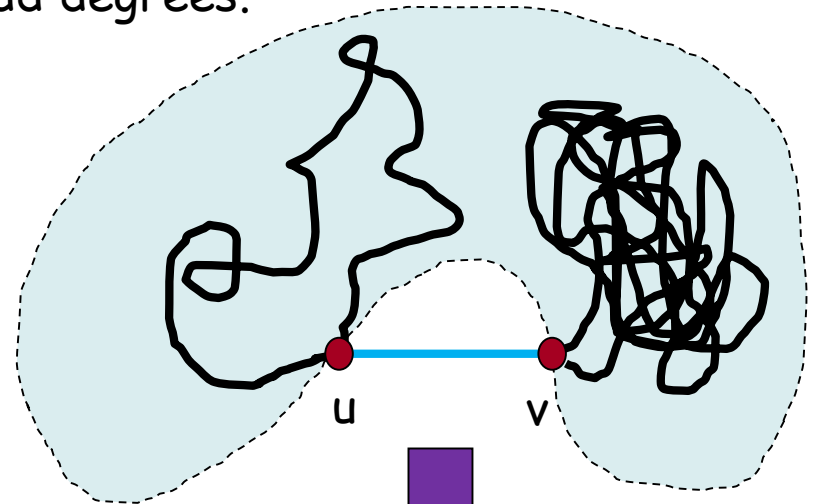
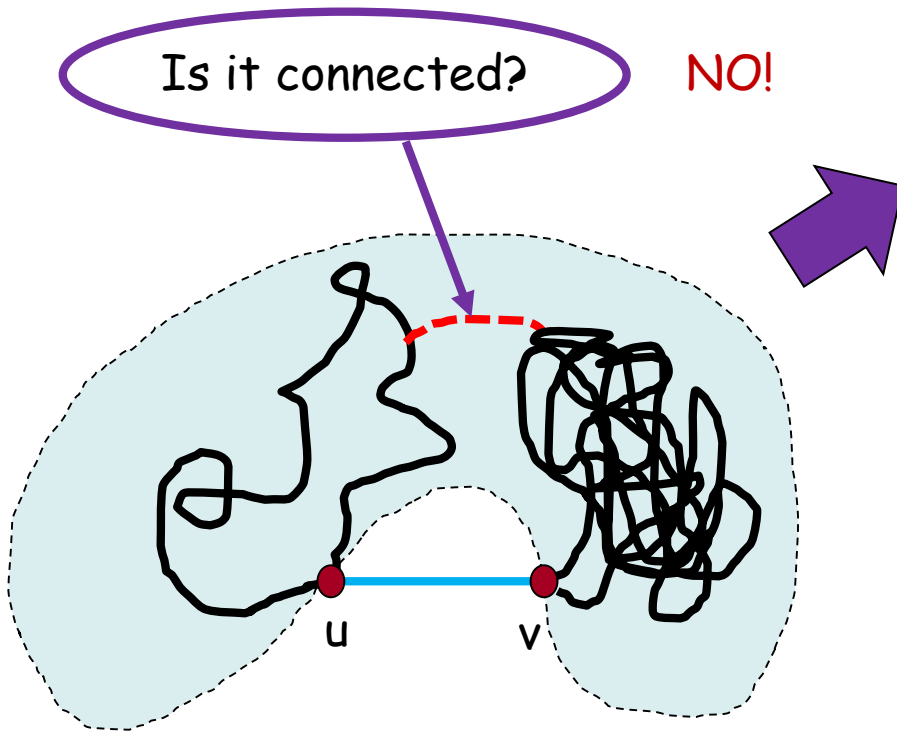
# Odd-Degree Case is Eulerian

Assume  $u, v$  are the two vertices with odd degrees.

Case 1:  $u, v$  are adjacent

Is it connected?

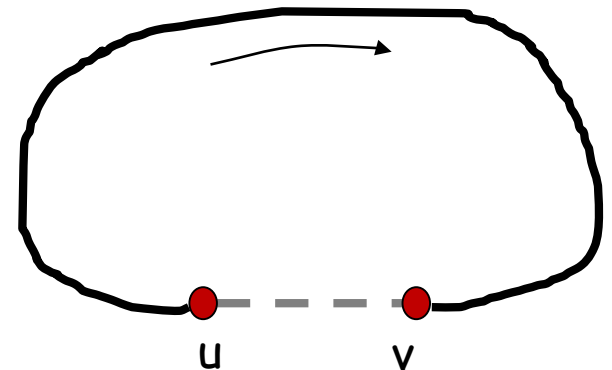
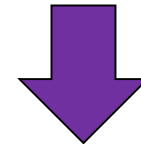
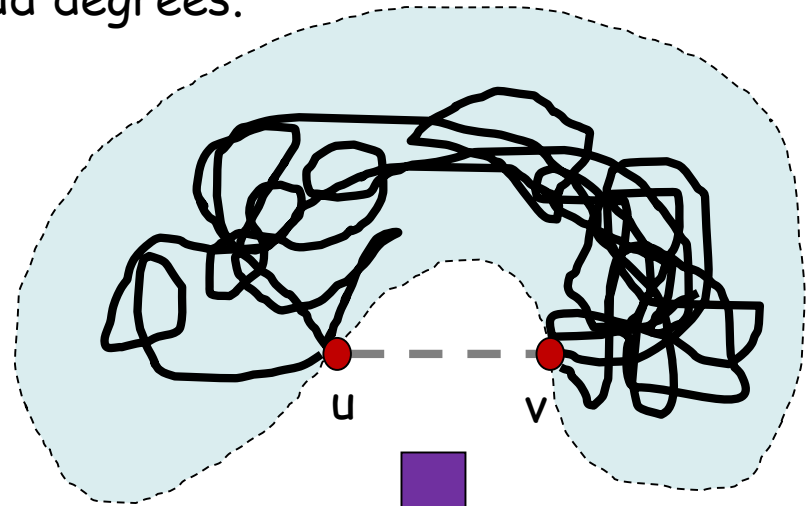
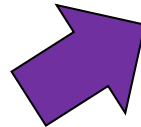
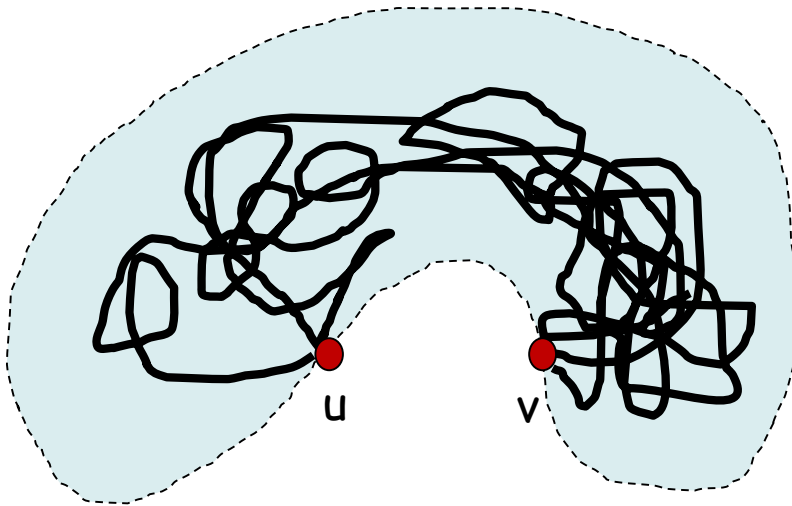
NO!



# Odd-Degree Case is Eulerian

Assume  $u, v$  are the two vertices with odd degrees.

Case 2:  $u, v$  are not adjacent



# Proof

Now we have proved the following claim.

**Claim 3.** If a connected graph has two vertices of odd degree, then it has an Eulerian path.

Recall that we have proved the following theorem.

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

So we are finally done! 

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



# This Lecture

- Seven bridges of Königsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Directed graphs

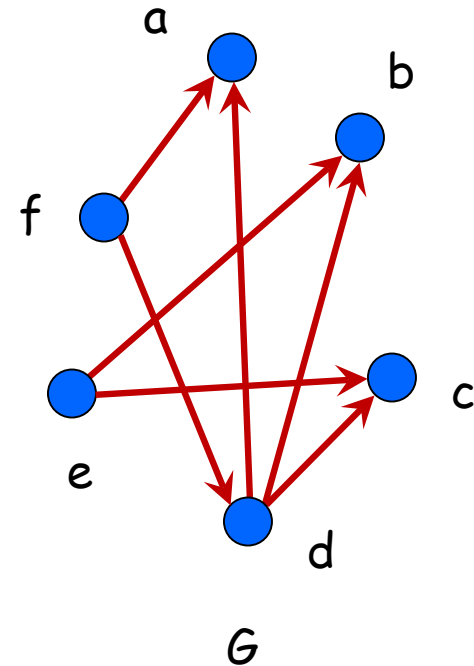
# Directed Graphs

A digraph  $G=(V,A)$  consists of:

a set of vertices,  $V$

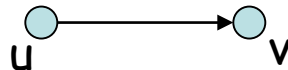
a set of directed edges (**arcs**),  $A$

- $V(G) = \{a,b,c,d,e,f\}$
- $A(G) = \{da, fa, db, eb, dc, ec, fd\}$



For an arc  $uv$ , we say  $u$  is the **tail** of the arc and  $v$  is the **head** of the arc.

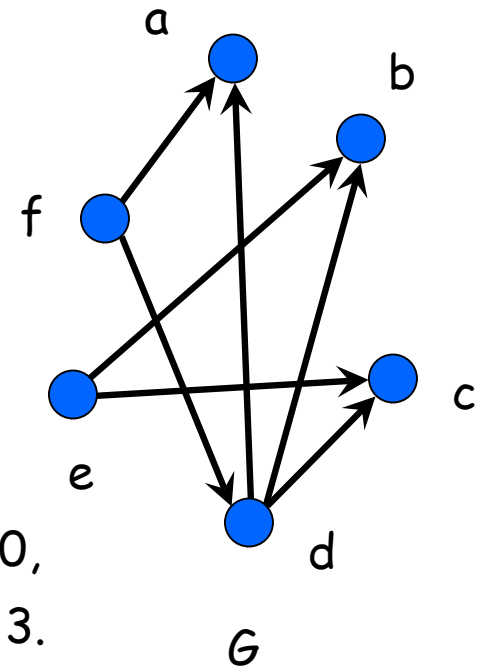
Also, we say  $v$  is an **out-neighbor** of  $u$ , and  $u$  is an **in-neighbor** of  $v$ .



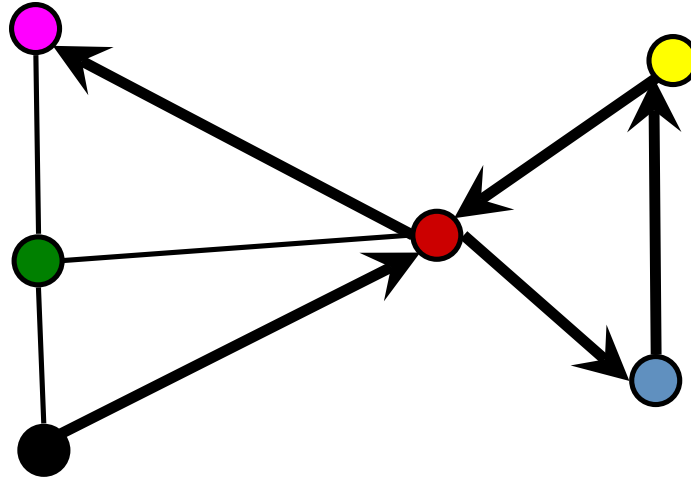
# In-Degrees and Out-Degrees

The **out-degree** of a vertex  $v$  is the number of out-neighbors of  $v$ ; similarly, the **in-degree** of a vertex  $v$  is the number of in-neighbors of  $v$ .

E.g. the in-degree of  $a$  is 2 and the out-degree of  $a$  is 0,  
the in-degree of  $d$  is 1 and the out-degree of  $d$  is 3.



# Directed Paths



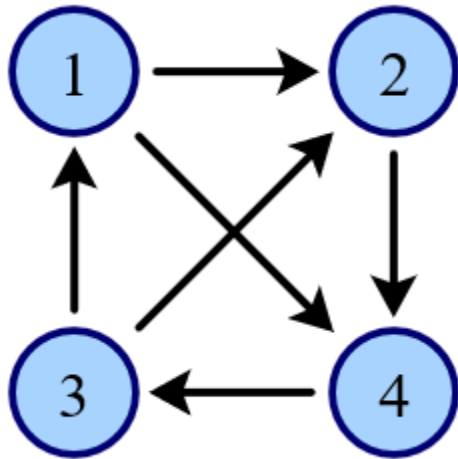
Informally, a directed path is a path that follows the arcs.

*Directed Path*: sequence of vertices  $v_1, v_2, \dots, v_k$  such that there is an arc from  $v_i$  to  $v_{i+1}$  for each  $i = 1, \dots, k-1$ .

*Simple Directed Path*: a directed path with no repeated vertices.

# Directed Cycles

A digraph is *connected* if replacing its arcs by undirected edges results in a connected simple graph.



This is a connected digraph with four vertices.

(1,4,3) is a directed cycle;  
(2,4,3) is also a directed cycle.

But (1,2,3) is not a directed cycle;  
neither is (1,2,4).

A *directed cycle* is a connected digraph that the out-degree and in-degree are both 1 for every vertex.

# Eulerian Problem in Directed Graphs

Given a directed graph, when is it possible to have a directed cycle that visits every arc exactly once?

This is similar to the case of simple graphs.

