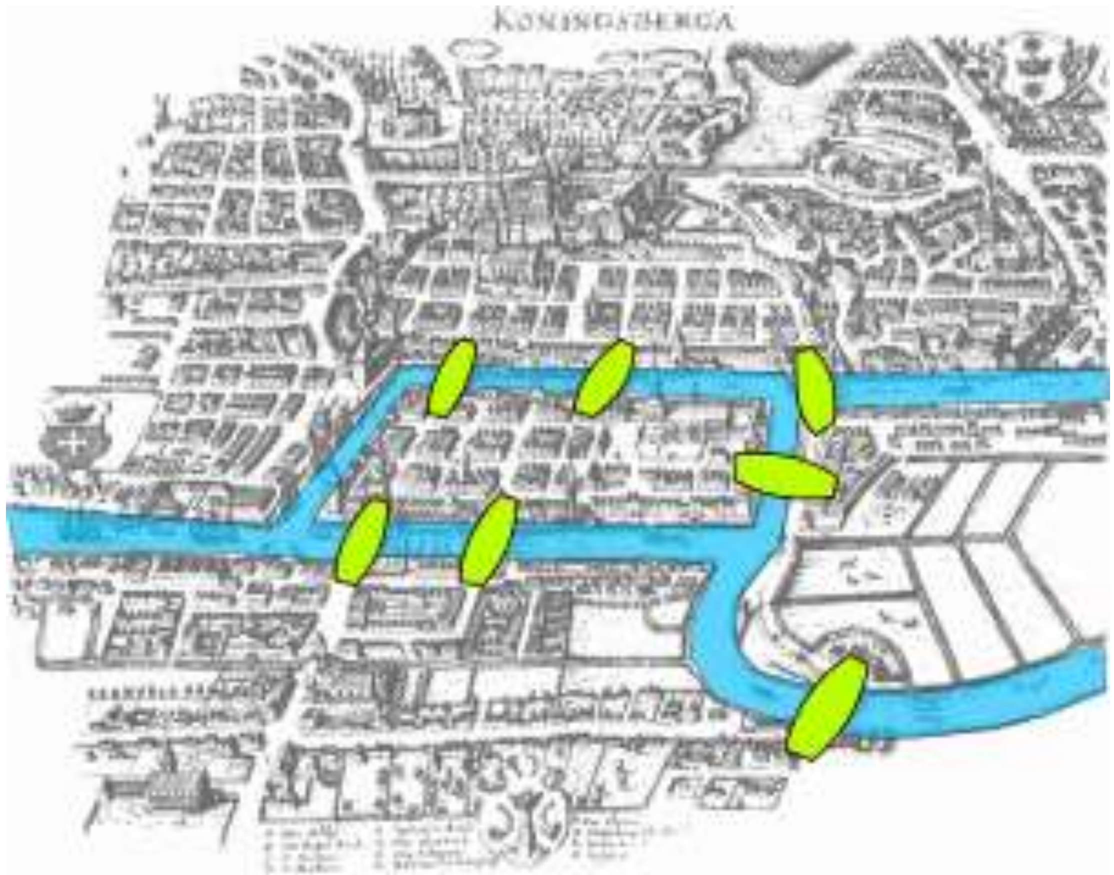
# Introduction to Graphs



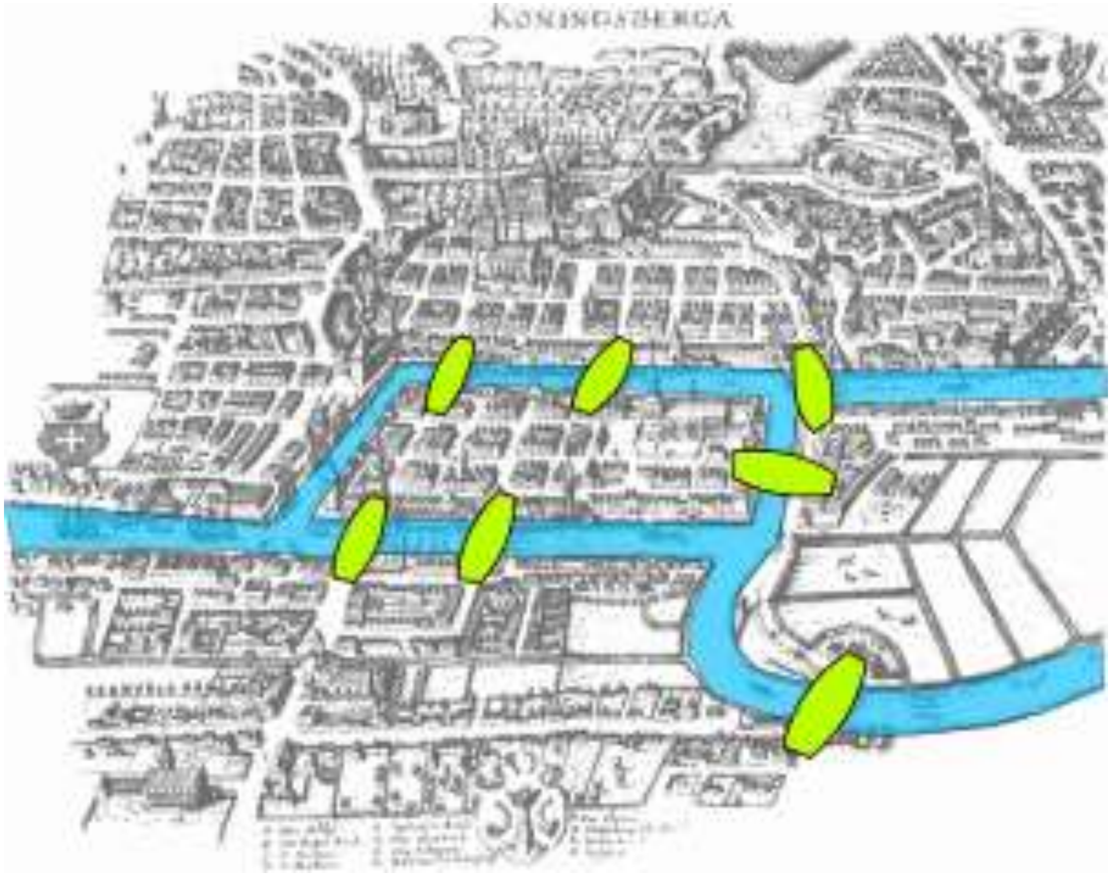
# This Lecture

In this part we will study some basic graph theory.

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

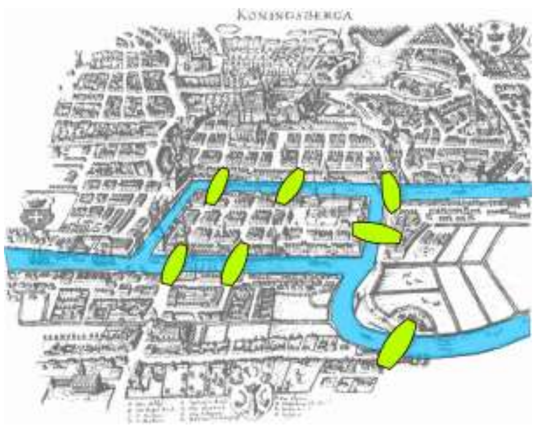
- Seven bridges of Konigsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Directed graphs

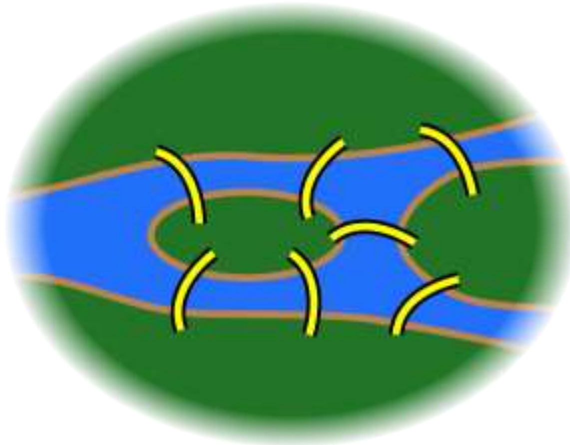
# Seven Bridges of Königsberg

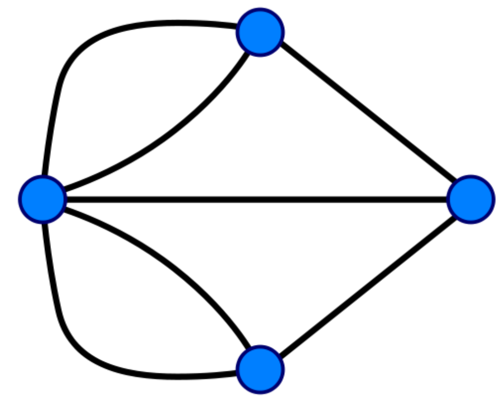



Is it possible to walk with a route that crosses each bridge exactly once?

# Seven Bridges of Königsberg



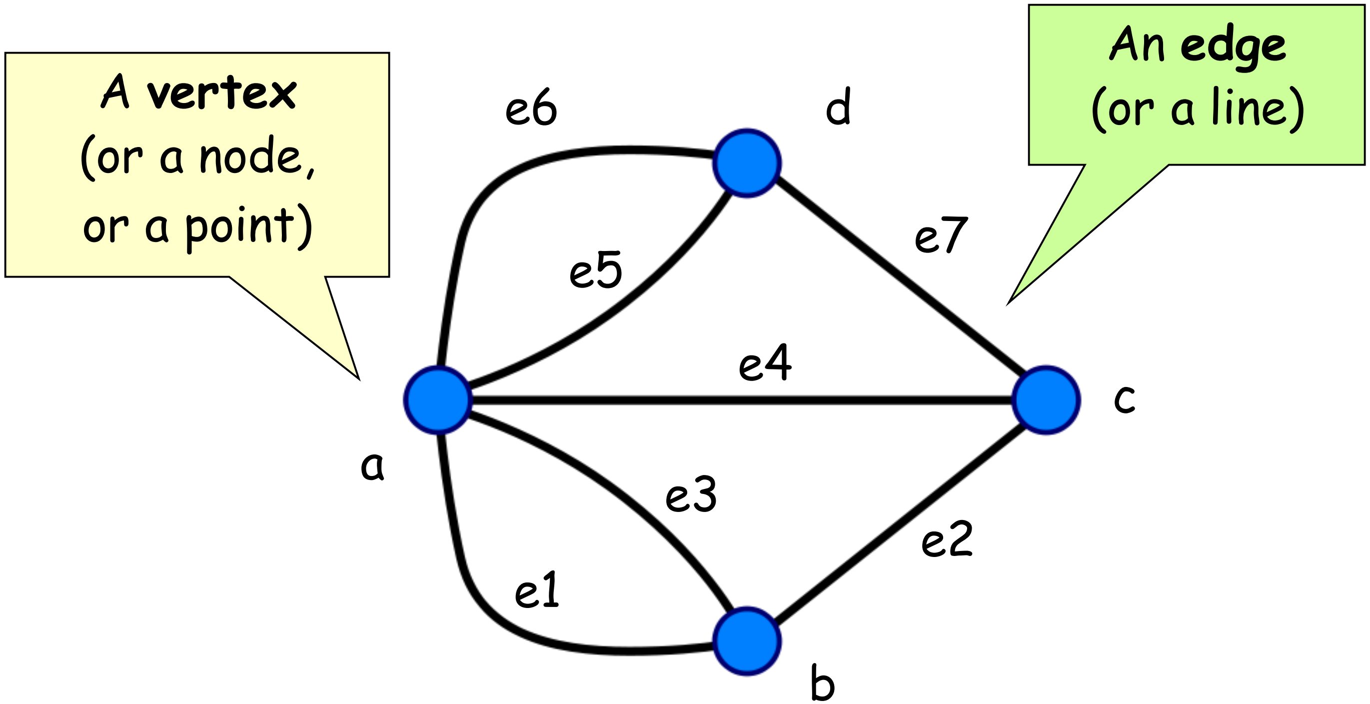




Forget unimportant details.

Forget even more.

# A Graph

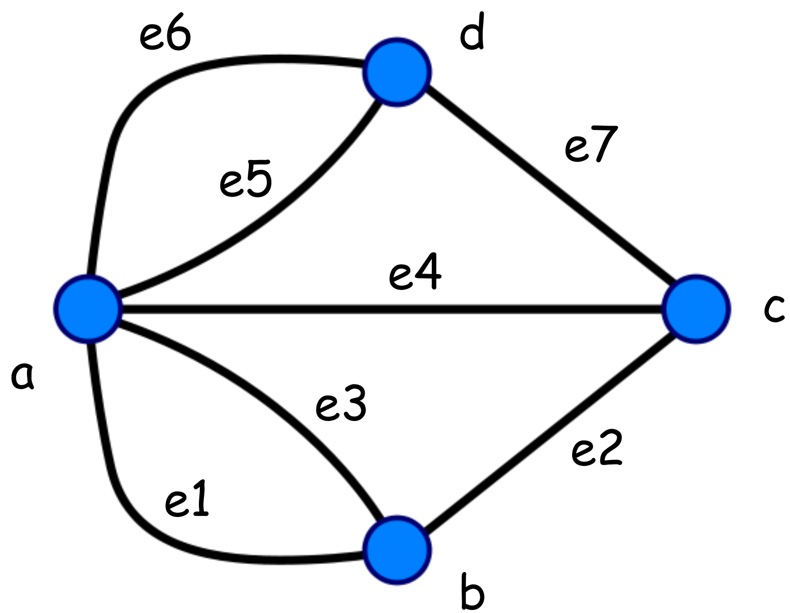


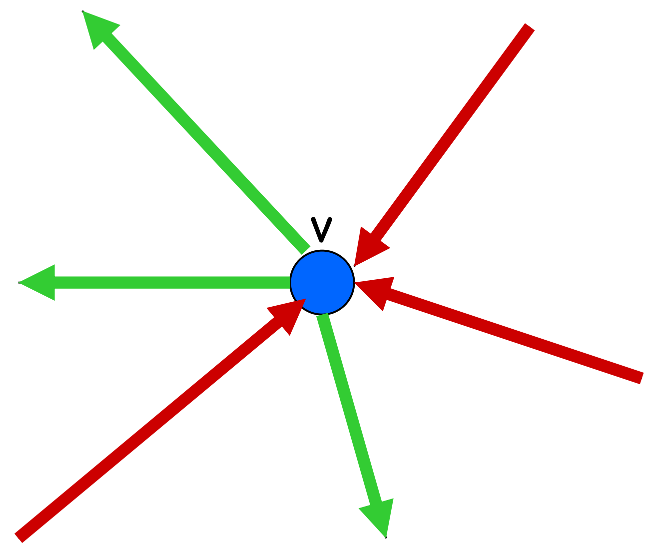

So, what is the **"Seven Bridges of Königsberg"** problem now?

To find a walk that visits each edge exactly once.

# Euler's Solution

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



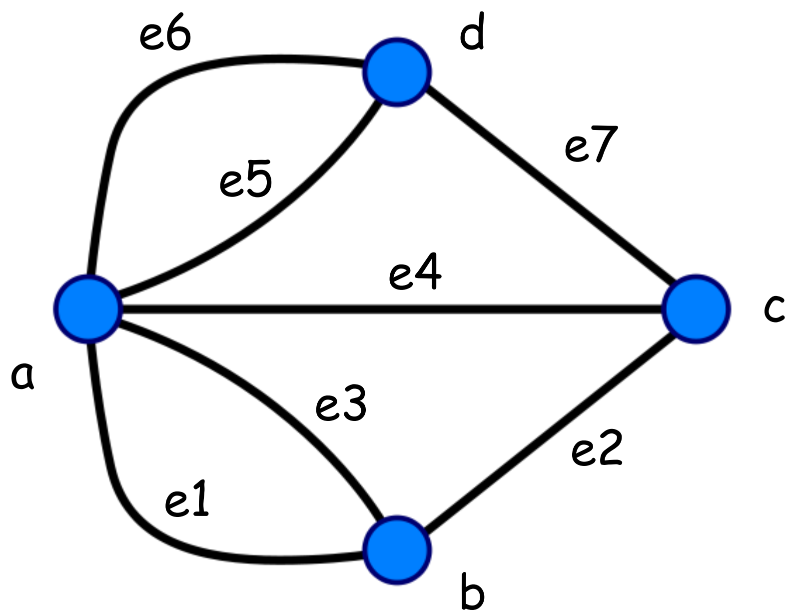




Suppose there is such a walk, there is a starting point and an endpoint point.

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.

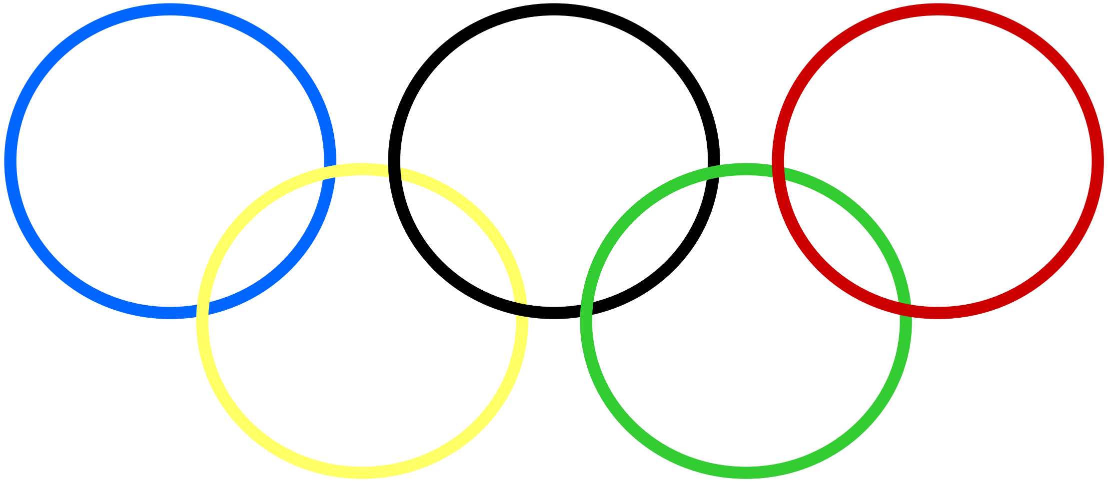
Suppose there is such a walk, there is a starting point and an endpoint point.

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?

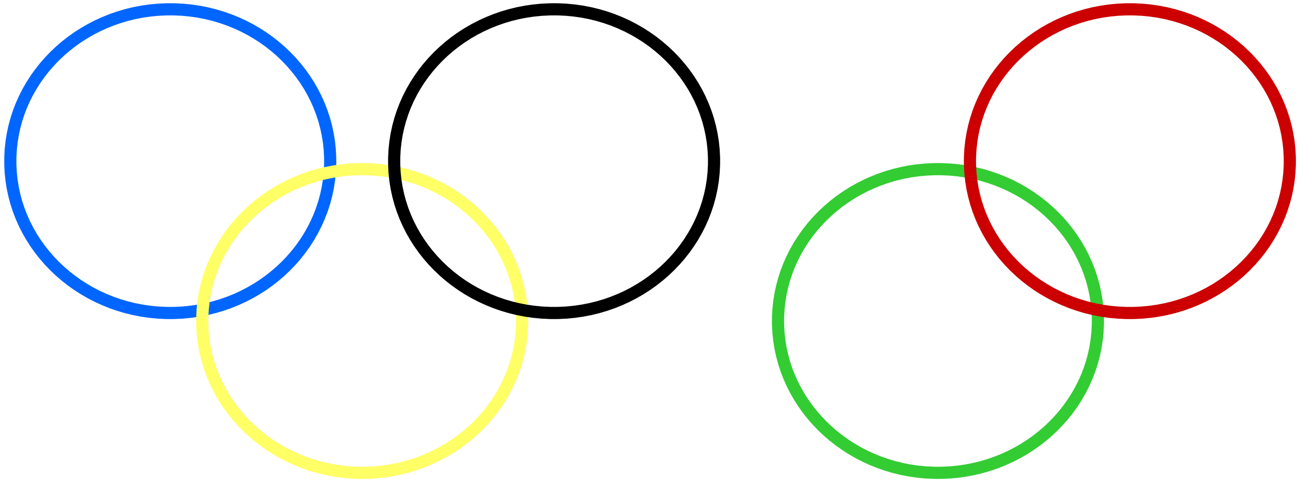


Is it always possible to find such a walk if there is **at most two** vertices with odd number of edges?

# 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?



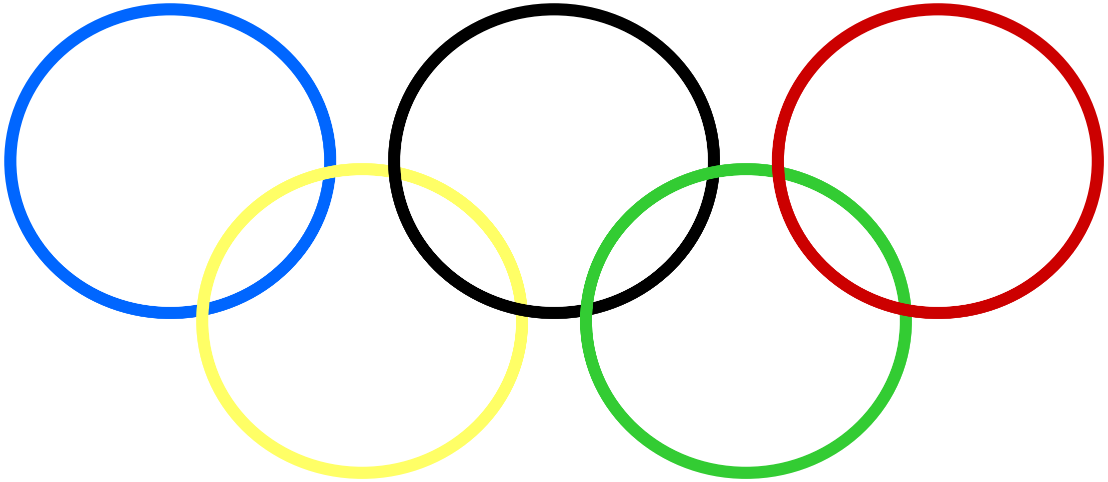
Is it always possible to find such a walk if there is **at most two** vertices with odd number of edges?

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.

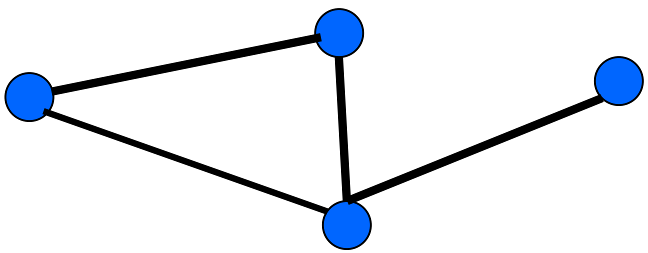
This theorem was proved in 1736,
and was regarded as the starting point of graph theory.

# This Lecture

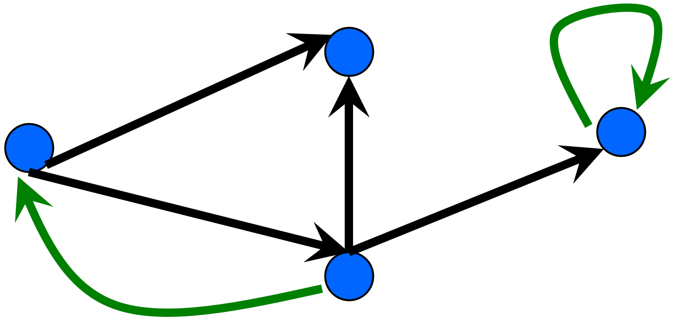
- Seven bridges of Konigsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Directed graphs

# Types of Graphs

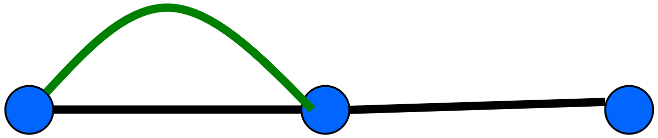
**Simple Undirected Graph**



Directed Graph



Multi-Graph



**Eulerian path problem**

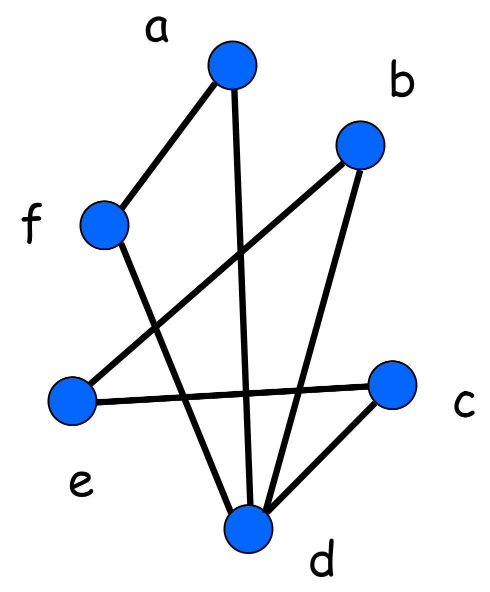
# Simple Graphs

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}

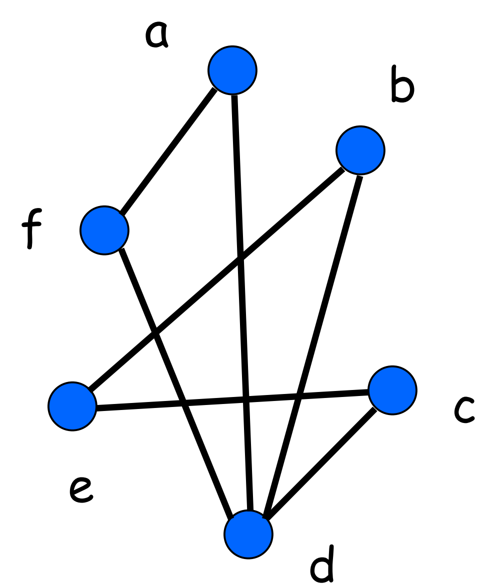


G

Two vertices a,b are ***adjacent*** (***neighbours***) if the edge ab 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

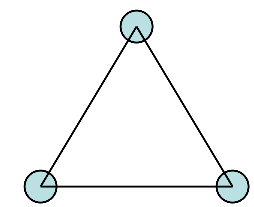
e.g. deg(d) = 4, deg(a)=deg(b)=deg(c)=deg(e)=deg(f)=2.

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

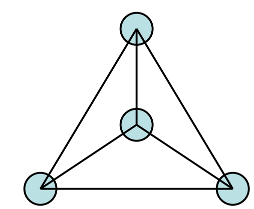
For multigraphs, NO. For simple graphs, YES.

# Degree Sequence

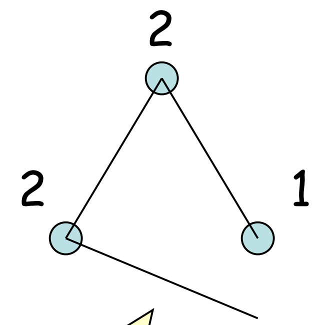
Is there a graph with degree sequence (2,2,2)? YES.



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



Is there a graph with degree sequence (2,2,1)? NO.




Where to go?

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

NO.

What's wrong with these sequences?

# 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 = odd, so impossible.

2+2+2+2+1 = odd, so impossible.

# Handshaking Lemma

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

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).

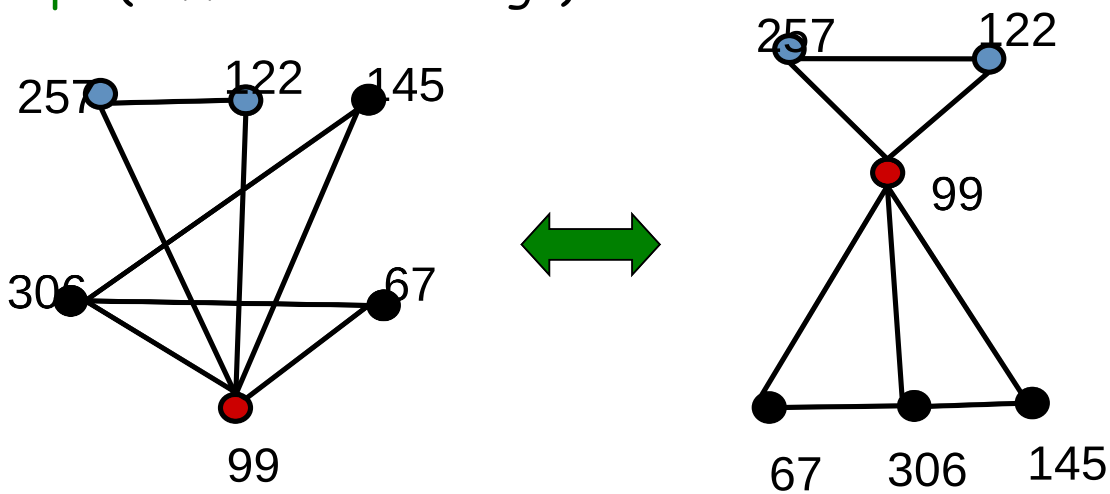
For multigraphs (with self loops), YES! (easy by induction)

# This Lecture

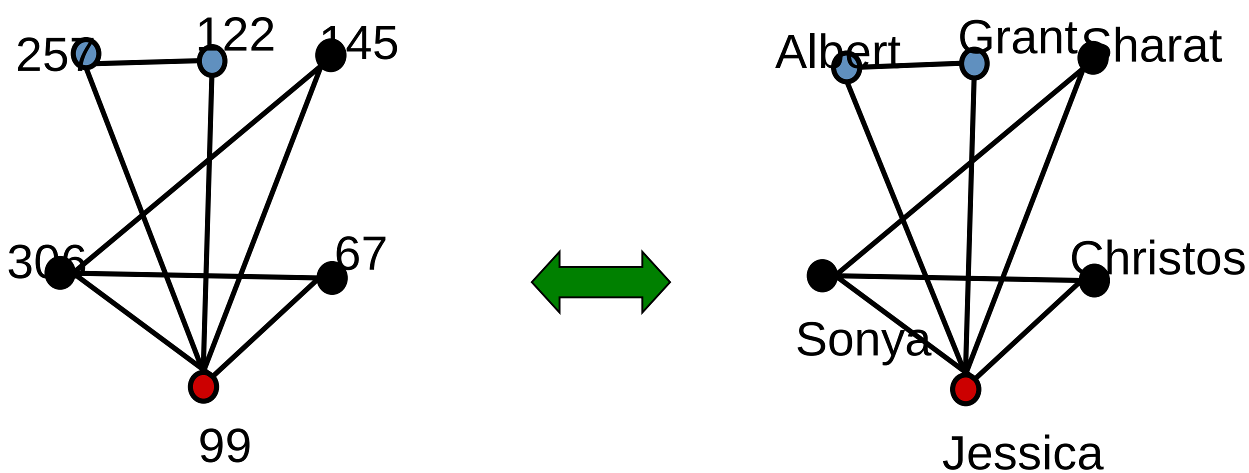
- Seven bridges of Konigsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Directed graphs

# Same Graphs?

Same graph (different *drawings*)




Same graph (different *labels*)



# Graph Isomorphism

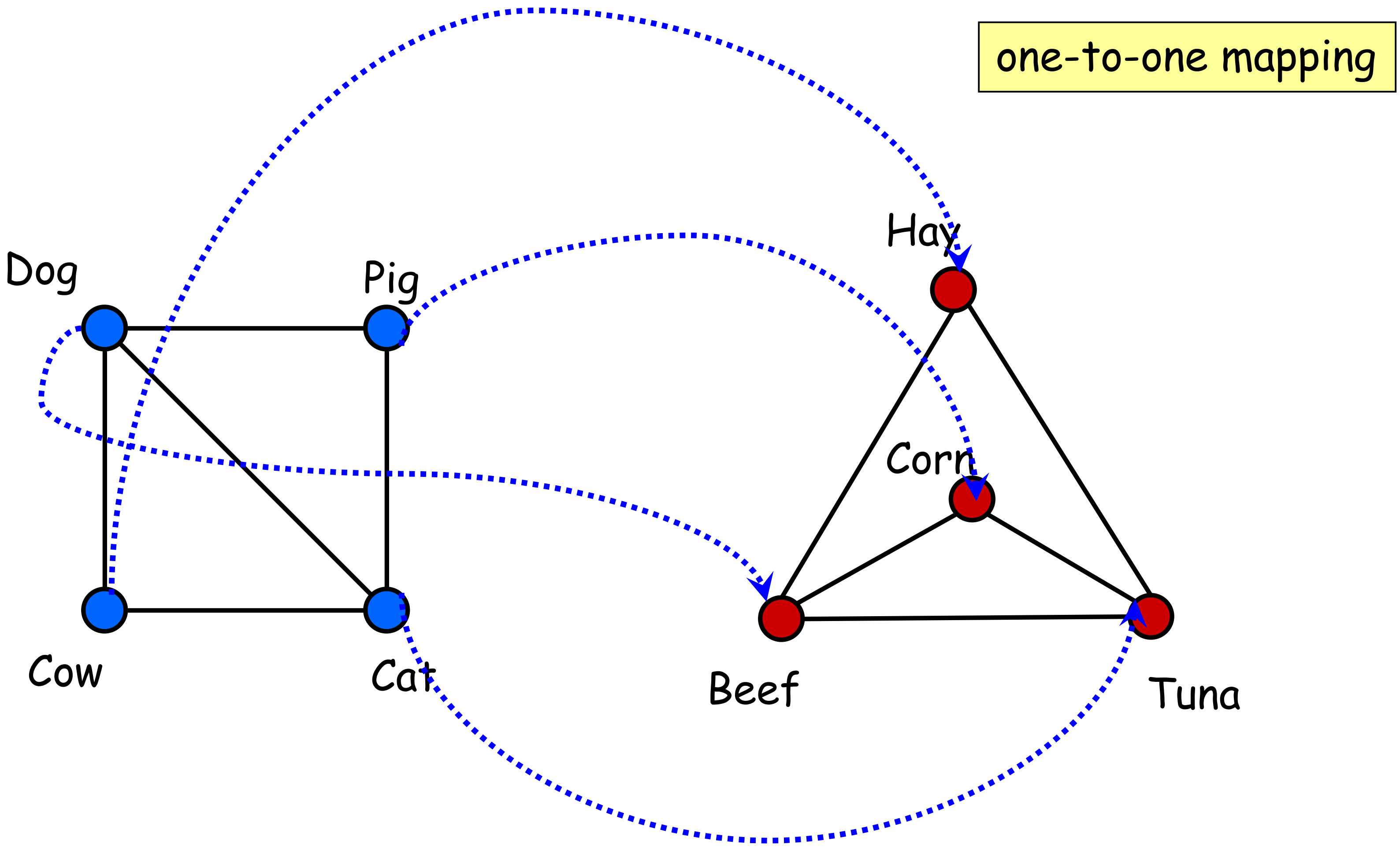
All that matters is the *connections.*

Graphs with the same connections are *isomorphic*.

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

Graph isomorphism has applications like checking fingerprint, testing molecules…

# Are These Isomorphic?



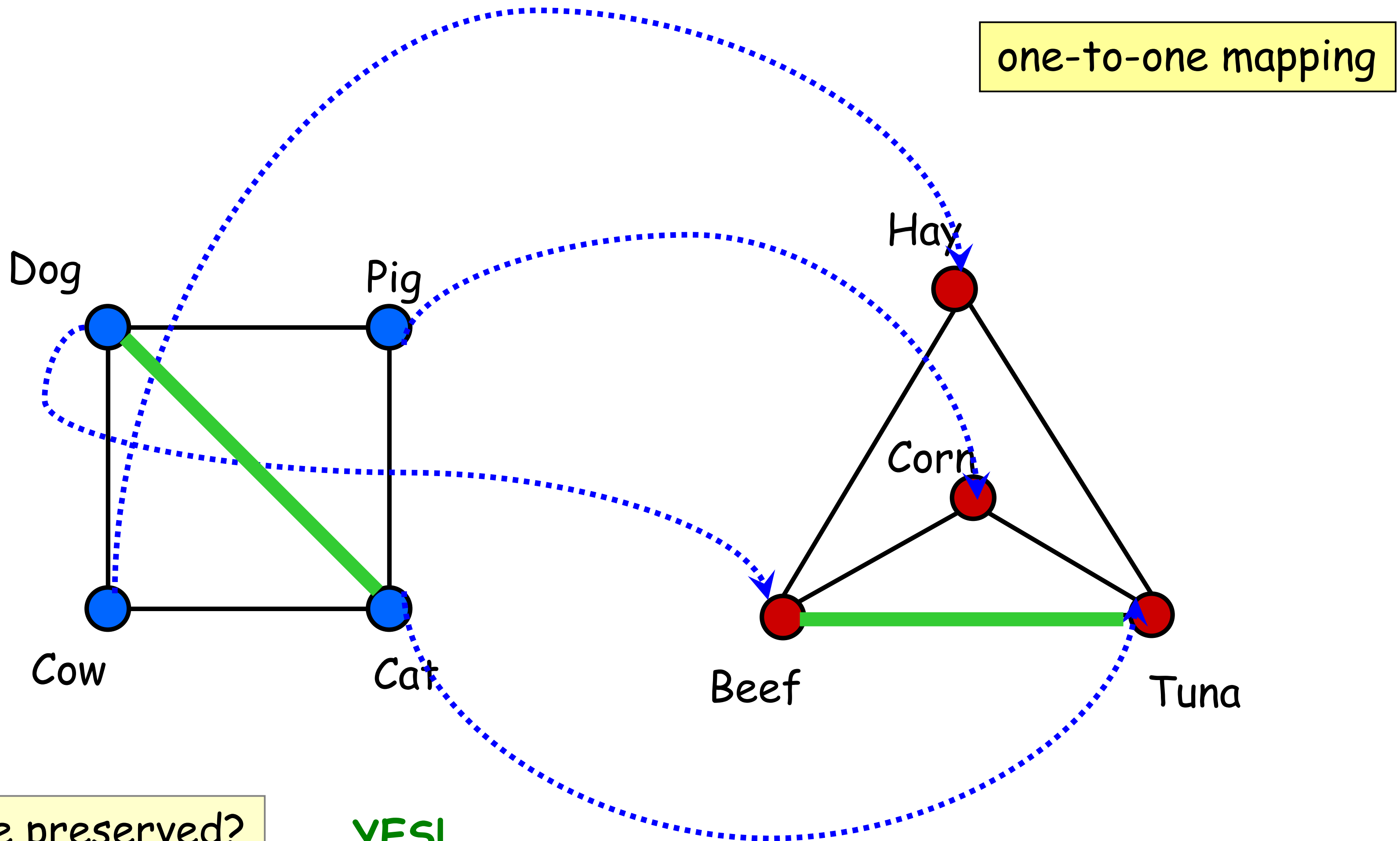

$f$ (Dog) = Beef

$f$ (Cat) = Tuna

$f$ (Cow) = Hay

$f$ (Pig) = Corn

# Are These Isomorphic?

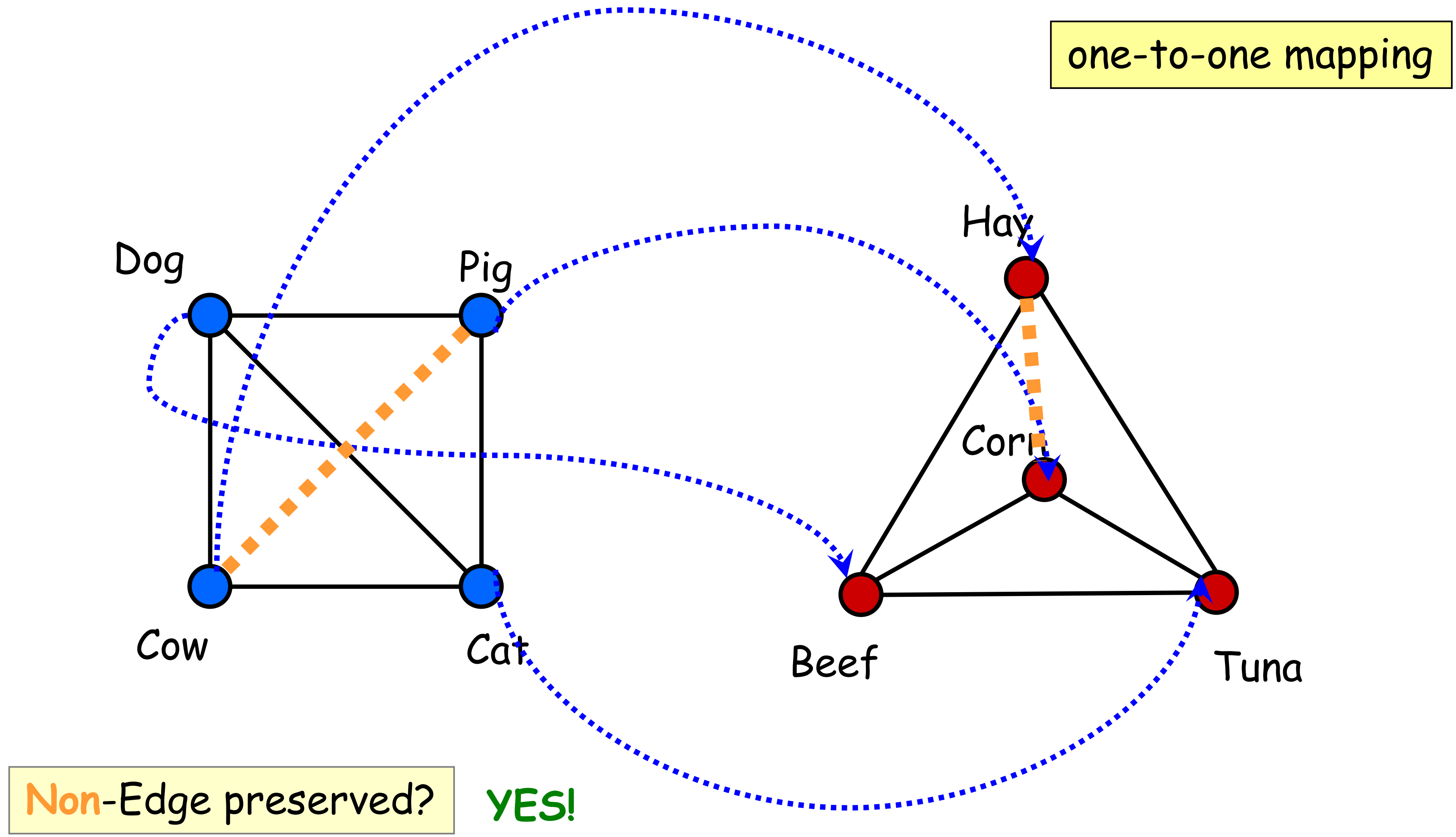



Edge preserved?

YES!

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 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 G1

f(u)f(v) is an edge in G2

- If G1 and G2 are isomorphic, do they have the same number of vertices? YES
- If G1 and G2 are isomorphic, do they have the same number of edges? YES
- If G1 and G2 are isomorphic, do they have the same degree sequence? YES
- If G1 and G2 have the same degree sequence, are they isomorphic? NO

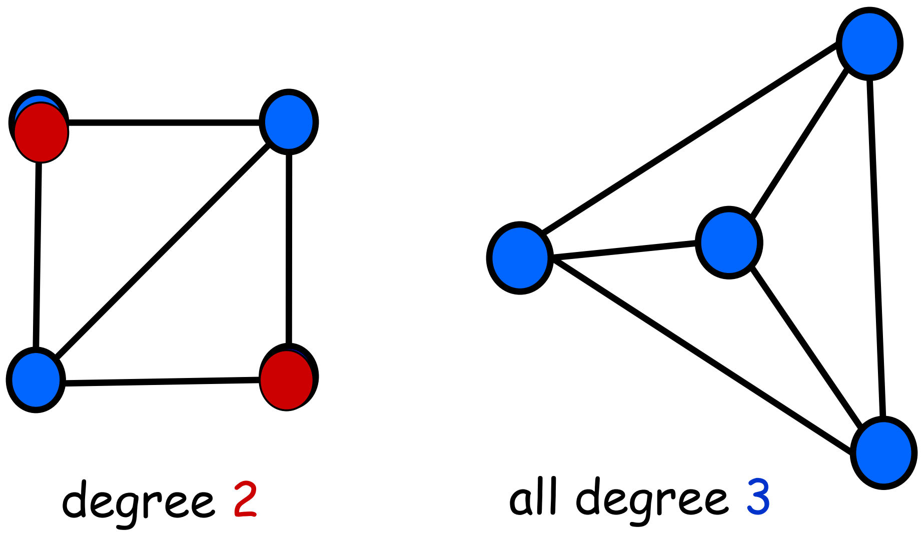
# Checking Graph Isomorphism

How to show two graphs are isomorphic?

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

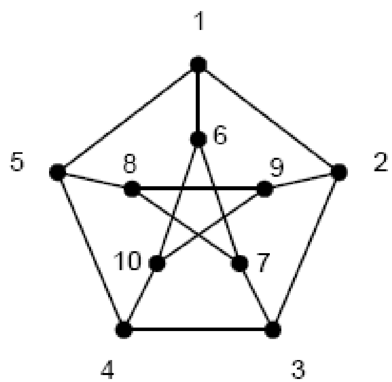
How to show two graphs are non-isomorphic?

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

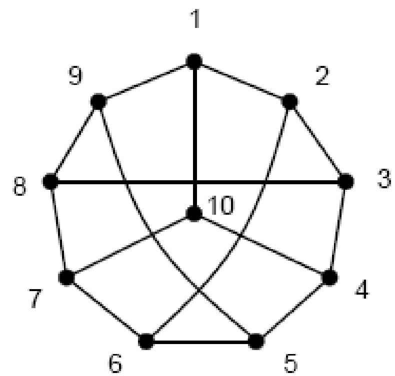



Non-isomorphic

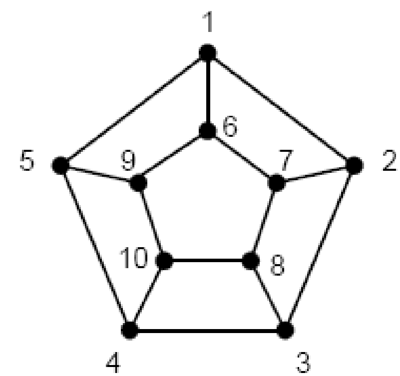
# Exercise




$G_1$

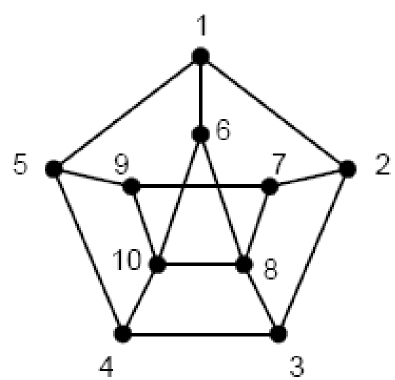



(a) $G_2$

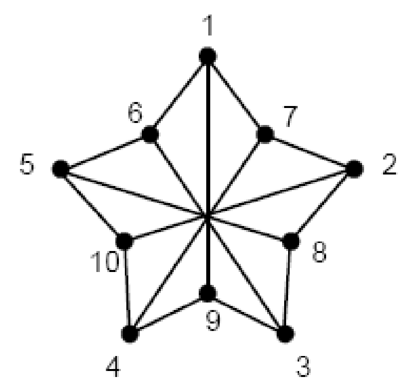



(b) $G_3$

Which is isomorphic to G1?




(c) $G_4$



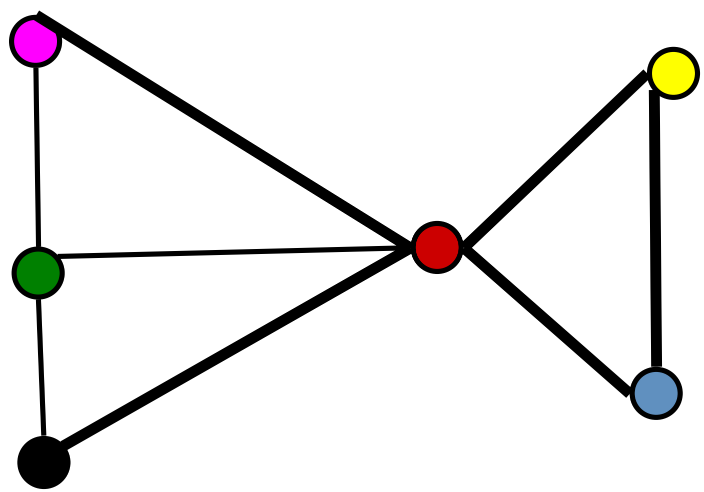

(d) $G_5$

***Testing graph isomorphism is not easy –***
No known general method to test graph ismorphism much more efficient than checking all possibilities.

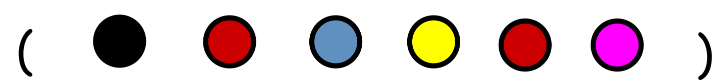
# This Lecture

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

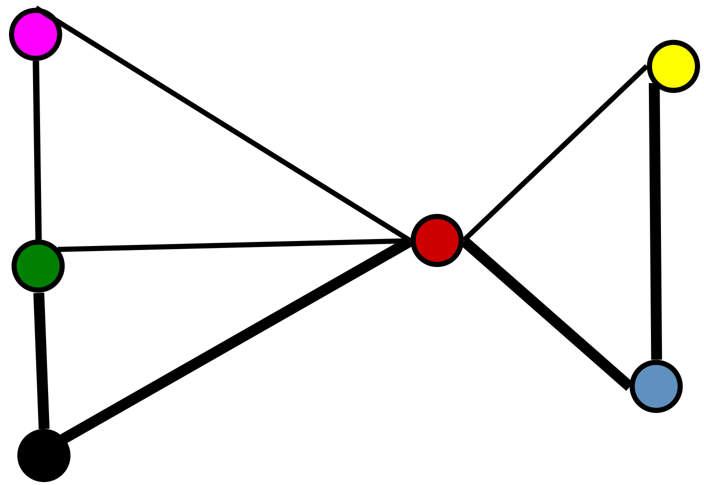
# Paths



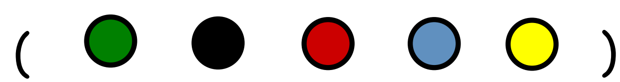
*Path*: sequence of *adjacent* vertices



# Simple Paths



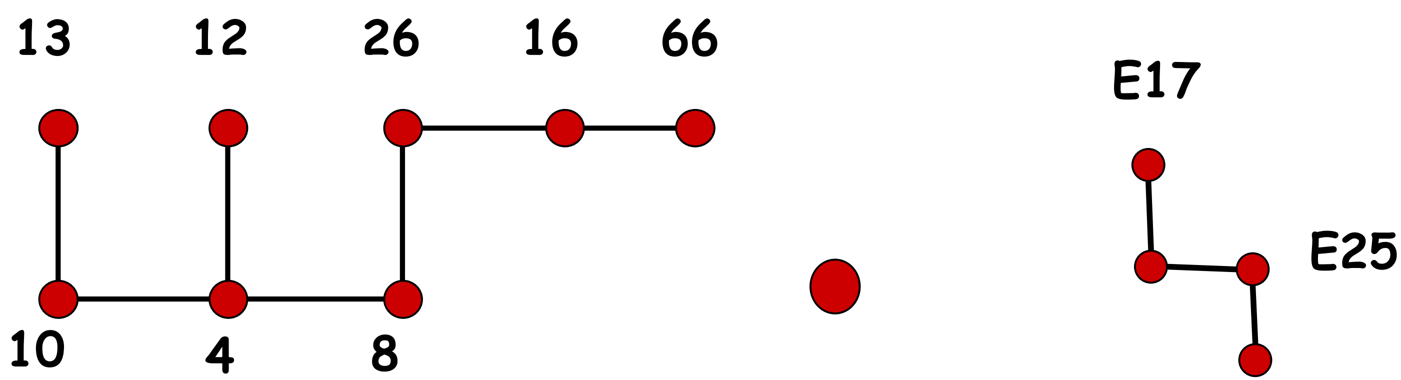
*Simple Path*: all vertices different



# Connectedness

- Vertices *v*, *w* are *connected* if and only if there is a path 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*

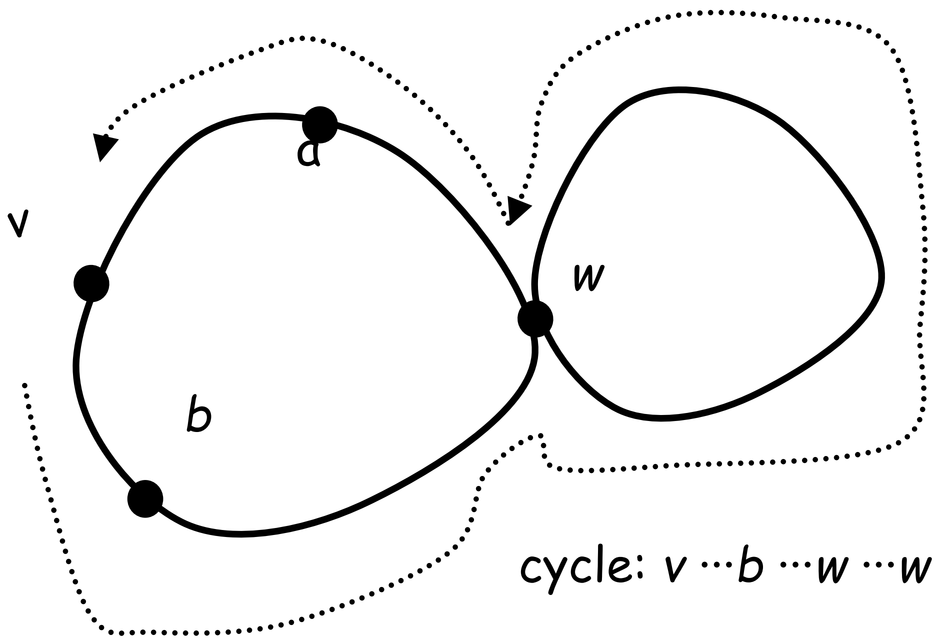



3 connected components

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.



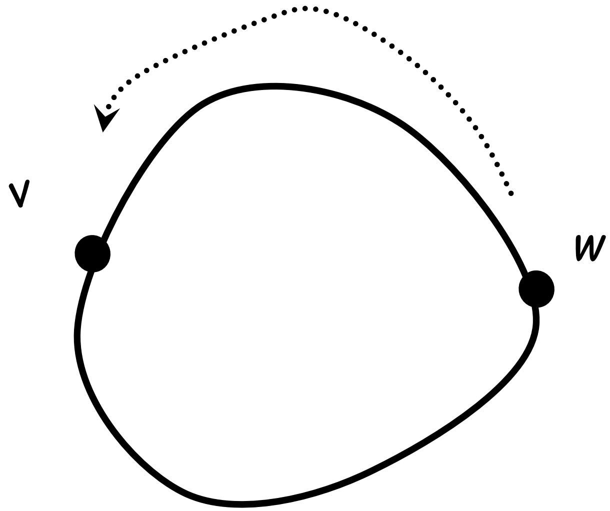

cycle: $v \cdots b \cdots w \cdots w \cdots a \cdots v$

also: $a \cdots v \cdots b \cdots w \cdots w \cdots a$

# Simple Cycles

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

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



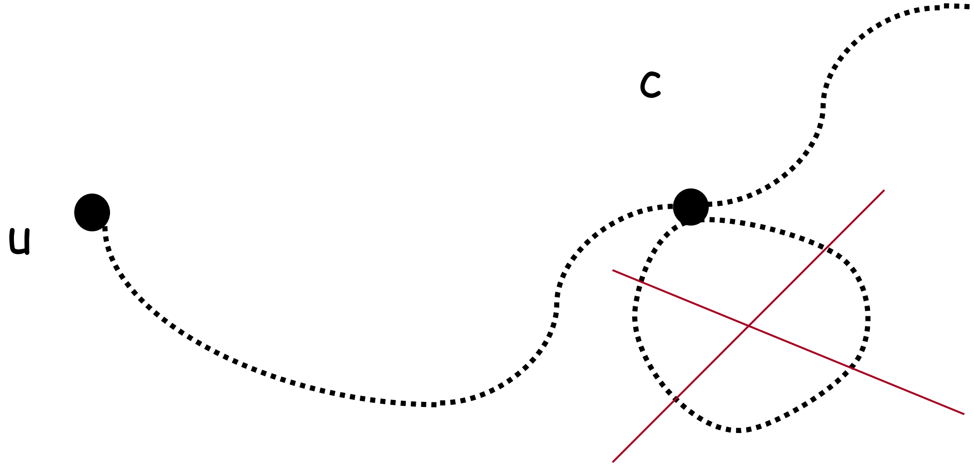

cycle: $v \cdots w \cdots v$ also: $w \cdots v \cdots w$

# 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.

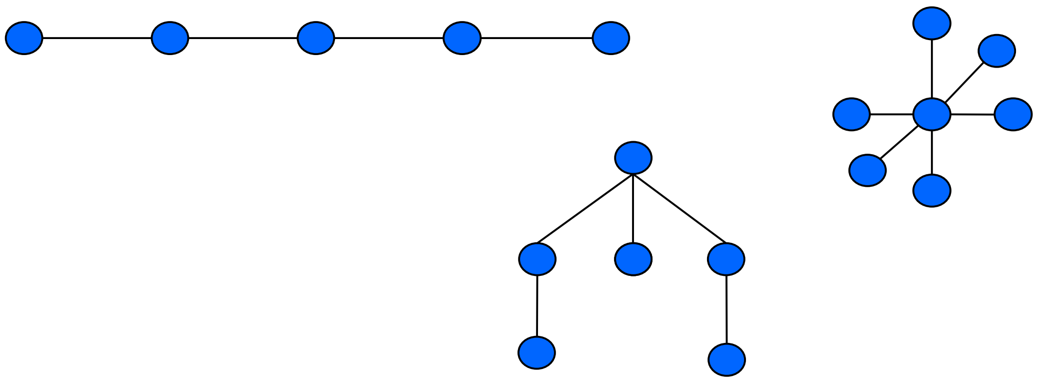


# This Lecture

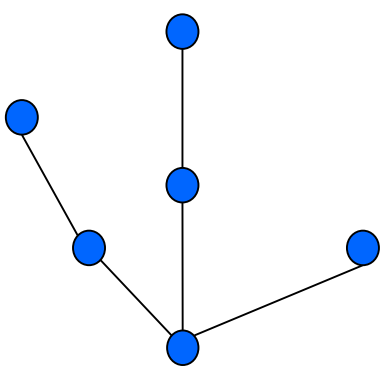
- Seven bridges of Konigsberg
- Graphs, degrees
- Isomorphism
- Path, cycle, connectedness
- Tree
- Eulerian cycle
- Directed graphs

# Tree

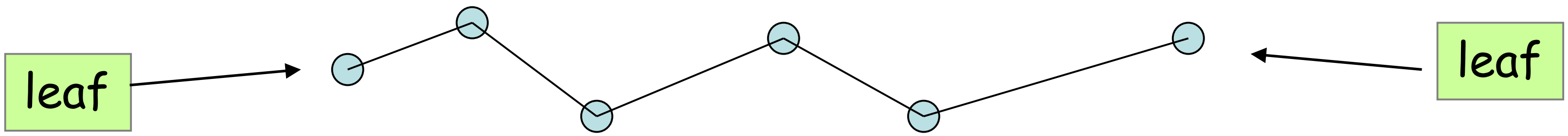
Graphs with no cycles? A forest.



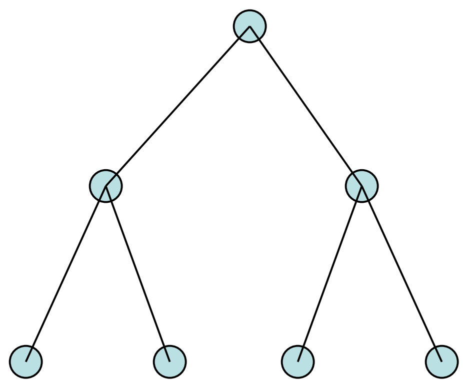
Connected graphs with no cycles? A tree.



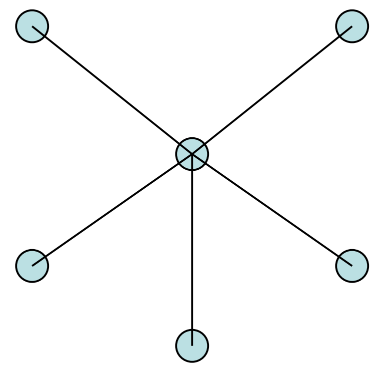
# More Trees




A leaf is a vertex of degree 1.



More leaves.



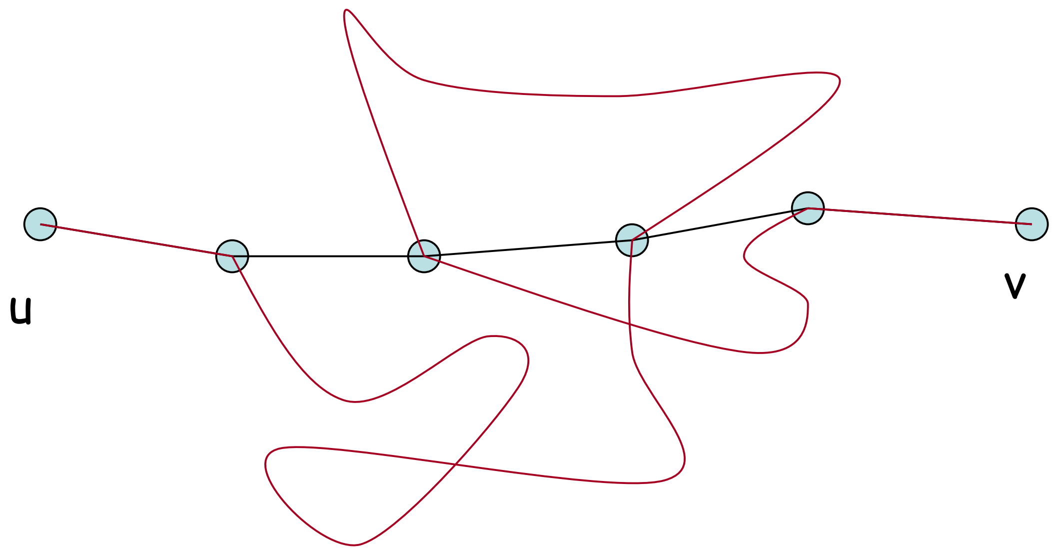
Even more leaves.

# 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.

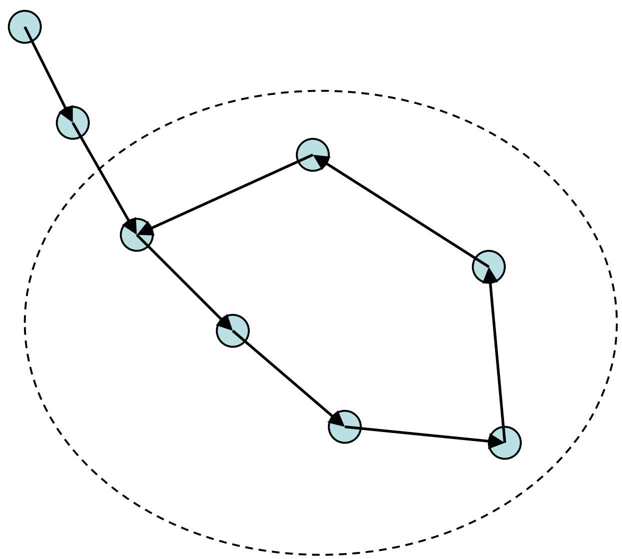
**Claim.** In a tree, there is a unique simple path between every pair of vertices.

# 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 edges as long as possible.

Cannot get stuck,

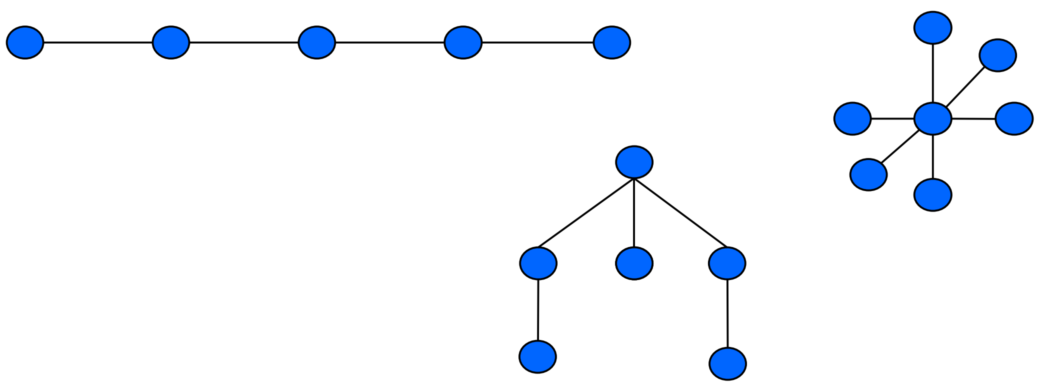
unless there is a cycle.

# 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?



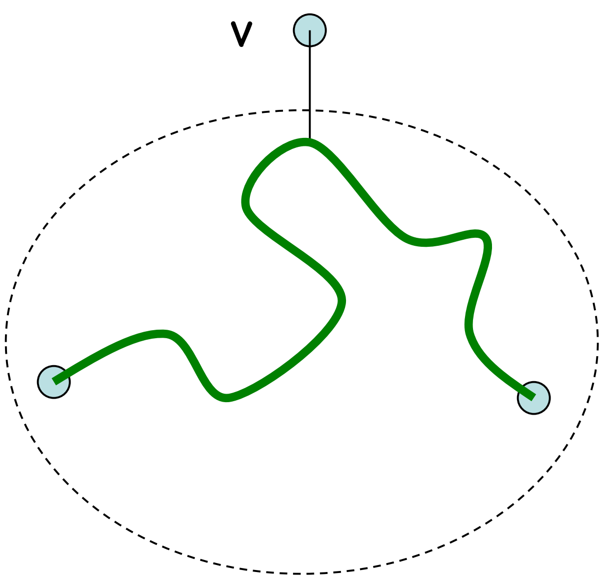
We usually use n to denote the number of vertices,
and use m to denote the number of edges in a graph.

# 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 has 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.

(We have only proved one direction.

The other direction is similar and left as an exercise.)

# This Lecture

- Seven bridges of Konigsberg
- 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.

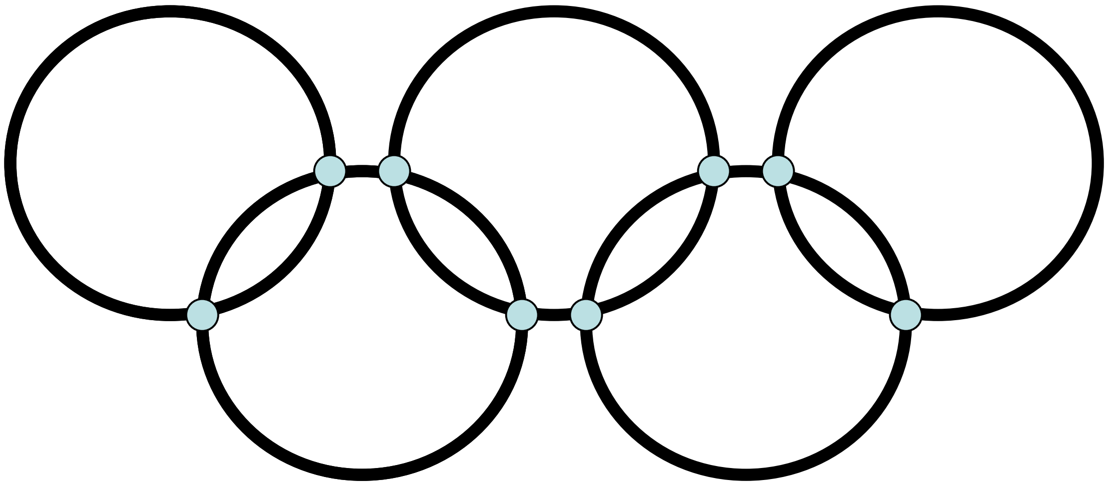
**Euler's theorem:** A connected graph has an Eulerian path if and only if it has zero or two vertices with odd degrees.

Proof by induction. Focus on the case of Eulerian cycle.

# Eulerian Cycle

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

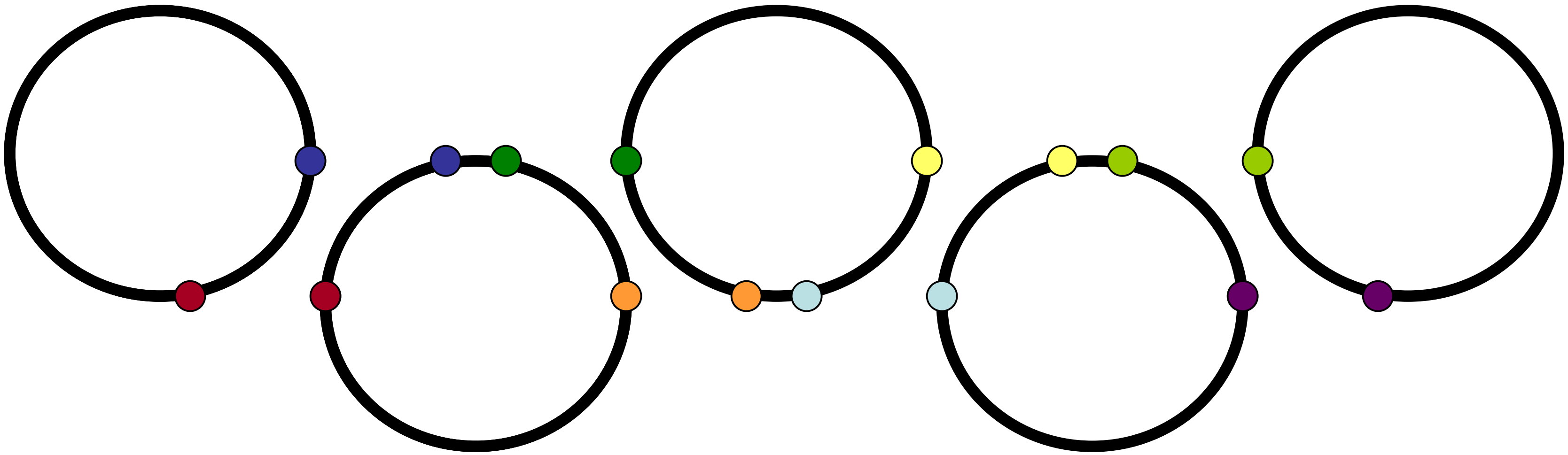
First we find an Eulerian cycle in the below example.



# 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.

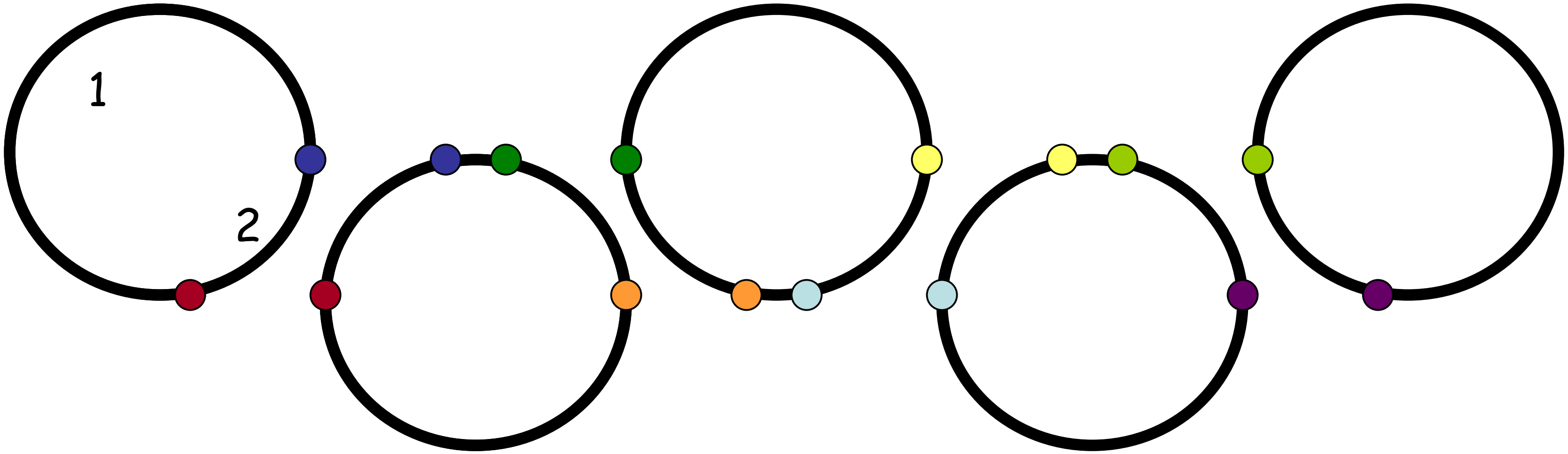


Vertices of the same color represent the same vertices.

# 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.

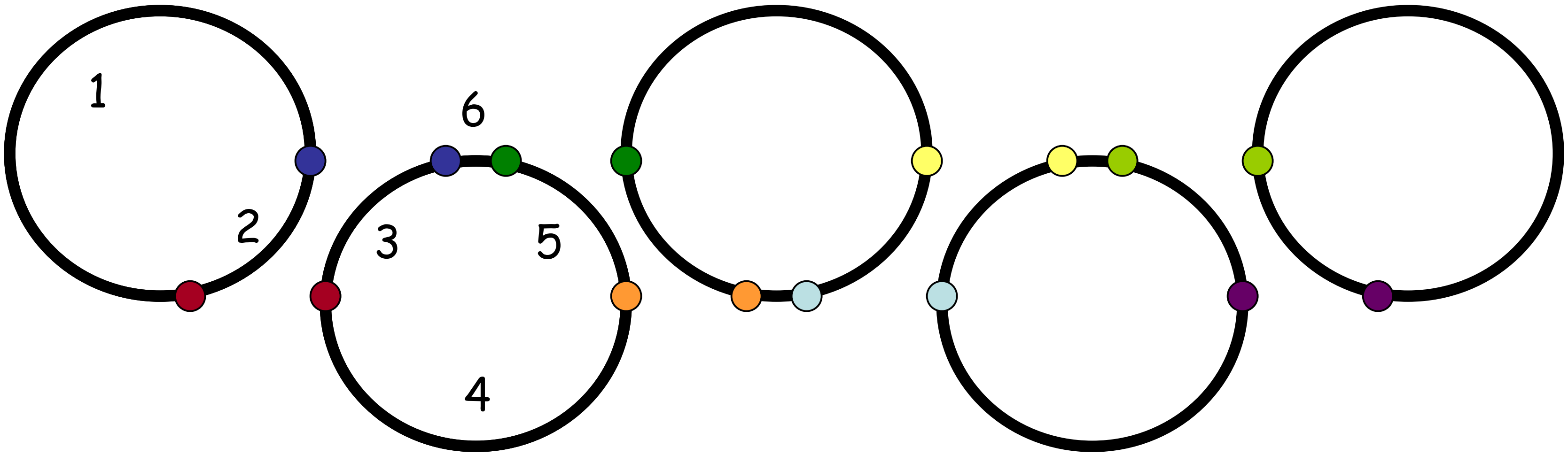



First transverse the first cycle.

# 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 cycle one by one.

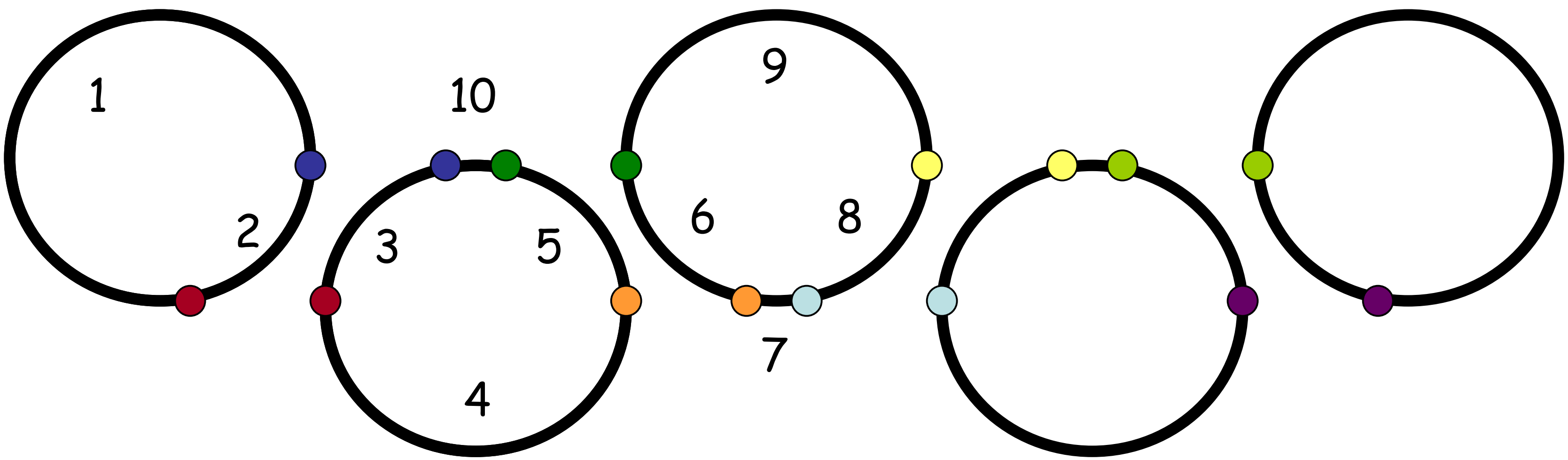



Then transverse the second cycle.

# 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?

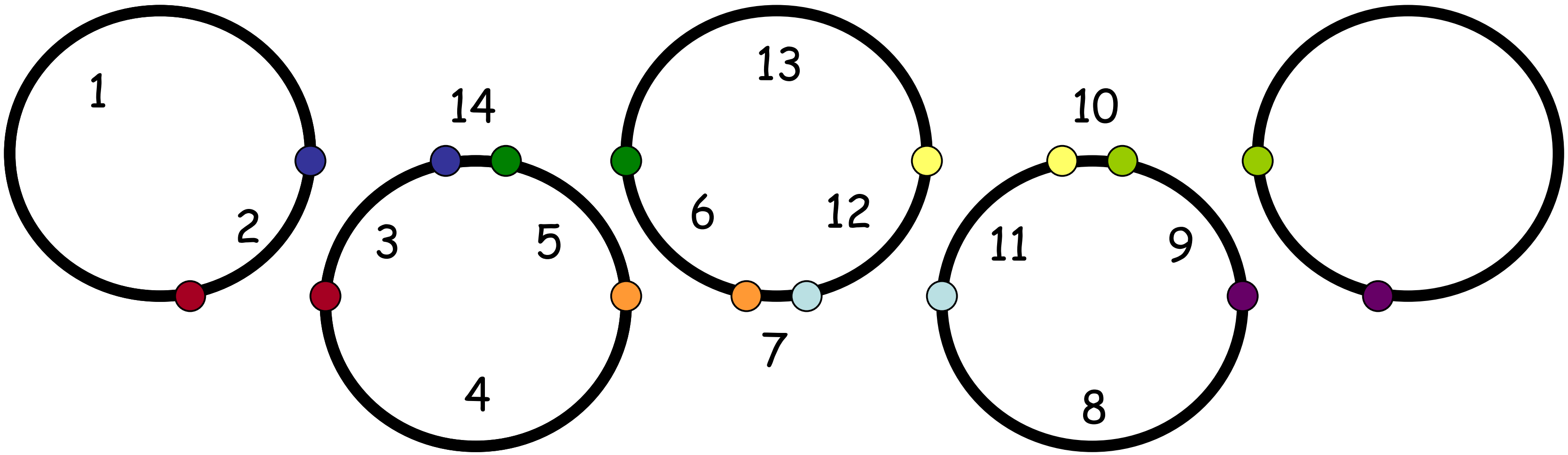



We can "detour" to the third cycle before finishing the second cycle.

# Eulerian Cycle

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

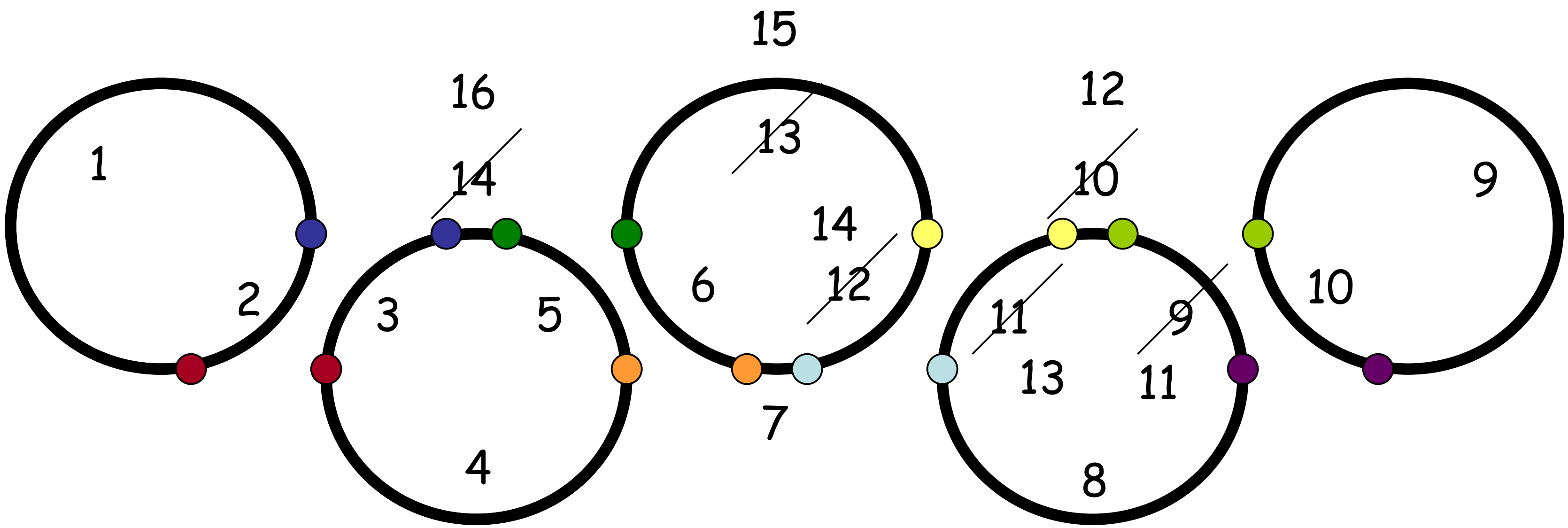
We use the same idea to deal with the fourth cycle




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

# Eulerian Cycle

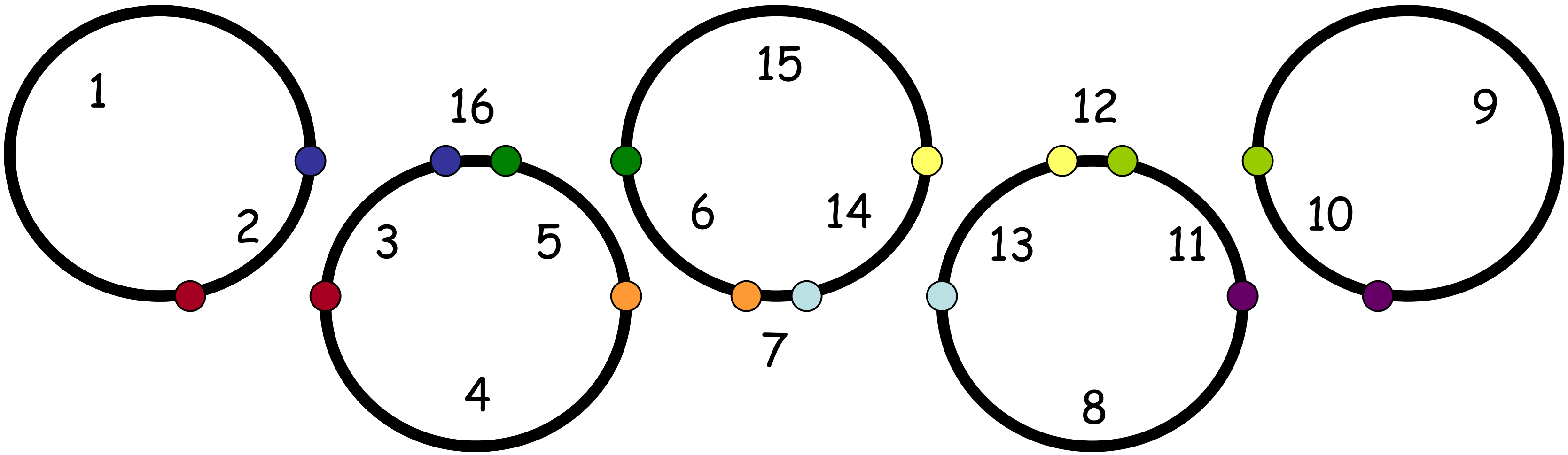
**Euler's theorem:** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.




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

# Eulerian Cycle

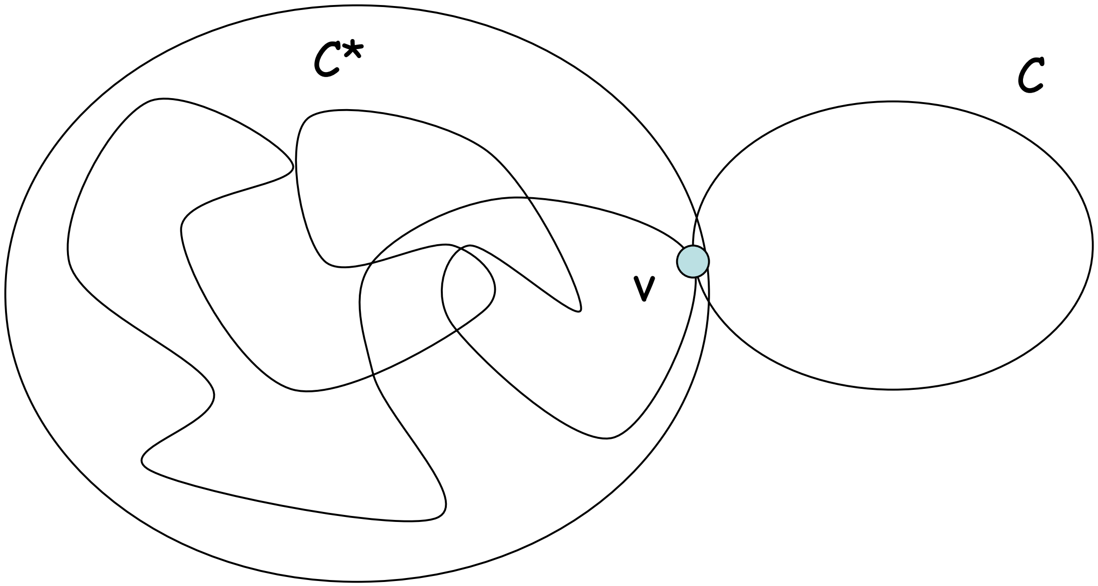
**Euler's theorem:** A connected graph has an Eulerian cycle if and only if every vertex is of even degree.




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 "partial 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.

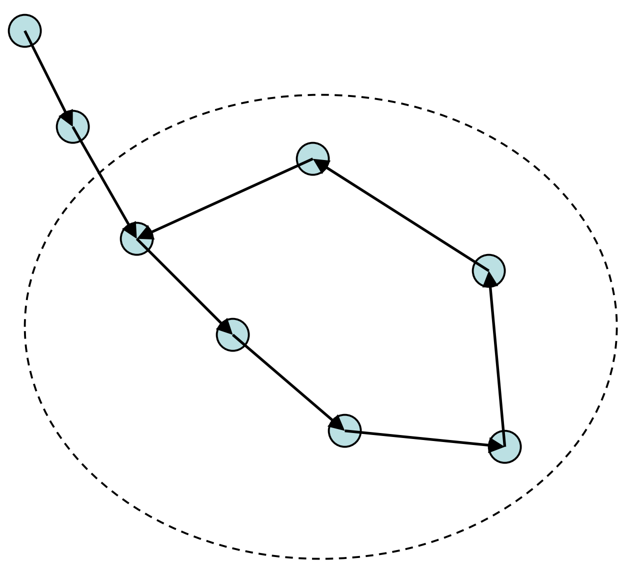
We can prove Euler's theorem if we can prove the following claim.

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

# 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.



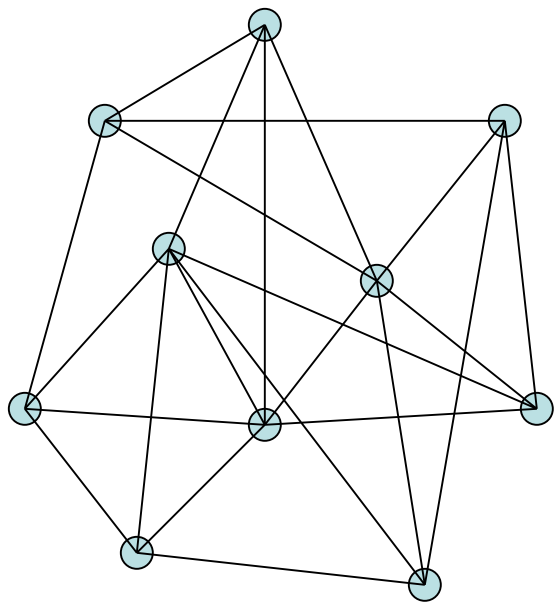
Start from any vertex.

Go to unvisited edges as long as possible.

Cannot get stuck before we find a cycle, because every vertex has degree >= 2.

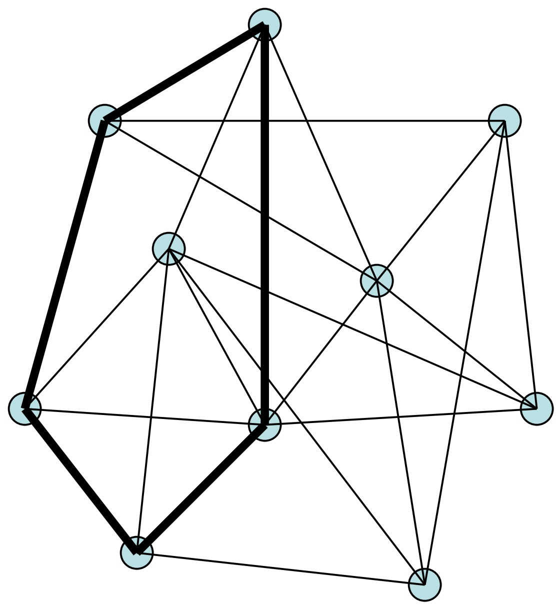
# Partitioned into Simple Cycles

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



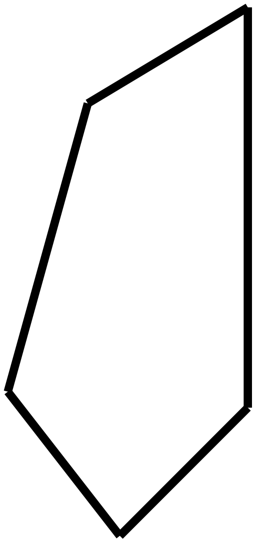
# Partitioned into Simple Cycles

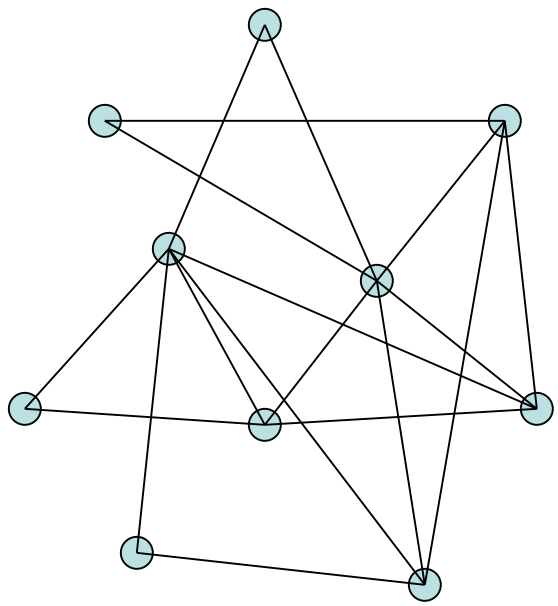
**Claim 2.** If every vertex is of even degree, then the edges can be partitioned into simple cycles.



# Partitioned into Simple Cycles

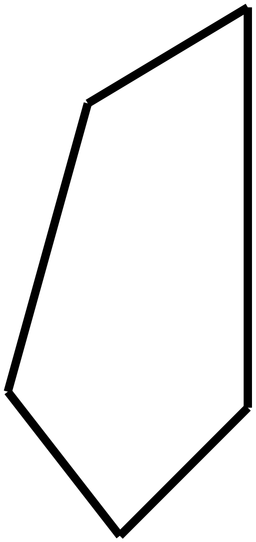
**Claim 2.** If every vertex is of even degree, then the edges can be partitioned into simple cycles.

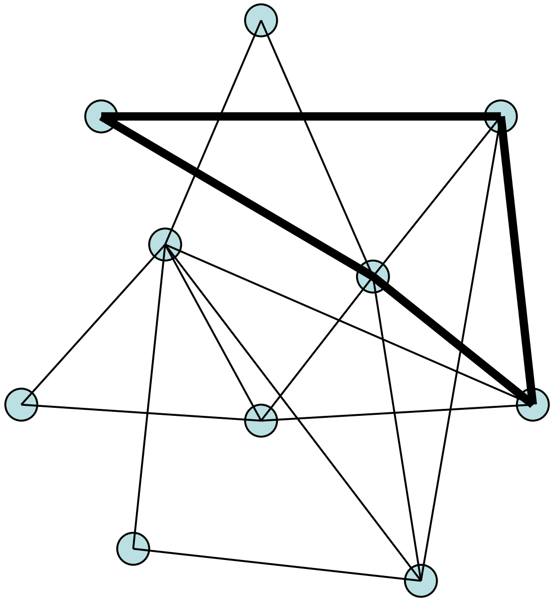




# Partitioned into Simple Cycles

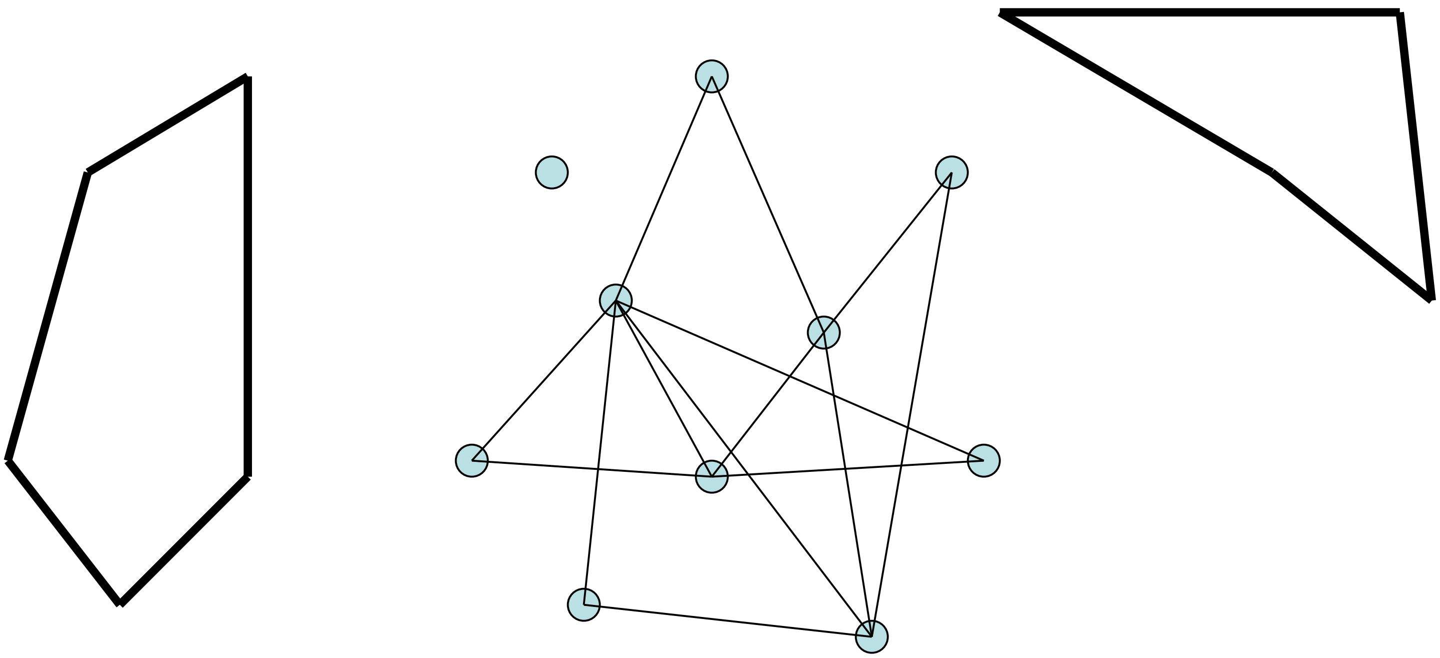
**Claim 2.** If every vertex is of even degree, then the edges can be partitioned into simple cycles.





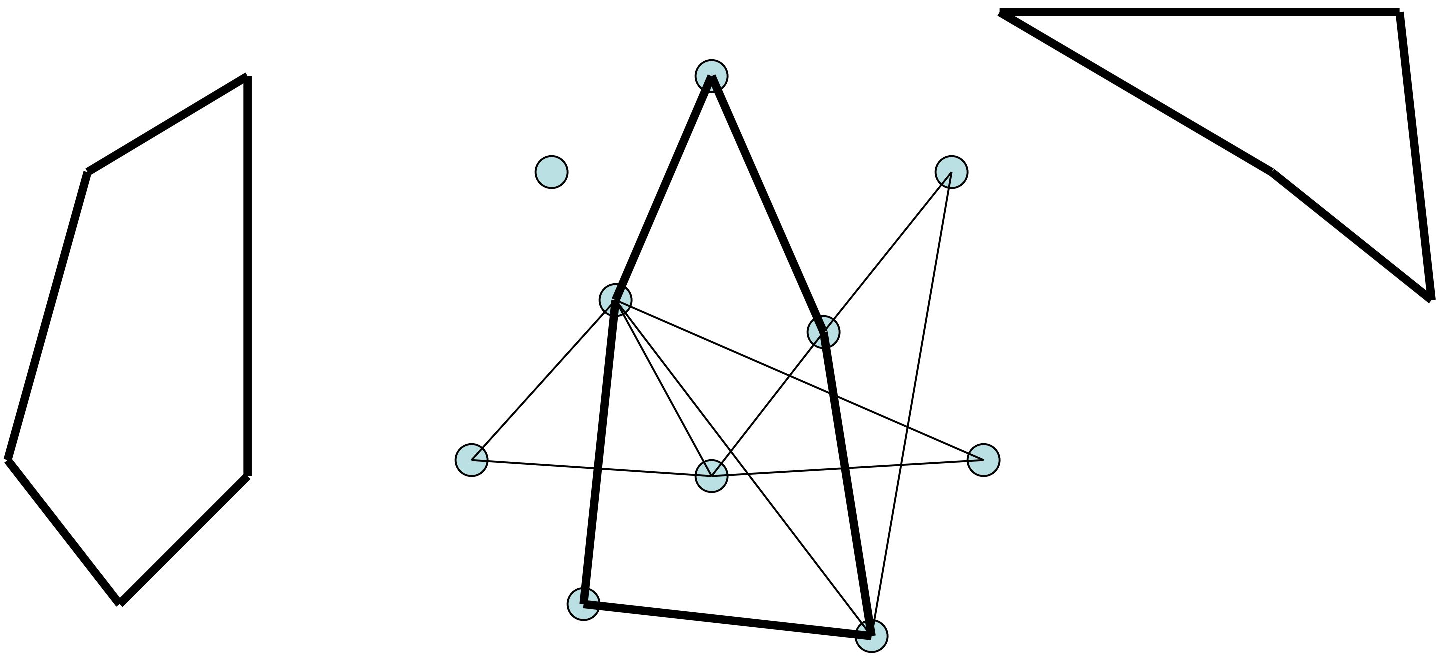
# Partitioned into Simple Cycles

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



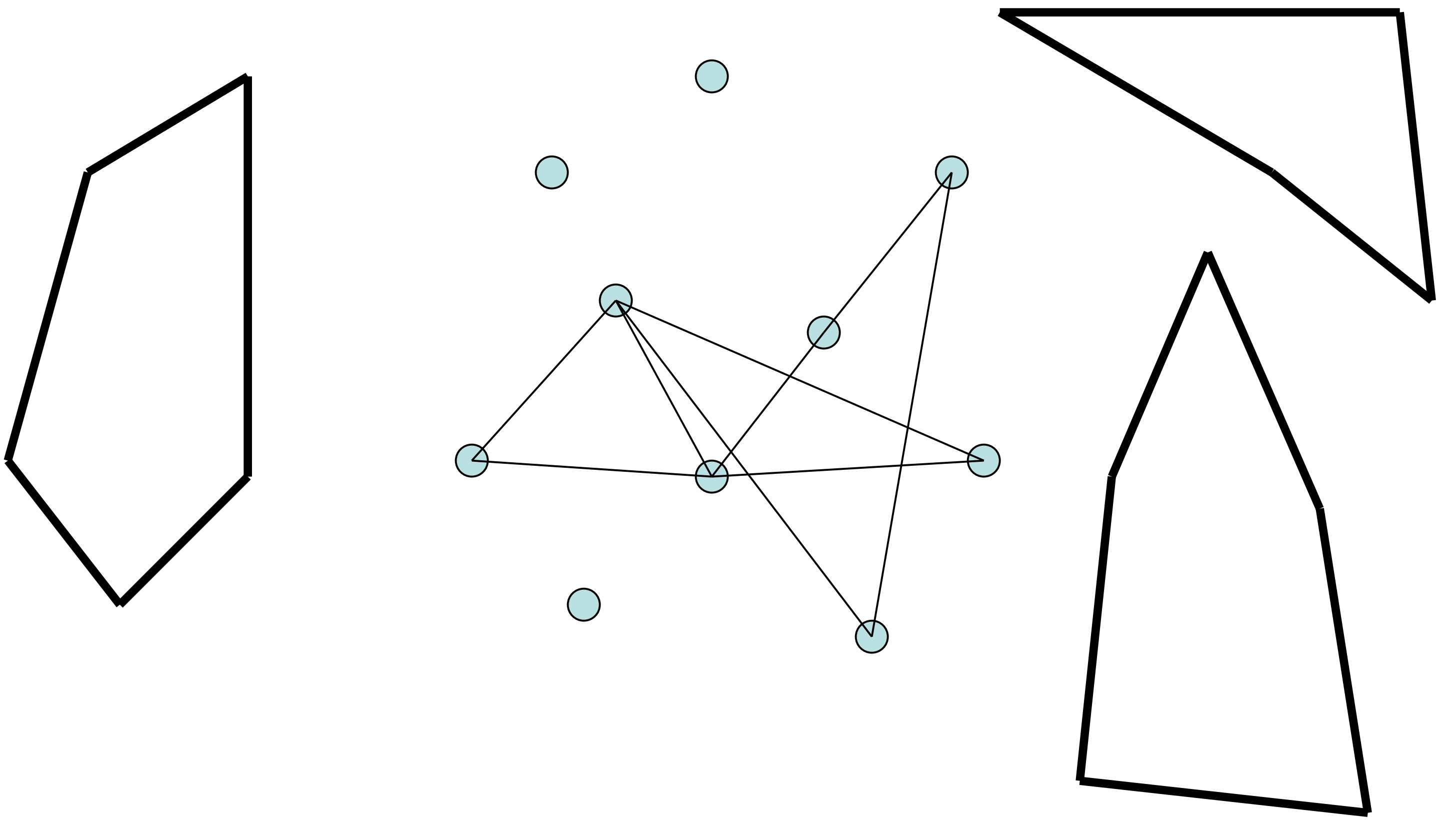
# Partitioned into Simple Cycles

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



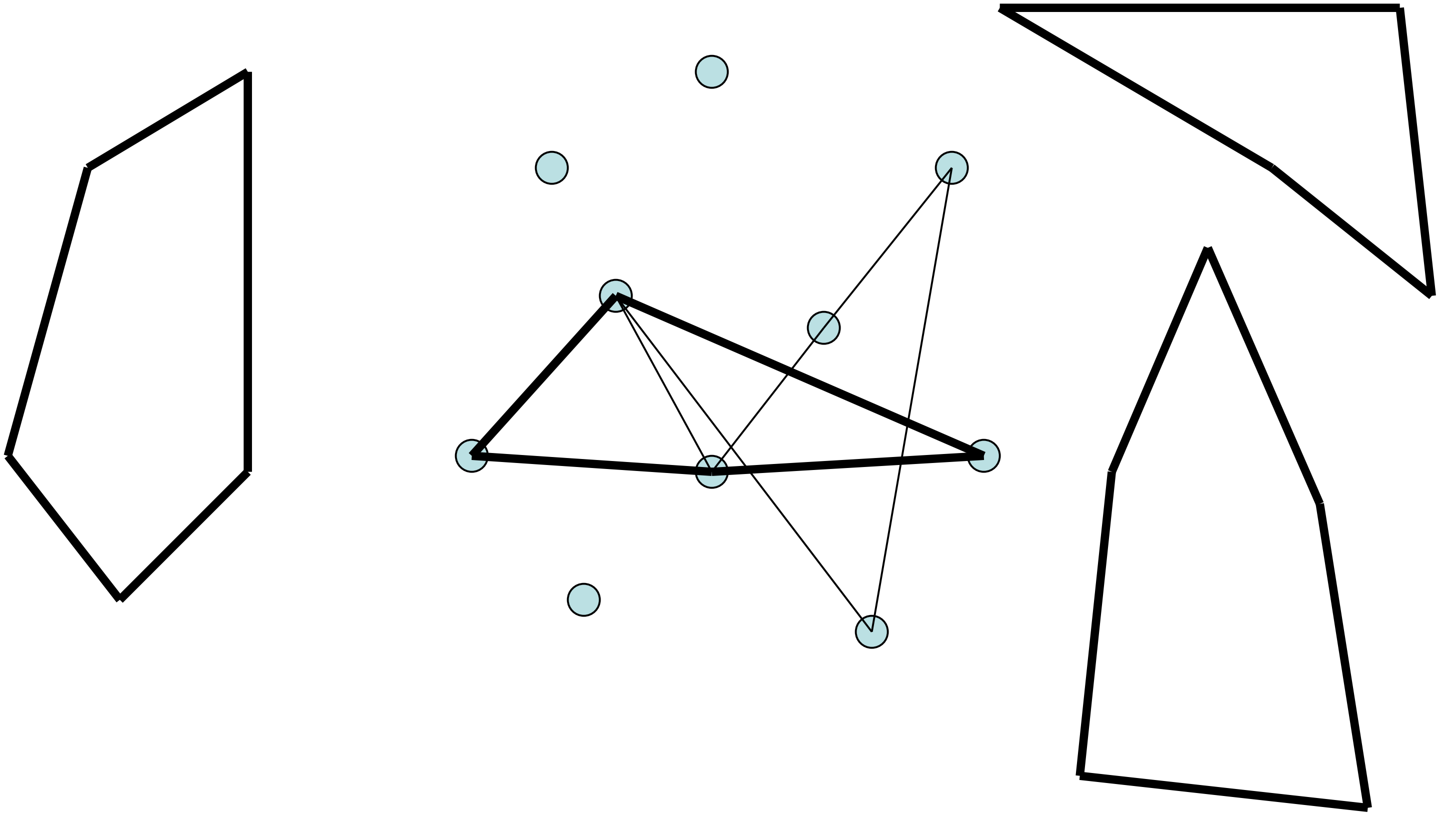
# Partitioned into Simple Cycles

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



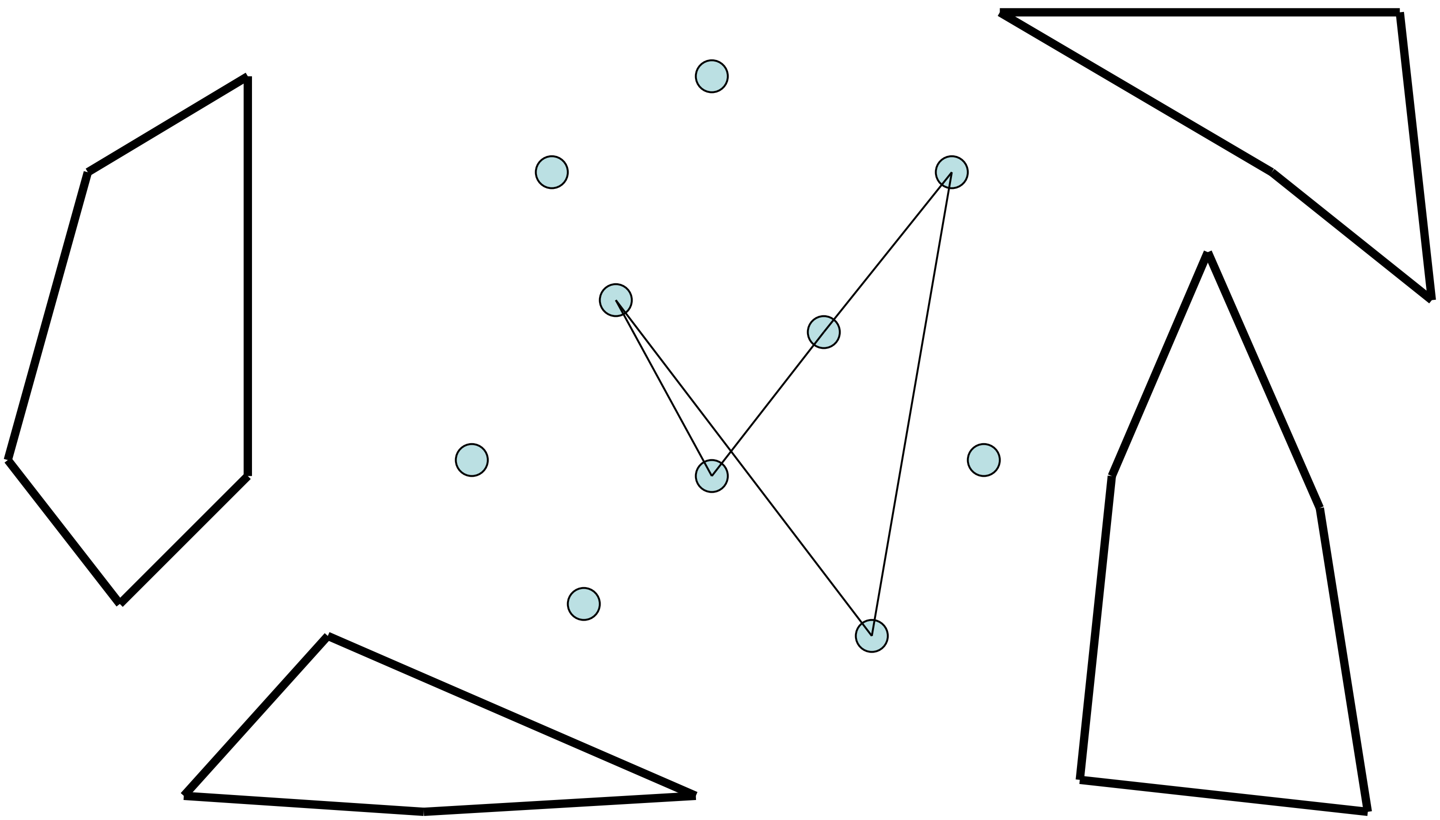
# Partitioned into Simple Cycles

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



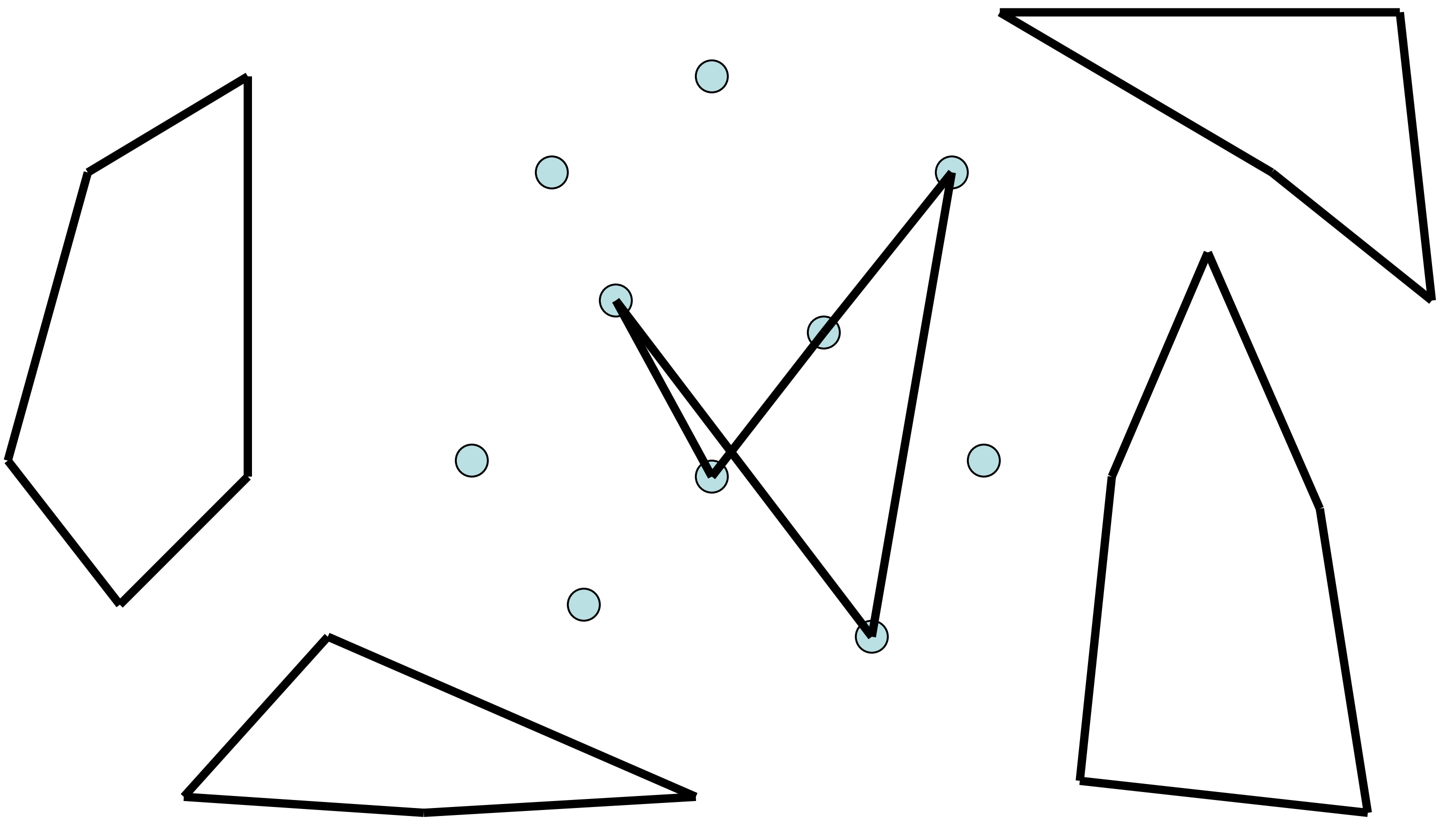
# Partitioned into Simple Cycles

**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 $G$.

By induction, $G'$ can be partitioned into simple cycles $C_1, C_2, \ldots, C_k$.

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

# 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 proved Euler's theorem.

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

# This Lecture

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

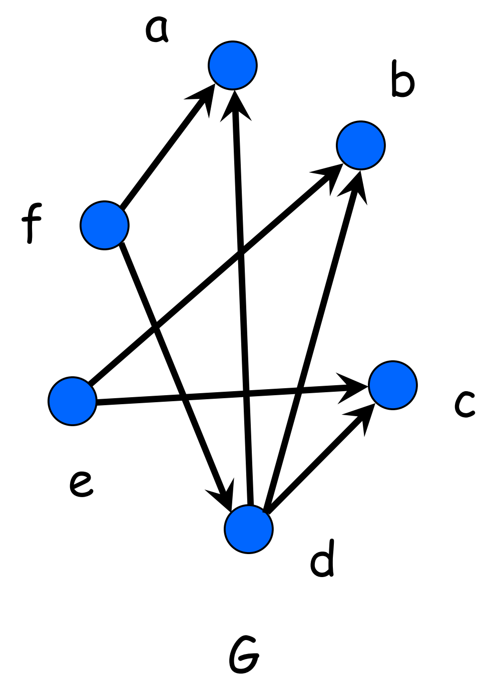
# Directed Graphs

A directed graph G=(V,A) consists of:

A set of vertices, V

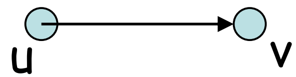
A set of *directed* edges (arcs), A

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



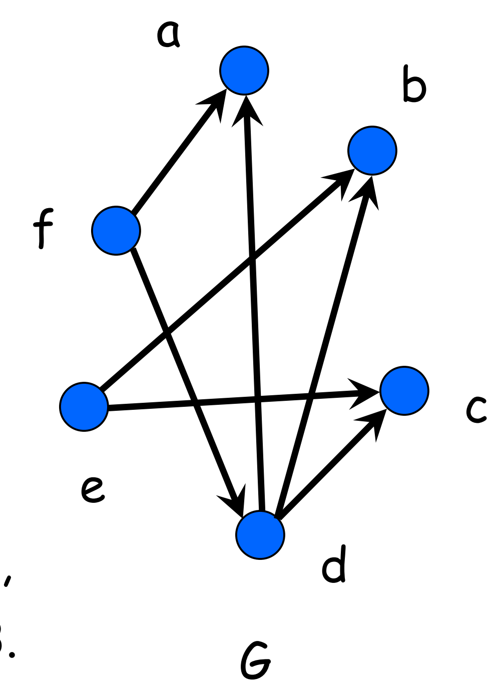

For an arc uv, we say u is the tail of the arc and v is the head of the arc.

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



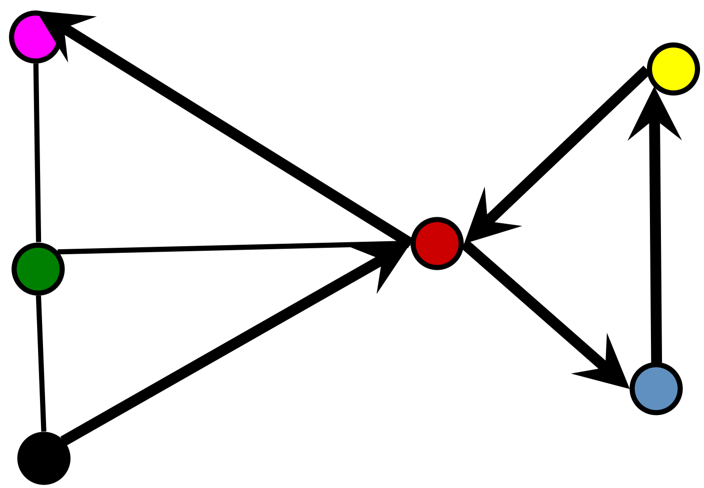
# In-Degrees and Out-Degrees

The **out-degree** of a vertex v is the number of arcs with v being tail; similarly, the **in-degree** of a vertex v is the number of arcs with v being head.




E.g. the indegree of a is 2 and the outdegree of a is 0,
the indegree of d is 1 and the outdegree of d is 3.

# Directed Paths



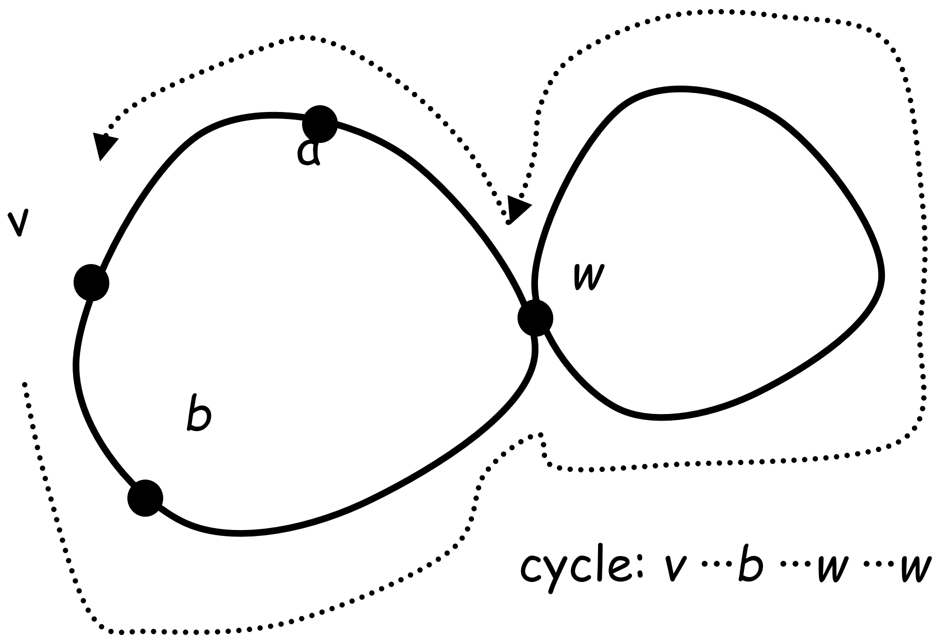
Informally, a directed path is a path that follows directions.

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

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

# Directed Cycles

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




cycle: $v \cdots b \cdots w \cdots w \cdots a \cdots v$

also: $a \cdots v \cdots b \cdots w \cdots w \cdots a$

A *simple directed cycle* is a directed cycle with no repeated vertices except the last.

# 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 will be your homework problem.