Lecture #12: Graphs

Colin Tan
(Slides Credit: A/P Aaron Tan, CS1231S)

10.1 Graphs: Definitions and Basic Properties

- Introduction, Basic Terminology
- Special Graphs
- The Concept of Degree

10.2 Trails, Paths, and Circuits

- Definitions
- Connectedness
- Euler Circuits and Hamiltonian Circuits

10.3 Matrix Representations of Graphs

- Matrices and Directed Graphs; Matrices and Undirected Graphs
- Matrix Multiplication
- Counting Walks of Length *N*

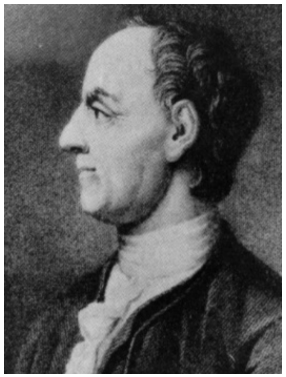
10.4 Isomorphisms of Graphs/Planar Graphs

- Definition of Graph Isomorphism
- Planar Graphs and Euler's Formula

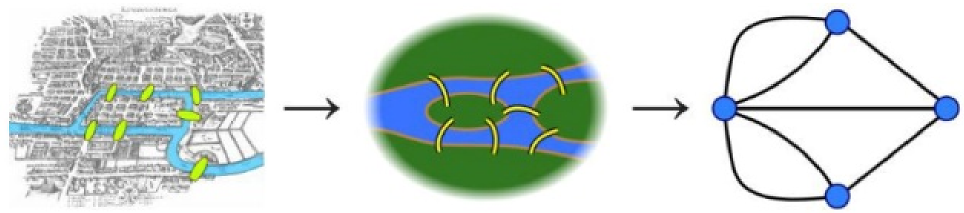
Reference: Epp's Chapter 10 Graphs and Trees

Graphs: Introduction

“The origins of graph theory are humble, even frivolous.” ~*Norman L. Biggs*



The Father of Graph Theory
Leonhard Euler (1707-1783)



Credit: Wikipedia

The 7 bridges of Königsberg

Euler’s formula: F + V = E + 2

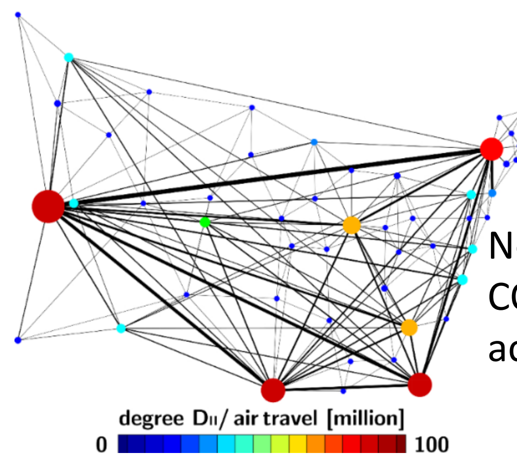
Knight’s Tour Problem

Travelling Salesman Problem

Applications:

- CS: hardware, data structures, image processing, data mining, network design, etc.
- GPS to find shortest path
- Ranking hyperlinks in search engines
- Social network analysis
- DNA sequence
- …

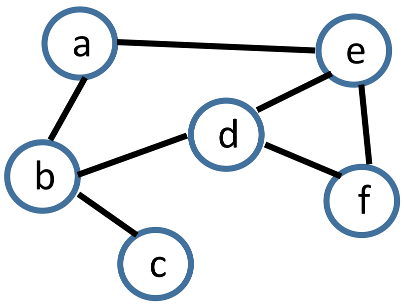
Graph as an excellent modelling tool…



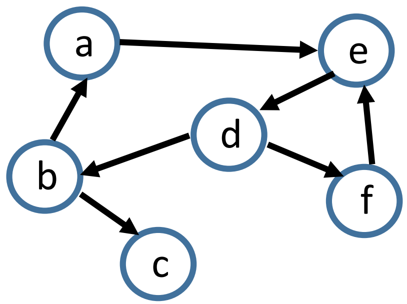
Network model of COVID-19 spreading across the United States

Graphs are mathematical structures used to model pairwise relations between objects.

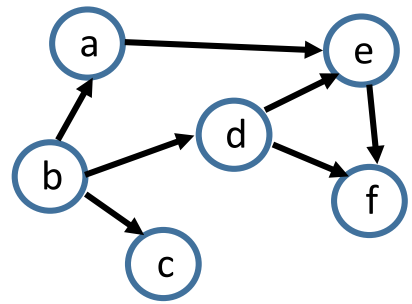
Types of graphs (informal intro):



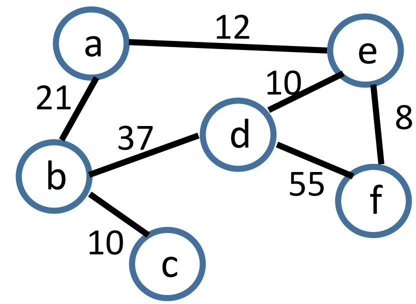
Undirected graph



Directed graph (Digraph)



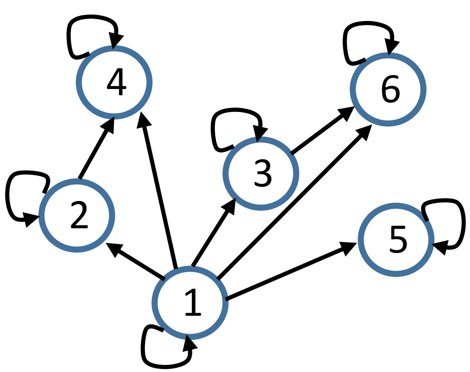
Directed acyclic graph (DAG)



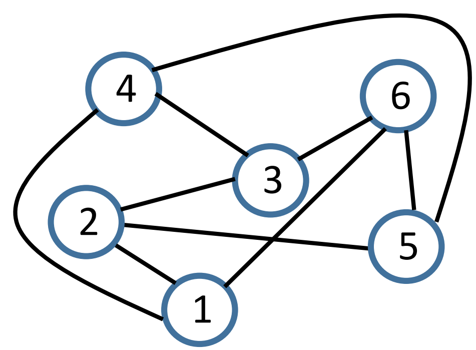
Weighted graph

Bet you have seen graphs in CS1231S before!

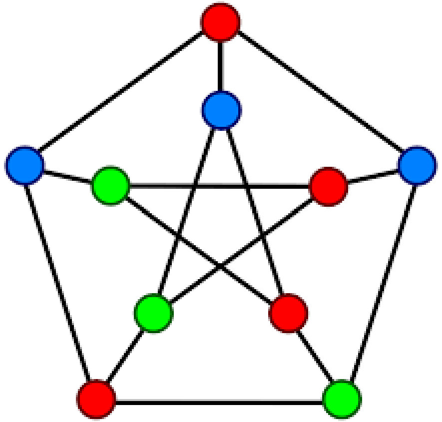
Relation R on a set $A = \{1,2,3,4,5,6\}$, s.t. $\forall x, y \in A$, $x\ R\ y$ iff $x \mid y$.



Relation T on a set $A = \{1,2,3,4,5,6\}$, s.t. $\forall x, y \in A$, $x\ T\ y$ iff $(x + y) = 2k + 1$ for some $k \in \mathbb{Z}$.

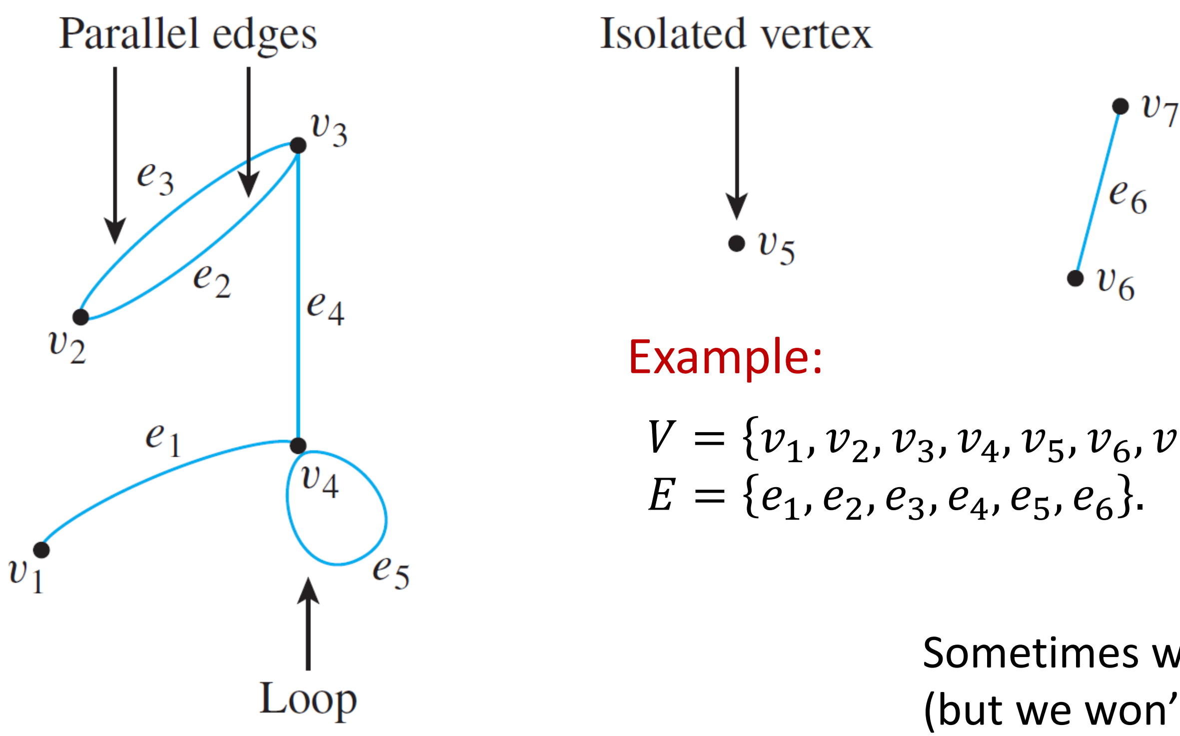


10.1 Graphs: Definitions and Basic Properties



Graphs: Definitions and Basic Properties

- An undirected graph is denoted by $G = (V, E)$ where
  - $V = \{v_1, v_2, \cdots, v_n\}$ is the set of vertices (or nodes) in G; and
  - $E = \{e_1, e_2, \cdots, e_k\}$ is the set of (undirected) edges in G.
  - An (undirected) edge e connecting v_i and v_j is denoted as $e = \{v_i, v_j\}$.



Example:

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\};$$
$$E = \{e_1, e_2, e_3, e_4, e_5, e_6\}.$$

$$e_1 = \{v_1, v_4\};$$
$$e_2 = e_3 = \{v_2, v_3\};$$
$$e_4 = \{v_3, v_4\};$$
$$e_5 = \{v_4, v_4\};$$
$$e_6 = \{v_6, v_7\}.$$

Sometimes we write $e_5 = \{v_4\}$
(but we won't use this.)

Graphs: Definitions and Basic Properties

Definition: Undirected Graph

An undirected **graph** G consists of 2 finite sets: a nonempty set V of **vertices** and a set E of **edges**, where each (undirected) edge is associated with a set consisting of either one or two vertices called its **endpoints**.

An edge is said to **connect** its endpoints; two vertices that are connected by an edge are called **adjacent vertices**; and a vertex that is an endpoint of a loop is said to be **adjacent to itself**.

An edge is said to be **incident on** each of its endpoints, and two edges incident on the same endpoint are called **adjacent edges**.

We write $e = \{v, w\}$ for an undirected edge e incident on vertices v and w.

Graphs: Definitions and Basic Properties

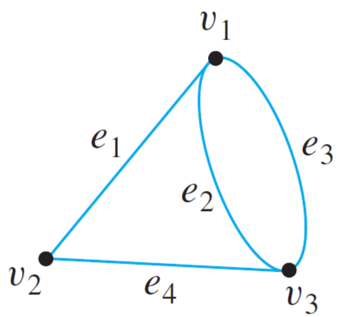
Definition: Directed Graph

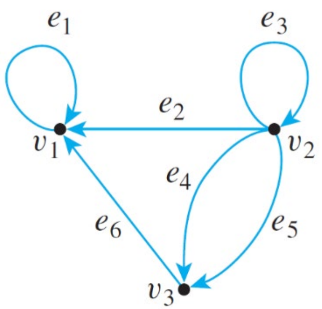
A **directed graph**, or **digraph**, *G*, consists of 2 finite sets: a nonempty set *V* of **vertices** and a set *E* of **directed edges**, where each (directed) edge is associated with an ordered pair of vertices called its **endpoints**.

We write $e = (v, w)$ for a directed edge e from vertex v to vertex w.

Undirected graph

$e_2 = \{v_1, v_3\}$



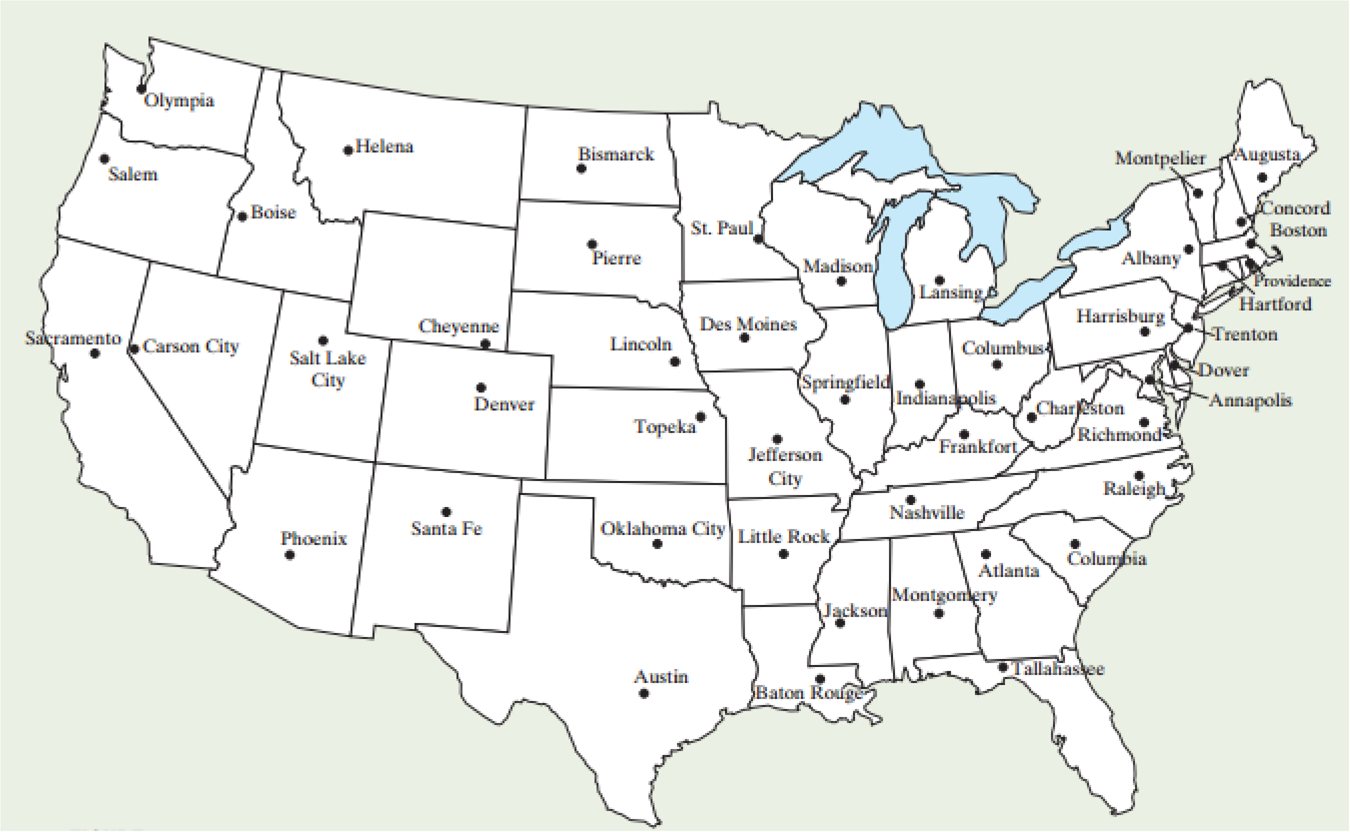


Directed graph

$e_2 = (v_2, v_1)$

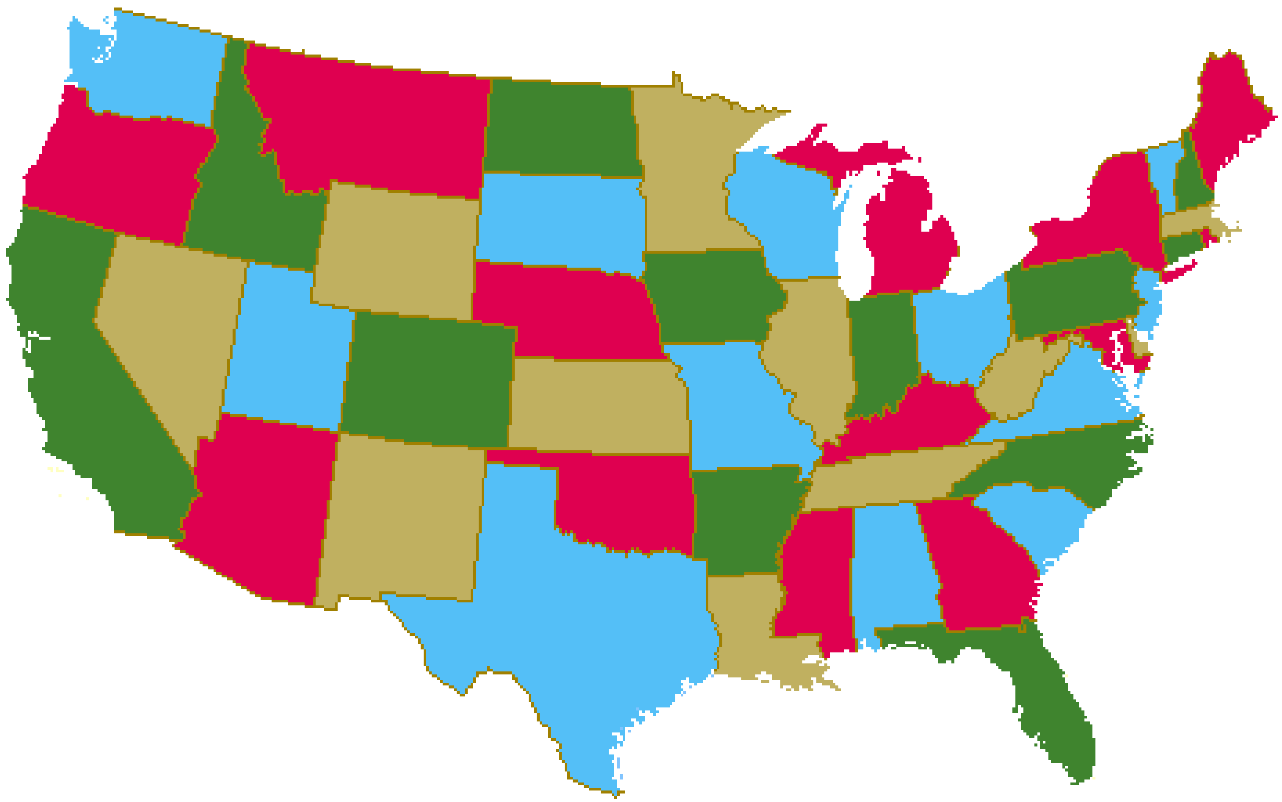
Map Colouring

Shall we add some colours to this map of the United States?



Map Colouring

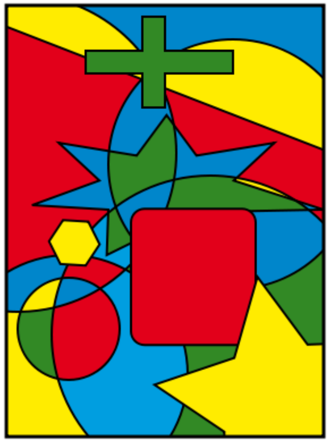
Shall we add some colours to this map of the United States?



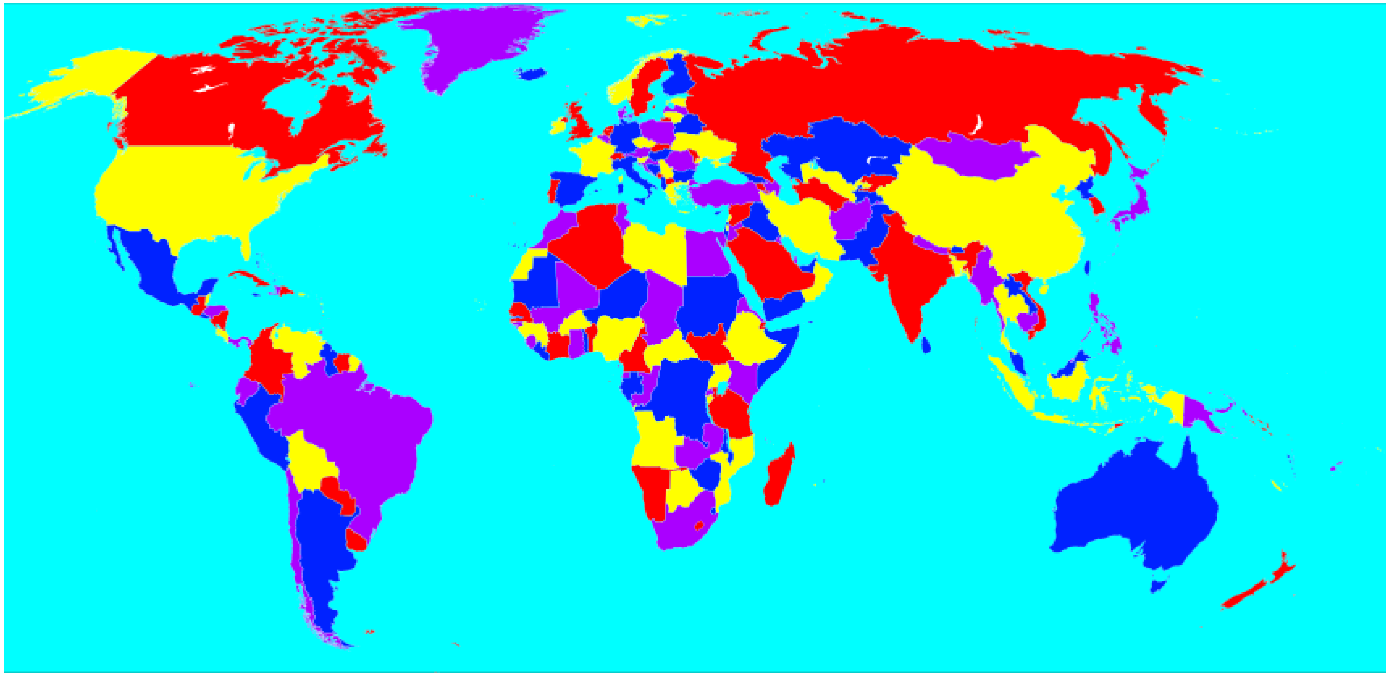
Map Colouring

- Four-Colour Conjecture
  - Proposed by Guthrie in 1852, who conjectured that...
  - Four colours are sufficient to colour any map in a plane, such that regions that share a common boundary do not share the same colour.
  - Many false proofs since then.
  - Finally proved by Appel and Haken in 1977, with the help of computer.
  - Robertson et al. provided another proof in 1996.

Map Colouring



Example of a 4-coloured map.



World map with 4 colours.

- But this is a map, not a graph!
- However, we can model it as a graph.
- But what is a graph?

Modelling Graph Problems

Map Colouring Problem

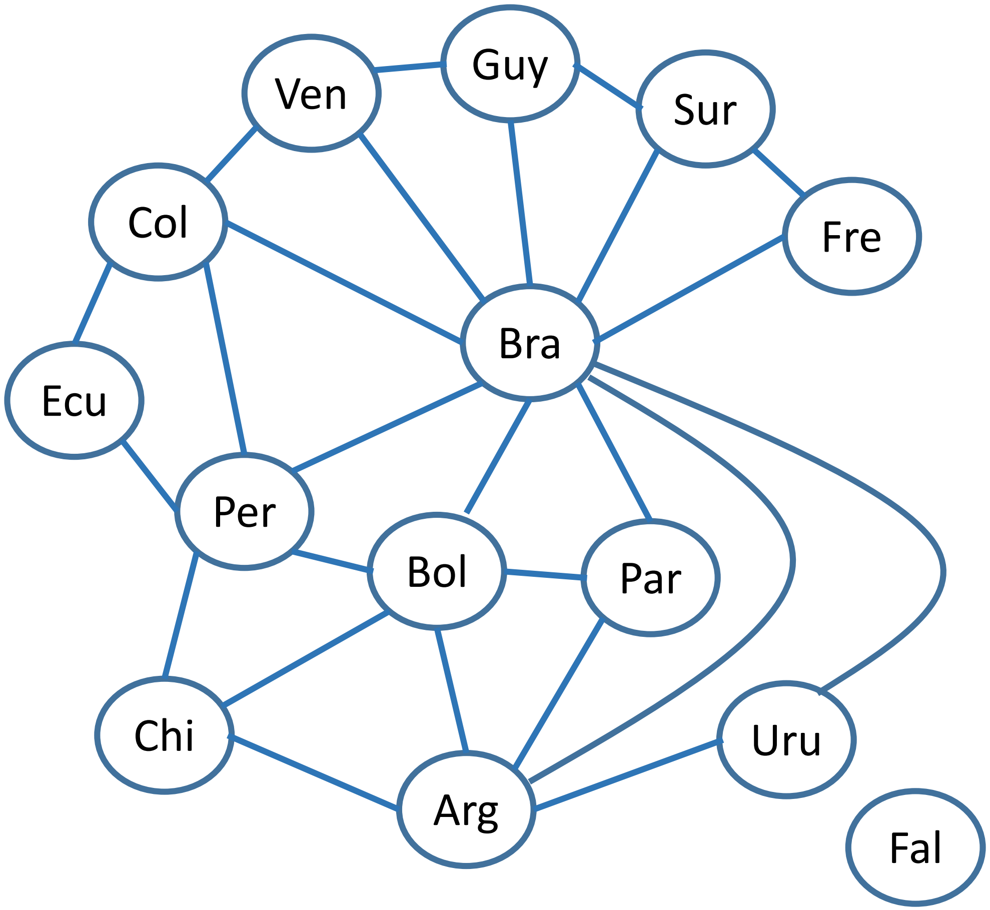
Solve it as a graph problem.

Draw a graph in which the vertices represent the states, with every edge joining two vertices represents the states sharing a common border.

Such two vertices cannot be coloured with the same colour.

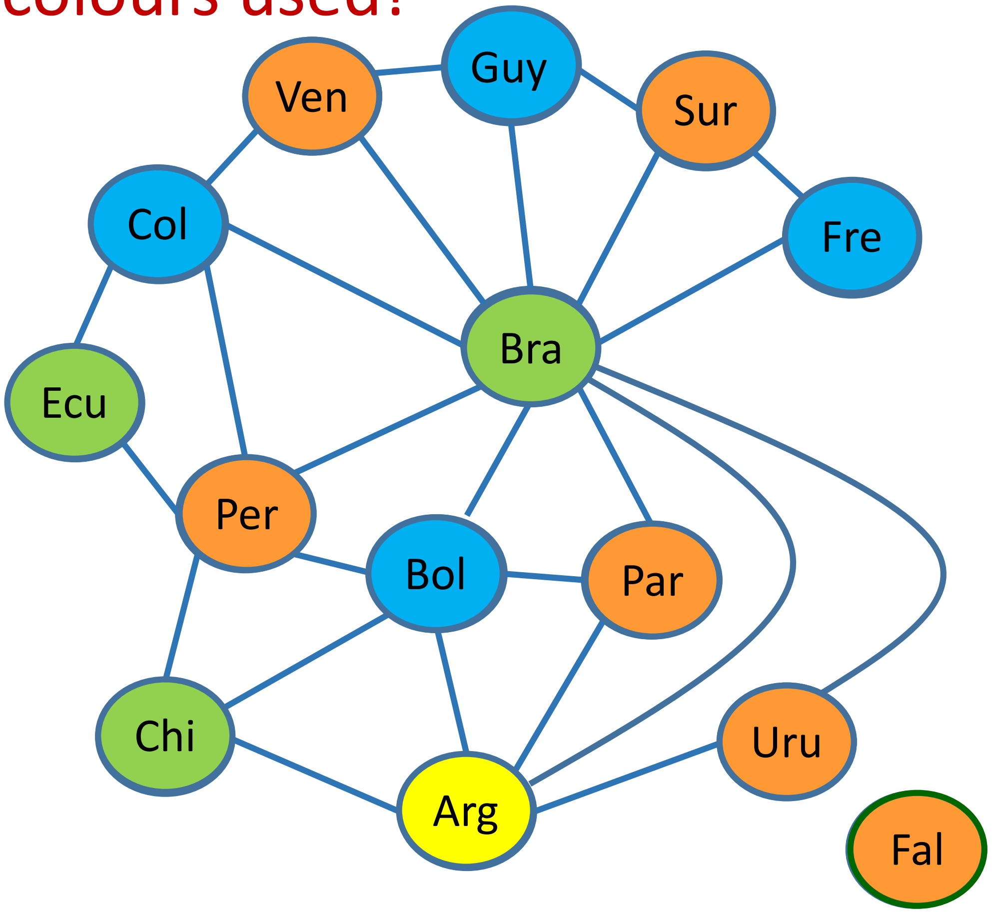
A **vertex colouring** of a graph is an assignment of colours to vertices so that no two adjacent vertices have the same colour.

Modelling Graph Problems



Modelling Graph Problems

4 colours used!



Wedding Planner

You are your best friend's wedding planner and you need to plan the seating arrangement for his 16 guests attending his wedding dinner. However, some of the guests cannot get along with some others.

- *A* doesn't get along with *F*, *G* or *H*.
- *B* doesn't get along with *C*, *D* or *H*.
- *C* doesn't get along with *B*, *D*, *E*, *G* or *H*.
- *D* doesn't get along with *B*, *C* or *E*.
- *E* doesn't get along with *C*, *D*, *F*, or *G*.
- *F* doesn't get along with *A*, *E* or *G*.
- *G* doesn't get along with *A*, *C*, *E* or *F*.
- *H* doesn't get along with *A*, *B* or *C*.

You don't want to put guests who cannot get along with each other at the same table!

How many tables do you need?

Acknowledgement: http://www.math.uri.edu/~eaton/0131873814_MEb.pdf

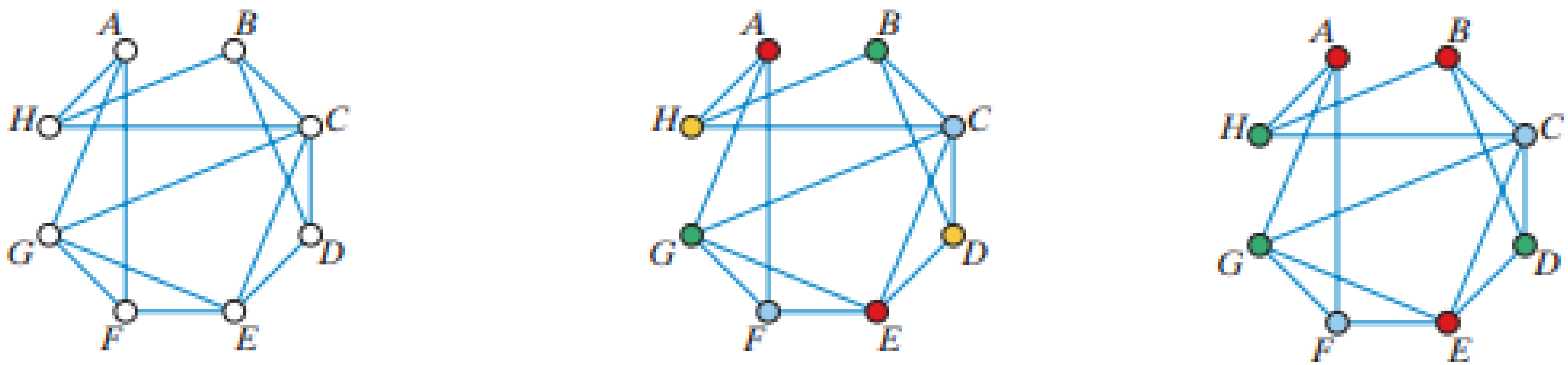
Wedding Planner

- *A* doesn't get along with *F*, *G* or *H*.
- *B* doesn't get along with *C*, *D* or *H*.
- *C* doesn't get along with *B*, *D*, *E*, *G* or *H*.
- *D* doesn't get along with *B*, *C* or *E*.
- *E* doesn't get along with *C*, *D*, *F*, or *G*.
- *F* doesn't get along with *A*, *E* or *G*.
- *G* doesn't get along with *A*, *C*, *E* or *F*.
- *H* doesn't get along with *A*, *B* or *C*.

Graph with vertices representing the guests, and an edge is drawn between two guests who don't get along.

Vertex colouring problem.
4 colours (4 tables)?

3 colours
(3 tables)!



(a) (b) (c)

Acknowledgement: http://www.math.uri.edu/~eaton/0131873814_MEb.pdf

Other Vertex Colouring Problems

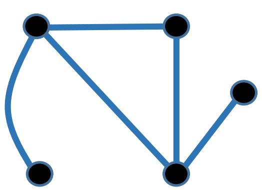
	If the vertices represent…	And two vertices are adjacent if ….	Then a **vertex colouring** can be used to…
1.	classes,	the corresponding classes have students in common,	schedule classes.
2.	radio stations,	the stations are close enough to interfere with each other,	assign non-interfering frequencies to the stations.
3.	traffic signals at an intersection,	the corresponding signals cannot be green at the same time,	designate sets of signals that can be green at the same time.

Simple Graphs

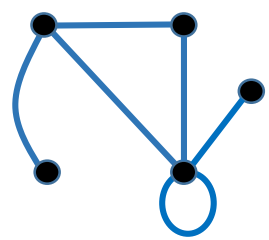
Definition: Simple Graph

A **simple graph** is an undirected graph that does not have any loops or parallel edges. (That is, there is at most one edge between each pair of distinct vertices.)

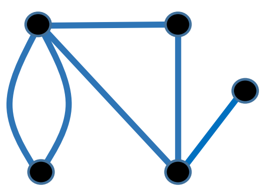
Simple graph



Non simple graph



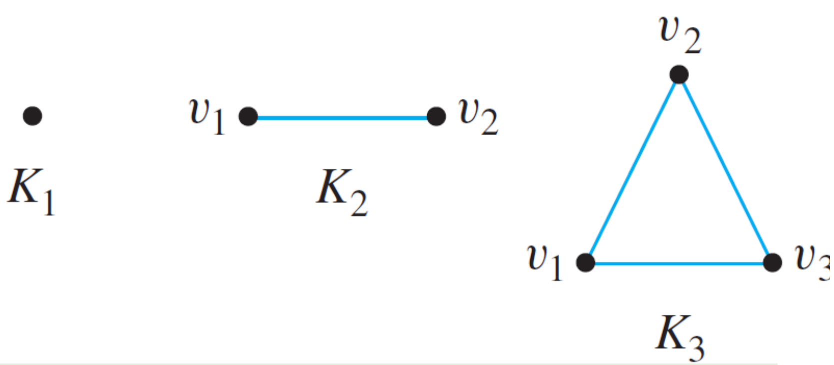
Non simple graph



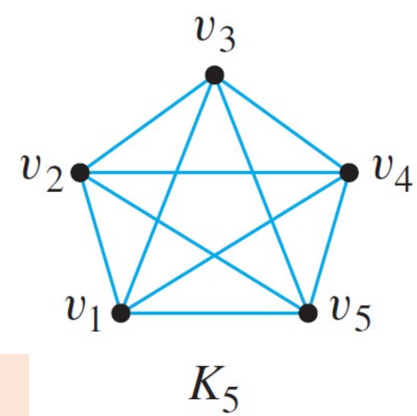
Complete Graphs

Definition: Complete Graph

A **complete graph** on ***n*** **vertices**, $n > 0$, denoted $\boldsymbol{K_n}$, is a simple graph with n vertices and exactly one edge connecting each pair of distinct vertices .



Draw K_5.



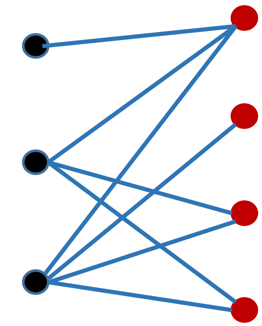
Fun: The notation K_n is rumoured to be used to honour the contributions of Kazimierz Kuratowski to graph theory.

Important fact:
How many edges are there in K_n?

Bipartite Graphs and Complete Bipartite Graphs

Definition: Bipartite Graph

A **bipartite graph** (or bigraph) is a simple graph whose vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V.

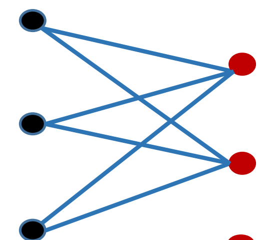


Bipartite graph

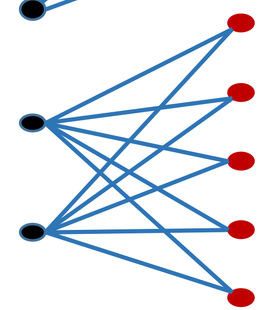
Definition: Complete Bipartite Graph

A **complete bipartite graph** is a bipartite graph on two disjoint sets U and V such that every vertex in U connects to every vertex in V.

If $|U| = m$ and $|V| = n$, the complete bipartite graph is denoted as $K_{m,n}$.



$K_{3,2}$



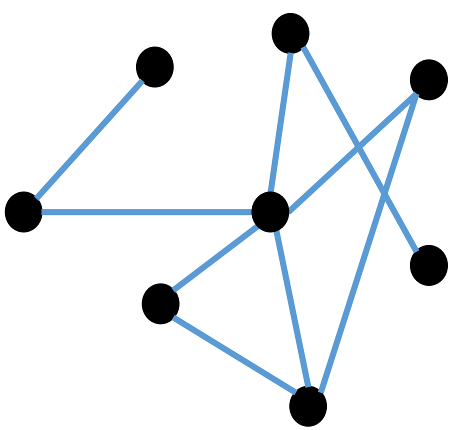
$K_{2,5}$

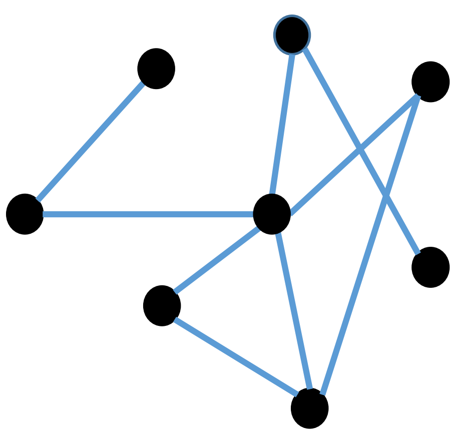
Subgraph of a Graph

Definition: Subgraph of a Graph

A graph H is said to be a **subgraph** of graph G if and only if every vertex in H is also a vertex in G, every edge in H is also an edge in G, and every edge in H has the same endpoints as it has in G.

A graph G





Subgraphs of G

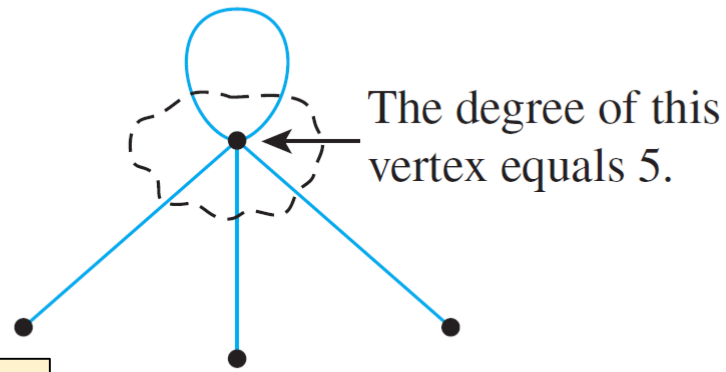
Degree of a Vertex and Total Degree of an Undirected Graph

Definition: Degree of a Vertex and Total Degree of an Undirected Graph

Let G be a undirected graph and v a vertex of G. The **degree** of v, denoted **deg(v)**, equals the number of edges that are incident on v, with an edge that is a loop counted twice.

The **total degree of G** is the sum of the degrees of all the vertices of G.

The degree of a vertex can be obtained from the drawing of a graph by counting how many end segments of edges are incident on the vertex.

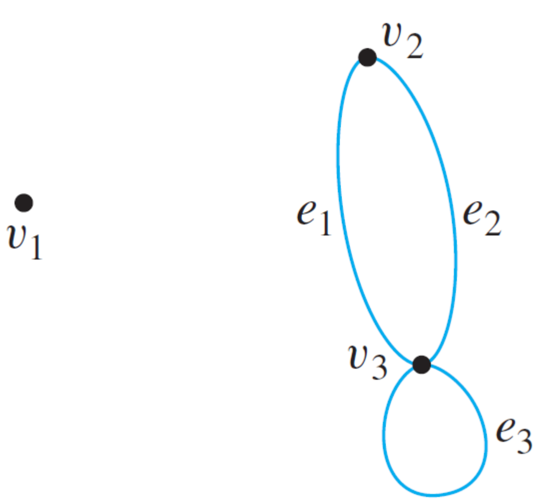


For directed graphs: Each vertex has an indegree and an outdegree. We will not cover that here.

Degree of a Vertex and Total Degree of a Graph

Example: Find the degree of each vertex of the graph G shown below. Then find the total degree of G.

$\deg(v_1) = 0$
$\deg(v_2) = 2$
$\deg(v_3) = 4$
Total degree of $G = 6$



The Concept of Degree

Theorem 10.1.1 The Handshake Theorem

If G is any graph, then the sum of the degrees of all the vertices of G equals twice the number of edges of G. Specifically, if the vertices of G are $v_1, v_2, \ldots, v_n$, where $n \geq 0$, then

$$\text{The total degree of } G = \deg(v_1) + \deg(v_2) + \ldots + \deg(v_n)$$
$$= 2 \times (\text{the number of edges of } G).$$

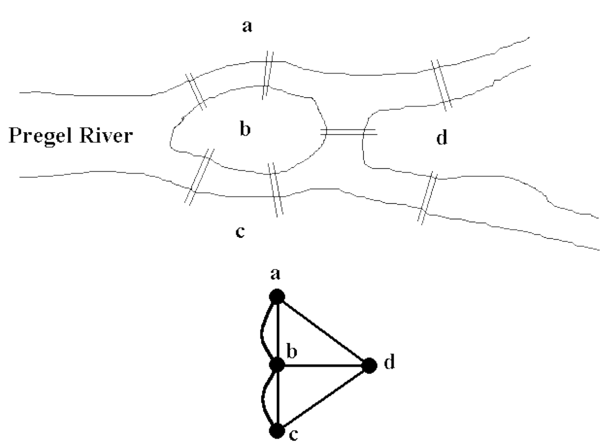
Corollary 10.1.2

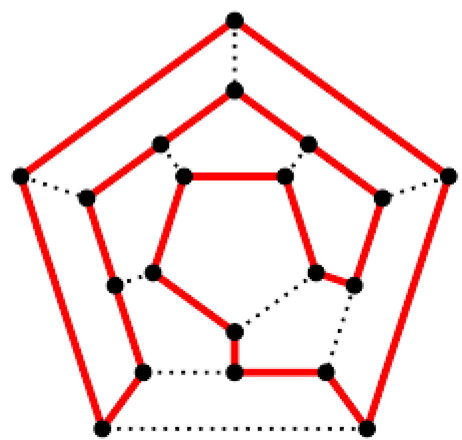
The total degree of a graph is even.

Proposition 10.1.3

In any graph there are an even number of vertices of odd degree.

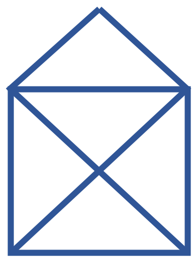
10.2 Trails, Paths, and Circuits



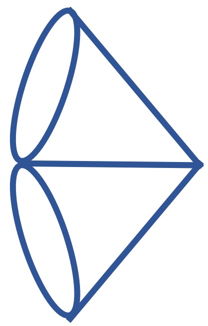


Let’s Have Some Fun

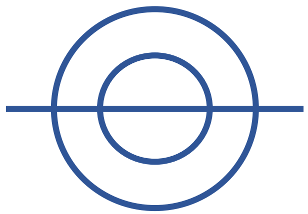
Can you draw the following figures without lifting up your pencil?



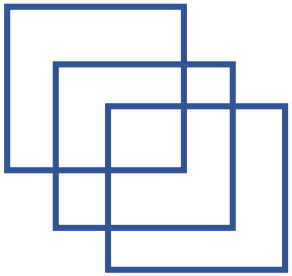
(1)



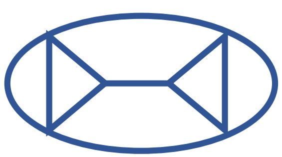
(2)



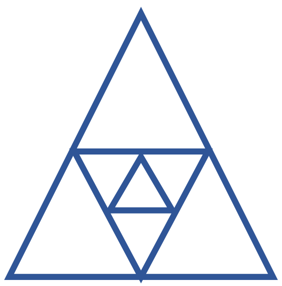
(3)



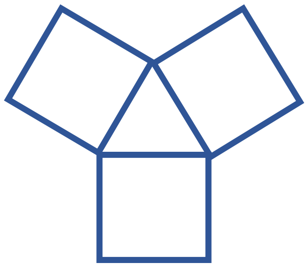
(4)



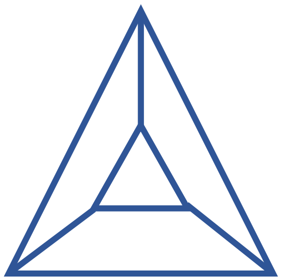
(5)



(6)



(7)

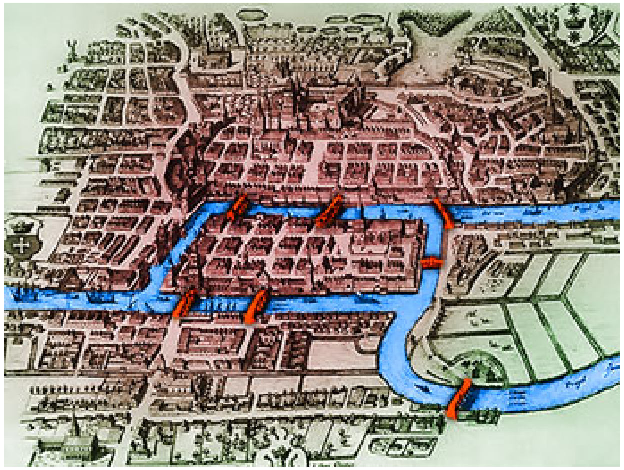


(8)

Königsberg bridges

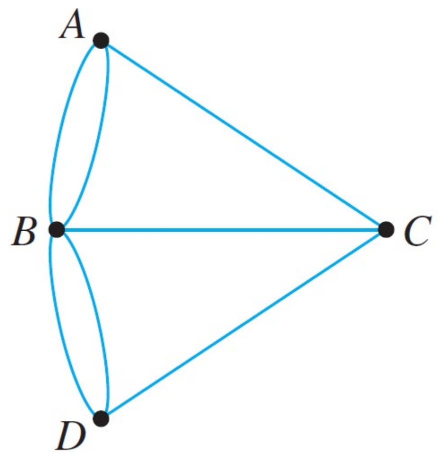
The subject of graph theory began in the year 1736 when the great mathematician Leonhard Euler published a paper giving the solution to the following puzzle:

The town of Königsberg in Prussia (now Kaliningrad in Russia) was built at a point where two branches of the Pregel River came together. It consisted of an island and some land along the river banks. These were connected by 7 bridges.



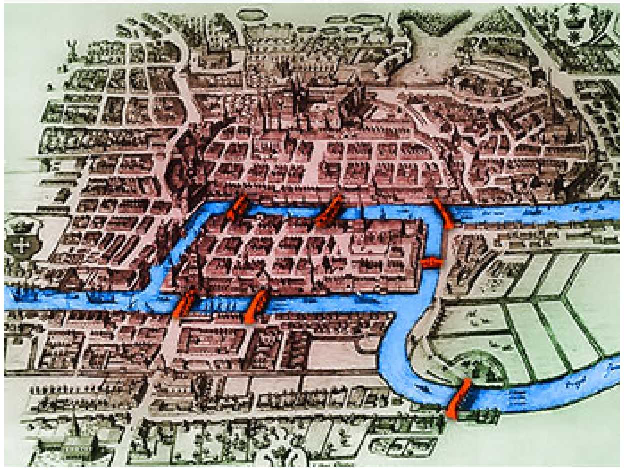
Euler asked: Is it possible to take a walk around town, starting and ending at the same location and crossing each of the 7 bridges exactly once?

Königsberg bridges



In terms of this graph, the question is:

Is it possible to find a route through the graph that starts and ends at some vertex (*A*, *B*, *C*, or *D*) and traverses each edge exactly once?

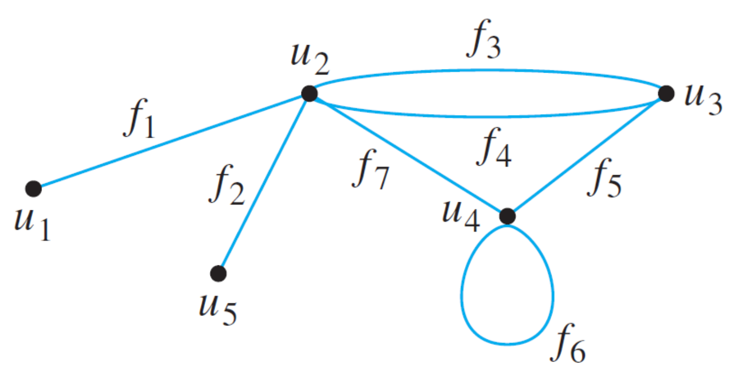


Euler asked: Is it possible to take a walk around town, starting and ending at the same location and crossing each of the 7 bridges exactly once?

Definitions

Travel in a graph is accomplished by moving from one vertex to another along a sequence of adjacent edges.

In the graph below, for instance, you can go from u_1 to u_4 by taking f_1 to u_2 and then f_7 to u_4. This is represented by writing $u_1 f_1 u_2 f_7 u_4$.



Or, you could take a longer route:

$$u_1 f_1 u_2 f_3 u_3 f_4 u_2 f_3 u_3 f_5 u_4$$

Walk, Trail, Path, Closed Walk, Circuit, Simple Circuit

Definitions

Let *G* be a graph, and let *v* and *w* be vertices of *G*.

A **walk from *v* to *w*** is a finite alternating sequence of adjacent vertices and edges of *G*. Thus a walk has the form

$$v_0\, e_1\, v_1\, e_2\, \ldots\, v_{n-1}\, e_n\, v_n\,,$$

where the *v*'s represent vertices, the *e*'s represent edges, $v_0=v$, $v_n=w$, and for all $i \in \{1, 2, \ldots, n\}$, v_{i-1} and v_i are the endpoints of e_i. The number of edges, *n*, is the **length** of the walk.

The **trivial walk** from *v* to *v* consists of the single vertex *v*.

A **trail from *v* to *w*** is a walk from *v* to *w* that does not contain a repeated edge.

A **path from *v* to *w*** is a trail that does not contain a repeated vertex.

Walk, Trail, Path, Closed Walk, Circuit, Simple Circuit

Definitions

A **closed walk** is a walk that starts and ends at the same vertex.

Circuit (or **cycle**): Let $n \in \mathbb{Z}_{\geq 3}$. An undirected graph $G(V, E)$ where $V = \{x_1, x_2, \cdots, x_n\}$ and $E = \{\{x_1, x_2\}, \{x_2, x_3\}, \cdots, \{x_{n-1}, x_n\}, \{x_n, x_1\}\}$ is called **a circuit/cycle**.

(A cycle is a closed walk that does not contain a repeated edge.)

A **simple circuit** (or **simple cycle**) is a circuit that does not have any other repeated vertex except the first and last.

An undirected graph is **cyclic** if it contains a loop or a cycle; otherwise, it is **acyclic**.

Notes

Because most of the major developments in graph theory have happened relatively recently and in a variety of different contexts, the terms used in the subject have not been standardized.

Susanna Epp's book	Others
Graph	Multigraph
Simple graph	Graph
Vertex	Node
Edge	Arc
Trail	Path
Path	Simple path
Simple circuit	Cycle

The terminology in this book is among the most common, but if you consult other sources, be sure to check their definitions.

For CS1231S, we will follow the terminology in Epp's book.

Connectedness

A graph is connected if it is possible to travel from any vertex to any other vertex along a sequence of adjacent edges of the graph.

Definition: Connectedness

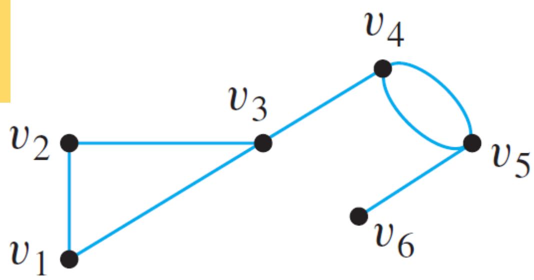
Two vertices v and w of a graph $G=(V,E)$ are **connected** if and only if there is a walk from v to w.

The graph *G* is connected if and only if given *any* two vertices v and w in G, there is a walk from v to w. Symbolically,

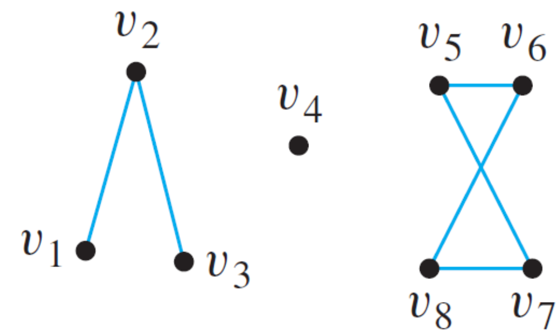
G is connected iff $\forall$ vertices v, $w \in V$, $\exists$ a walk from v to w.

Example: Which of the following graphs are connected?

Yes

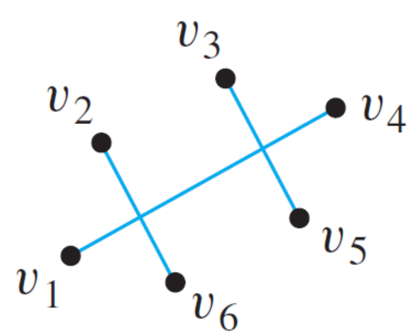


(a)



(b)

No



No

(c)

Some useful facts relating circuits and connectedness are collected in the following lemma.

Lemma 10.2.1

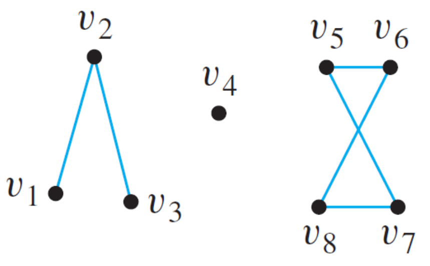
Let G be a graph.

a. If G is connected, then any two distinct vertices of G can be connected by a path.
b. If vertices v and w are part of a circuit in G and one edge is removed from the circuit, then there still exists a trail from v to w in G.
c. If G is connected and G contains a circuit, then an edge of the circuit can be removed without disconnecting G.

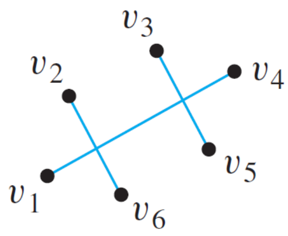
Connected Component

The graphs in (*b*) and (*c*) are both made up of three pieces, each of which is itself a connected graph.

A *connected component* of a graph is a connected subgraph of largest possible size.



(b)



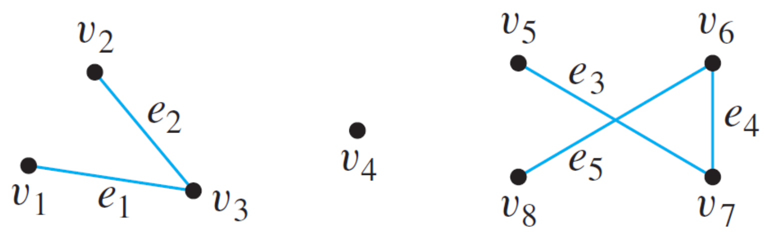
(c)

Definition: Connected Component

A graph H is a **connected component** of a graph G if and only if

1. The graph H is a subgraph of G;
2. The graph H is connected; and
3. No connected subgraph of G has H as a subgraph and contains vertices or edges that are not in H.

Find all connected components of the following graph G.



G has 3 connected components H_1, H_2 and H_3 with vertex sets V_1, V_2 and V_3 and edge sets E_1, E_2 and E_3 , where

$$V_1 = \{v_1, v_2, v_3\}, \qquad E_1 = \{e_1, e_2\}$$

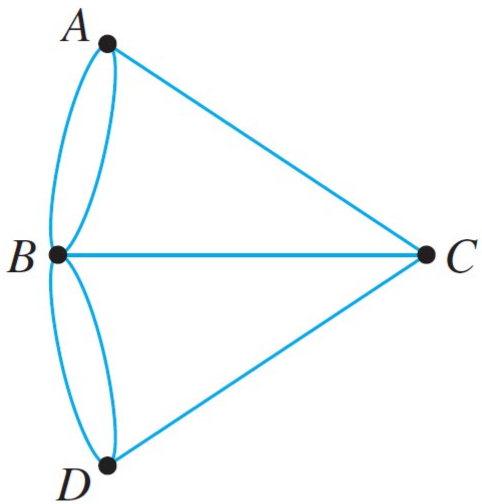
$$V_2 = \{v_4\}, \qquad E_2 = \emptyset$$

$$V_3 = \{v_5, v_6, v_7, v_8\}, \quad E_3 = \{e_3, e_4, e_5\}$$

Euler Circuits

Now, let’s go back to the puzzle of the Königsberg bridges.

Is it possible to find a route through the graph that starts and ends at some vertex, one of *A*, *B*, *C*, or *D*, and traverses each edge exactly once?



Definition: Euler Circuit

Let G be a graph. An **Euler circuit** for G is a circuit that contains every vertex and traverses every edge of G exactly once.

Definition: Eulerian Graph

An **Eulerian graph** is a graph that contains an Euler circuit.

Theorem 10.2.2

If a graph has an Euler circuit, then every vertex of the graph has positive even degree.

Contrapositive Version of Theorem 10.2.2

If some vertex of a graph has odd degree, then the graph does not have an Euler circuit.

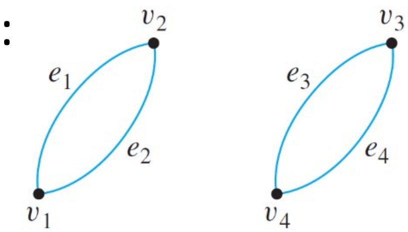
Euler Circuits

Is this true?

If every vertex of a graph has positive even degree, then the graph has an Euler circuit.

Not true!

Counterexample:



Theorem 10.2.3

If a graph G is connected and the degree of every vertex of G is a positive even integer, then G has an Euler circuit.

Theorem 10.2.4

A graph G has an Euler circuit if and only if G is connected and every vertex of G has positive even degree.

Definition: Euler Trail

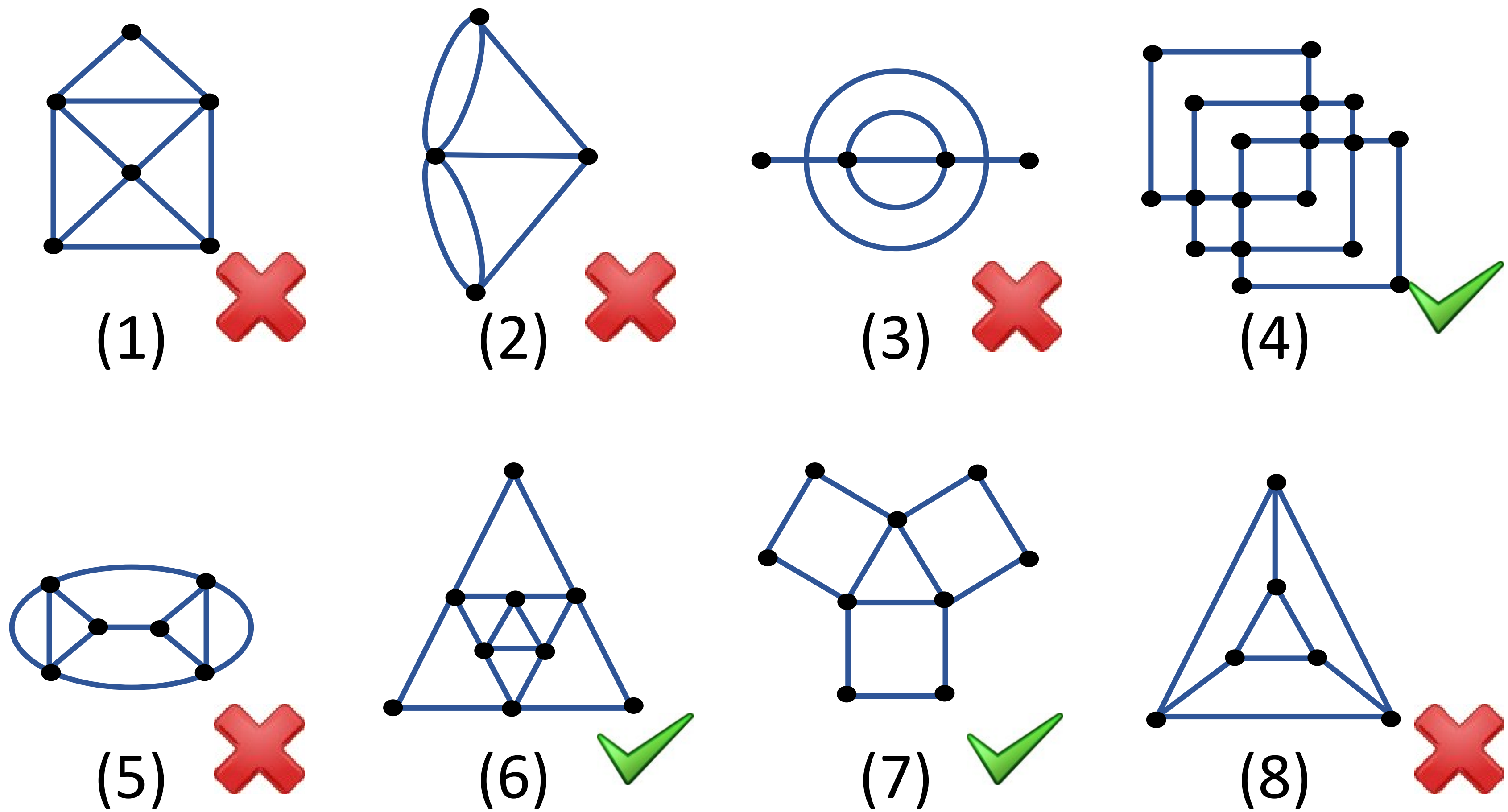
Let G be a graph, and let v and w be two distinct vertices of G. An **Euler trail/path from *v* to *w*** is a sequence of adjacent edges and vertices that starts at v, ends at w, passes through every vertex of G at least once, and traverses every edge of G exactly once.

Corollary 10.2.5

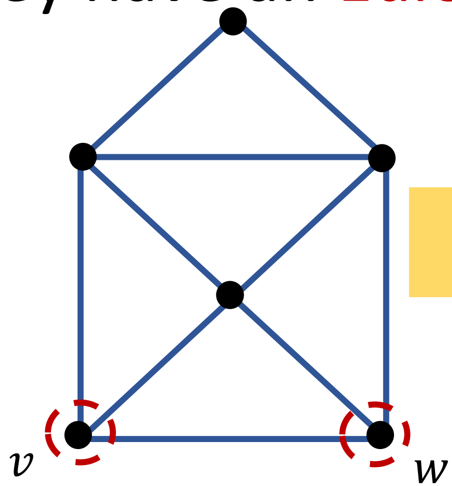
Let G be a graph, and let v and w be two distinct vertices of G. There is an Euler trail from v to w if and only if G is connected, v and w have odd degree, and all other vertices of G have positive even degree.

Euler Circuits

Does each of the following graphs have an Euler circuit?

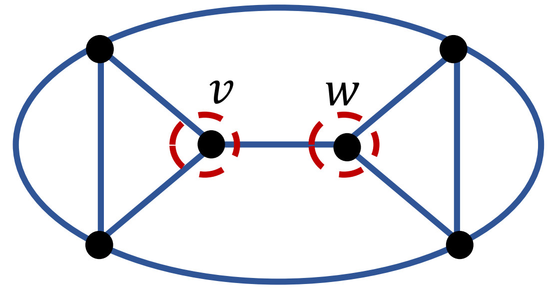


The following graphs do not have an Euler circuit.
Do they have an **Euler trail**?



Yes

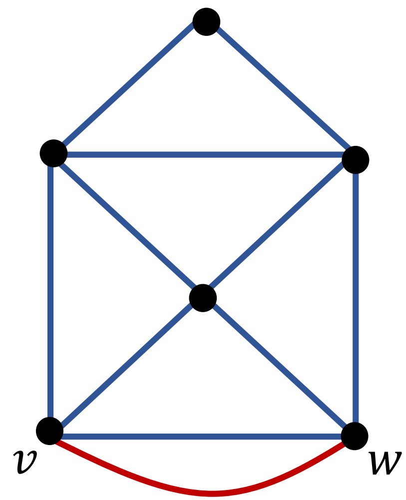
(1)

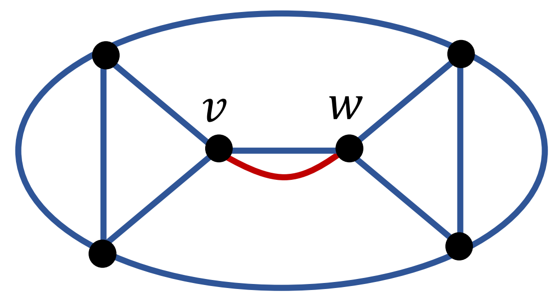


Yes

(5)

Adding an edge between the two vertices with odd degree will give us an Euler circuit.





Hamiltonian Circuits

Recall Theorem 10.2.4:

Theorem 10.2.4

A graph *G* has an Euler circuit if and only if *G* is connected and every vertex of *G* has positive even degree.

A related question:

Given a graph *G*, is it possible to find a circuit for *G* in which all the *vertices* of *G* (except the first and the last) appear exactly once?

Hamiltonian Circuits

In 1859 the Irish mathematician Sir William Rowan Hamilton introduced a puzzle in the shape of a dodecahedron (DOH-dek-a-HEE-dron). (Figure 10.2.6 contains a drawing of a dodecahedron, which is a solid figure with 12 identical pentagonal faces.)

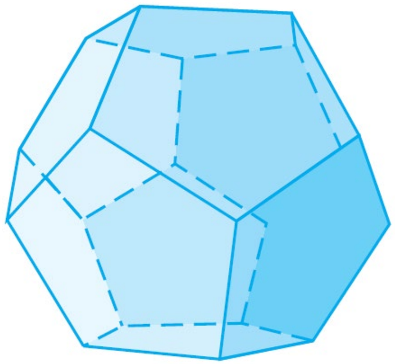
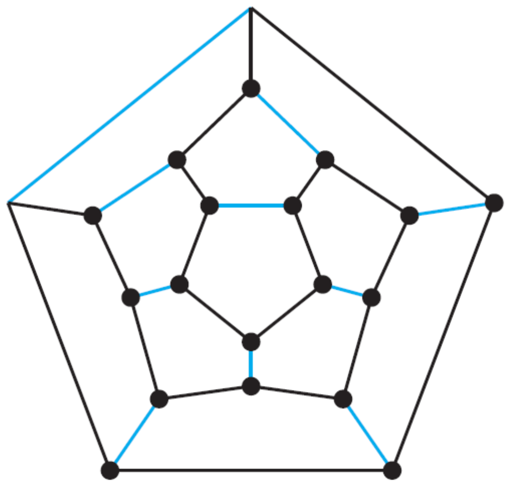


Figure 10.2.6 Dodecahedron

Each vertex was labeled with the name of a city — London, Paris, Singapore, New York, and so on.

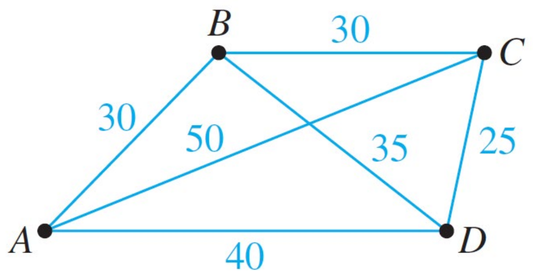
The problem Hamilton posed was to start at one city and tour the world by visiting each other city exactly once and returning to the starting city.

One way to solve the puzzle is to imagine the surface of the dodecahedron stretched out and laid flat in the plane, as follows:



The circuit denoted with black lines is one solution. (Note that although every city is visited, many edges are omitted from the circuit.)

If we add values (called weights) to each edge, this becomes the travelling salesman problem.



Definition: Hamiltonian Circuit

Given a graph *G*, a **Hamiltonian circuit** for *G* is a simple circuit that includes every vertex of *G*. (That is, every vertex appears exactly once, except for the first and the last, which are the same.)

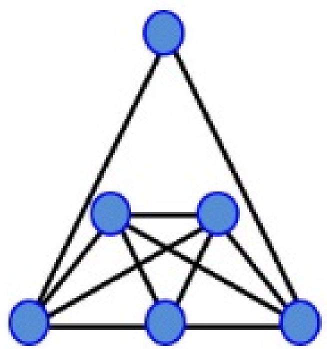
Definition: Hamiltonian Graph

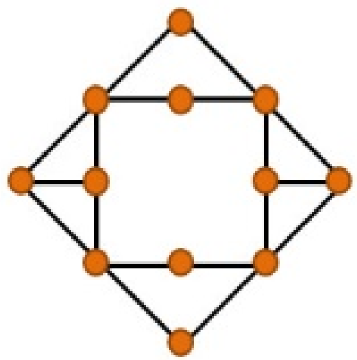
A **Hamiltonian graph** (also called **Hamilton graph**) is a graph that contains a Hamiltonian circuit.

Note that although an Euler circuit for a graph *G* must include every vertex of *G*, it may visit some vertices more than once and hence may not be a Hamiltonian circuit.

On the other hand, a Hamiltonian circuit for *G* does not need to include all the edges of *G* and hence may not be an Euler circuit.

AY2019/20 Sem1 Exam Question


Graph *A*


Graph *B*

Which of the following statements is true?

A. Graphs *A* and *B* are both Eulerian and Hamiltonian.

B. Graph *A* is both Eulerian and Hamiltonian; graph *B* is neither Eulerian nor Hamiltonian.

C. Graph *A* is Eulerian but not Hamiltonian; graph *B* is neither Eulerian nor Hamiltonian.

D. Graph *A* is Eulerian but not Hamiltonian; graph *B* is Hamiltonian but not Eulerian.

E. Graphs *A* and *B* are Hamiltonian but not Eulerian.

Despite the analogous-sounding definitions of Euler and Hamiltonian circuits, the mathematics of the two are <u>very different</u>.

Determining whether a graph has an Euler circuit is easy – Theorem 10.2.4 gives a simple criterion.

Theorem 10.2.4

A graph G has an Euler circuit if and only if G is connected and every vertex of G has positive even degree.

Unfortunately, there is <u>no analogous criterion</u> for determining whether a given graph has a Hamiltonian circuit, nor is there even an efficient algorithm for finding such a circuit.

There is, however, a simple technique that can be used in many cases to show that a graph does *not* have a Hamiltonian circuit.

Proposition 10.2.6

If a graph G has a Hamiltonian circuit, then G has a subgraph H with the following properties:

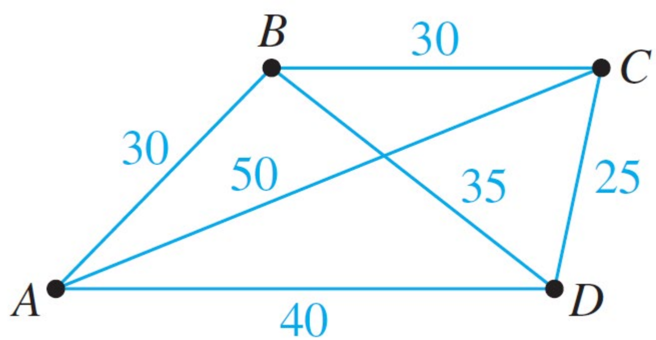
1. H contains every vertex of G.
2. H is connected.
3. H has the same number of edges as vertices.
4. Every vertex of H has degree 2.

The contrapositive of Proposition 10.2.6 says that if a graph G does *not* have a subgraph H with properties (1)–(4), then G does *not* have a Hamiltonian circuit.

For reading only.

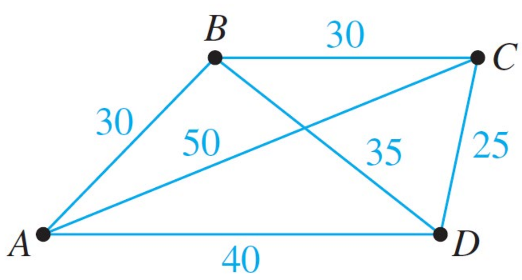
Travelling Salesman Problem

Imagine that the drawing below is a map showing four cities and the distances in kilometers between them.



Suppose that a salesman must travel to each city exactly once, starting and ending in city *A*. Which route from city to city will minimize the total distance that must be travelled?

This problem can be solved by writing all possible Hamiltonian circuits starting and ending at *A* and calculating the total distance travelled for each.



Route	Total Distance (In Kilometers)	
$ABCDA$	$30 + 30 + 25 + 40 = 125$	
$ABDCA$	$30 + 35 + 25 + 50 = 140$	
$ACBDA$	$50 + 30 + 35 + 40 = 155$	
$ACDBA$	140	[$ABDCA$ backwards]
$ADBCA$	155	[$ACBDA$ backwards]
$ADCBA$	125	[$ABCDA$ backwards]

Thus either route *ABCDA* or *ADCBA* gives a minimum total distance of 125 km.

The general travelling salesman problem involves finding a Hamiltonian circuit to minimize the total distance travelled for an arbitrary graph with n vertices in which each edge is marked with a distance.

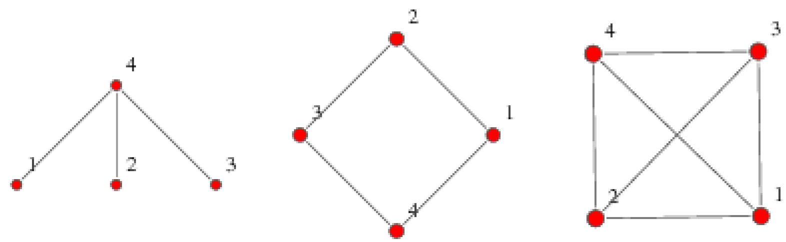
One way to solve the general problem is to use the previous method: Write down all Hamiltonian circuits starting and ending at a particular vertex, compute the total distance for each, and pick one for which this total is minimal.

However, this is impractical for even medium-sized values of n. For $n = 30$ vertices, there would be $(29!)/2 \approx 4.42 \times 10^{30}$ Hamiltonian circuits starting and ending at a particular vertex to check. If each circuit could be found and its total distance computed in just one nanosecond, it would take approximately 1.4×10^{14} years to compute!

At present, there is no known algorithm for solving the general travelling salesman problem that is more efficient.

However, there are efficient algorithms that find “pretty good” solutions — that is, circuits that, while not necessarily having the least possible total distances, have smaller total distances than most other Hamiltonian circuits.

10.3 Matrix Representations of Graphs



$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Matrices

Definition: Matrix

An $m \times n$ (read "m by n") **matrix** A over a set S is a rectangular array of elements of S arranged into m rows and n columns.

$$\mathbf{A} = \begin{array}{c} \left[\begin{array}{cccccc} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mj} & \cdots & a_{mn} \end{array}\right] \end{array} \begin{array}{l} \\ \\ \\ \longleftarrow i\text{th row of } \mathbf{A} \\ \\ \\ \end{array}$$

*j*th column of **A**

We write **A** = (a_{ij}).

Matrices

If **A** and **B** are matrices, then **A** = **B** if, and only if, **A** and **B** have the same size and the corresponding entries of **A** and **B** are all equal; that is,

$$a_{ij} = b_{ij} \text{ for all } i = 1, 2, \cdots, m \text{ and } j = 1, 2, \cdots, n.$$

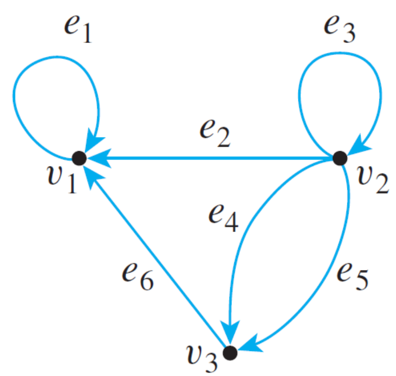
A matrix for which the numbers of rows and columns are equal is called a square matrix.

If **A** is a square matrix of size $n \times n$, then the main diagonal of **A** consists of all the entries $a_{11}, a_{22}, \cdots a_{nn}$.

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1i} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2i} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ii} & \dots & a_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{n1} & a_{n2} & \dots & a_{ni} & \dots & a_{nn} \end{bmatrix}$$

main diagonal of **A**

Matrices and Directed Graphs



Directed Graph G

(a)

$$\mathbf{A} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 2 \\ 1 & 0 & 0 \end{bmatrix} \end{array}$$

Adjacency Matrix

(b)

Figure 10.3.1 A Directed Graph and Its Adjacency Matrix

This graph G is represented by the matrix **A** = (a_{ij}) for which a_{ij} = number of arrows from v_i to v_j for all i = 1, 2, 3 and j = 1, 2, 3.

A is called the adjacency matrix of G.

Another common representation of a graph is the adjacency list, which is covered in algorithms module.

Definition: Adjacency Matrix of a Directed Graph

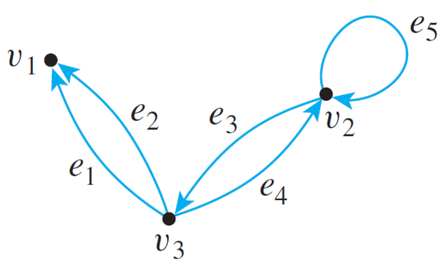
Let G be a directed graph with ordered vertices v_1, v_2, ... v_n. The **adjacency matrix of *G*** is the $n \times n$ matrix $\mathbf{A} = (a_{ij})$ over the set of non-negative integers such that

$$a_{ij} = \text{the number of arrows from } v_i \text{ to } v_j \text{ for all } i, j = 1, 2, \ldots, n.$$

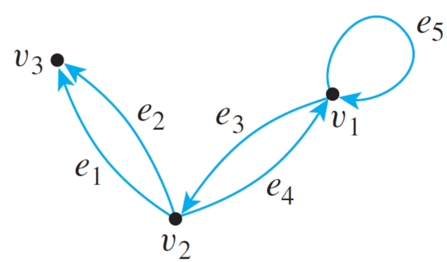
Example: Find the adjacency matrices of the two directed graphs below.

$$\begin{array}{c c} & \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \end{array} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 2 & 1 & 0 \end{bmatrix} \end{array}$$

(a)



(a)



(b)

$$\begin{array}{c c} & \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ v_3 \end{array} & \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

(b)

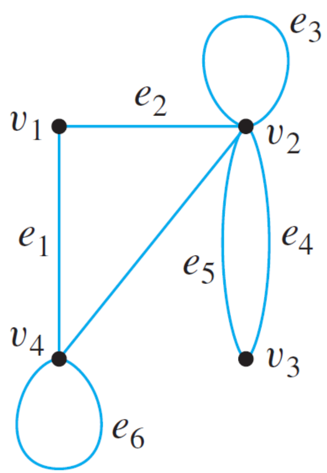
Matrices and Undirected Graphs

Definition: Adjacency Matrix of an Undirected Graph

Let G be an undirected graph with ordered vertices $v_1, v_2, \ldots v_n$. The **adjacency matrix of G** is the $n \times n$ matrix $\mathbf{A} = (a_{ij})$ over the set of non-negative integers such that

a_{ij} = the number of edges connecting v_i and v_j for all $i, j = 1, 2, \ldots, n$.

Example: Find the adjacency matrix for the graph G shown below.



$$\mathbf{A} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 2 & 1 \\ 0 & 2 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

Note that the matrix is symmetric.

Definition: Symmetric Matrix

An $n \times n$ square matrix $A = (a_{ij})$ is called **symmetric** if, and only if, $a_{ij} = a_{ji}$ for all $i, j = 1, 2, \ldots, n$.

Matrix Multiplication

Definition: Scalar Product

Suppose that all entries in matrices **A** and **B** are real numbers. If the number of elements, *n*, in the *i*th row of **A** equals the number of elements in the *j*th column of **B**, then the **scalar product** or **dot product** of the *i*th row of **A** and the *j*th column of **B** is the real number obtained as follows:

$$\begin{bmatrix} a_{i1} & a_{i2} & \cdots & a_{in} \end{bmatrix} \begin{bmatrix} b_{1j} \\ b_{2j} \\ \vdots \\ b_{nj} \end{bmatrix} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{in}b_{nj}.$$

Definition: Matrix Multiplication

Let **A** = (a_{ij}) be an $m \times k$ matrix and **B** = (b_{ij}) an $k \times n$ matrix with real entries. The (matrix) product of **A** times **B**, denoted **AB**, is that matrix (c_{ij}) defined as follows:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ik} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mk} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1j} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2j} & \cdots & b_{2n} \\ & \vdots & & \vdots & & \vdots \\ & \vdots & & \vdots & & \vdots \\ b_{k1} & b_{k2} & \cdots & b_{kj} & \cdots & b_{kn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1j} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2j} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ c_{i1} & c_{i2} & \cdots & \boxed{c_{ij}} & \cdots & c_{in} \\ \vdots & \vdots & & \vdots & & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mj} & \cdots & c_{mn} \end{bmatrix}$$

where

$$c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j} + \cdots + a_{ik}b_{kj} = \sum_{r=1}^{k} a_{ir}b_{rj}.$$

for all i = 1, 2, …, m and j = 1, 2, …, n.

Example – Computing a Matrix Product

Let $\mathbf{A} = \begin{bmatrix} 2 & 0 & 3 \\ -1 & 1 & 0 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 4 & 3 \\ 2 & 2 \\ -2 & -1 \end{bmatrix}$. Compute $\mathbf{AB}$.

Solution:

$$\begin{bmatrix} 2 & 0 & 3 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 4 & 3 \\ 2 & 2 \\ -2 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ c_{21} & c_{22} \end{bmatrix},$$

where

$$c_{11} = 2\cdot 4 + 0\cdot 2 + 3\cdot(-2) = 2 \qquad \begin{bmatrix} 2 & 0 & 3 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 4 & 3 \\ 2 & 2 \\ -2 & -1 \end{bmatrix}$$

$$c_{12} = 2\cdot 3 + 0\cdot 2 + 3\cdot(-1) = 3 \qquad \begin{bmatrix} 2 & 0 & 3 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 4 & 3 \\ 2 & 2 \\ -2 & -1 \end{bmatrix}$$

Example – Computing a Matrix Product

Let $\mathbf{A} = \begin{bmatrix} 2 & 0 & 3 \\ -1 & 1 & 0 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 4 & 3 \\ 2 & 2 \\ -2 & -1 \end{bmatrix}$. Compute $\mathbf{AB}$.

Solution:

$$\begin{bmatrix} 2 & 0 & 3 \\ -1 & 1 & 0 \end{bmatrix}\begin{bmatrix} 4 & 3 \\ 2 & 2 \\ -2 & -1 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ -2 & -1 \end{bmatrix},$$

where

$$c_{21} = (-1)\cdot 4 + 1\cdot 2 + 0\cdot(-2) = -2 \qquad \begin{bmatrix} 2 & 0 & 3 \\ -1 & 1 & 0 \end{bmatrix}\begin{bmatrix} 4 & 3 \\ 2 & 2 \\ -2 & -1 \end{bmatrix}$$

$$c_{22} = (-1)\cdot 3 + 0\cdot 2 + 3\cdot(-1) = -1 \qquad \begin{bmatrix} 2 & 0 & 3 \\ -1 & 1 & 0 \end{bmatrix}\begin{bmatrix} 4 & 3 \\ 2 & 2 \\ -2 & -1 \end{bmatrix}.$$

Matrix Multiplication

Multiplication of real numbers is commutative, but matrix multiplication is not.

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 0 & 1 \end{bmatrix}, \quad \text{but} \quad \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

On the other hand, both real number and matrix multiplications are associative ($(ab)c = a(bc)$, for all elements a, b, and c for which the products are defined).

Identity Matrix

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}$$

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}.$$

These computations show that $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ acts as an identity on the left side for multiplication with 2 × 3 matrices and that $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ acts as an identity on the right side for multiplication with 3 × 3 matrices.

Definition: Identity Matrix

For each positive integer *n*, the ***n* × *n* identity matrix**, denoted $I_n = (\delta_{ij})$ or just **I** (if the size of the matrix is obvious from context), is the *n* × *n* matrix in which all the entries in the main diagonal are 1's and all other entries are 0's. In other words,

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} \qquad \text{for all } i, j = 1, 2, \ldots, n.$$

The German mathematician Leopold Kronecker introduced the symbol δ_{ij} to make matrix computations more convenient. In his honour, this symbol is called the *Kronecker delta*.

n^{th} Power of a Matrix

Definition: n^{th} Power of a Matrix

For any $n \times n$ matrix **A**, the **powers of A** are defined as follows:

$\mathbf{A}^0 = \mathbf{I}$ where $\mathbf{I}$ is the $n \times n$ identity matrix

$\mathbf{A}^n = \mathbf{A}\,\mathbf{A}^{n-1}$ for all integers $n \geq 1$

Let $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix}$. Compute $\mathbf{A}^0$, $\mathbf{A}^1$, $\mathbf{A}^2$, and $\mathbf{A}^3$.

Solution:

$$\mathbf{A}^0 = \text{the } 2 \times 2 \text{ identity matrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{A}^1 = \mathbf{A}\mathbf{A}^0 = \mathbf{A}\mathbf{I} = \mathbf{A}$$

$$\mathbf{A}^2 = \mathbf{A}\mathbf{A}^1 = \mathbf{A}\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 5 & 2 \\ 2 & 4 \end{bmatrix}$$

$$\mathbf{A}^3 = \mathbf{A}\mathbf{A}^2 = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix}\begin{bmatrix} 5 & 2 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 9 & 10 \\ 10 & 4 \end{bmatrix}$$

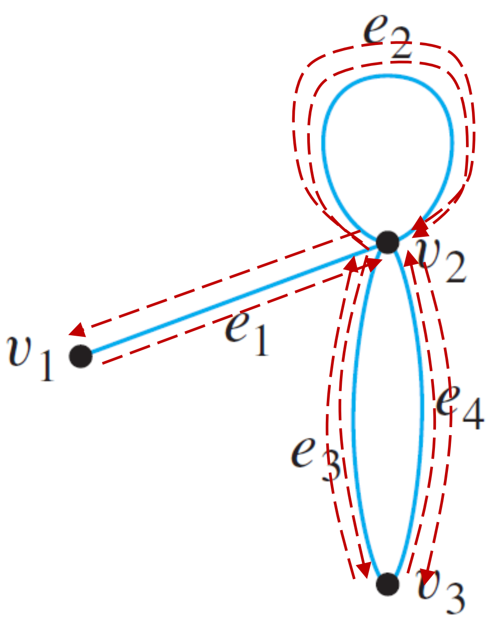
Counting Walks of Length N

A walk in a graph consists of an alternating sequence of vertices and edges.

If repeated edges are counted each time they occur, then the number of edges in the sequence is called the **length** of the walk.

For instance, the walk $v_2e_3v_3e_4v_2e_2v_2e_3v_3$ has length 4 (counting e_3 twice).

Example: Consider the following graph G.
How many distinct walks of length 2 connect v_2 and v_2?



One walk of length 2 from v_2 to v_2 via v_1:
$v_2e_1v_1e_1v_2$.

One walk of length 2 from v_2 to v_2 via v_2:
$v_2e_2v_2e_2v_2$.

Four walks of length 2 from v_2 to v_2 via v_3:
$v_2e_3v_3e_4v_2$,
$v_2e_4v_3e_3v_2$,
$v_2e_3v_3e_3v_2$,
$v_2e_4v_3e_4v_2$.

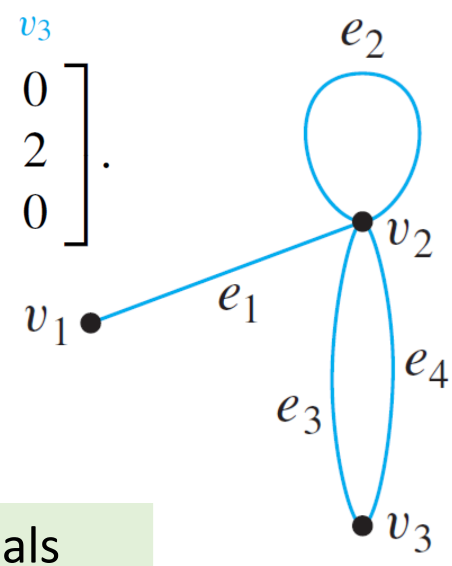
Total = 6

Counting Walks of Length N

The general question of finding the number of walks that have a given length and connect two particular vertices of a graph can easily be answered using matrix multiplication.

Consider the adjacency matrix **A** of the graph G.

$$\mathbf{A} = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \end{array} \begin{array}{c} \begin{array}{ccc} v_1 & v_2 & v_3 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 2 \\ 0 & 2 & 0 \end{bmatrix} \end{array}.$$



Compute $\mathbf{A}^2$: $$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 2 \\ 0 & 2 & 0 \end{bmatrix}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 2 \\ 0 & 2 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 2 \\ 1 & 6 & 2 \\ 2 & 2 & 4 \end{bmatrix}.$$

Note that the entry in row 2 and column 2 is 6, which equals the number of walks of length 2 from v_2 to v_2.

Reason: To compute a_{22}, you multiply row 2 of **A** with column 2 of **A** to obtain a sum of three terms:

$$\begin{bmatrix} 1 & 1 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = 1\cdot 1 + 1\cdot 1 + 2\cdot 2.$$

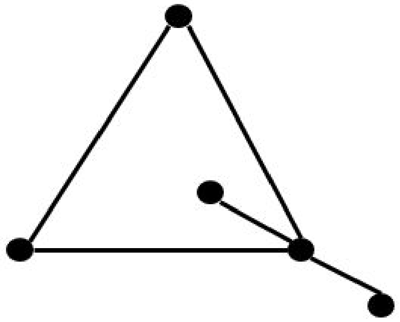
More generally, if **A** is the adjacency matrix of a graph *G*, the *i j*-th entry of $\mathbf{A}^2$ equals the number of walks of length 2 connecting the *i*-th vertex to the *j*-th vertex of *G*.

Even more generally, if *n* is any positive integer, the *i j*-th entry of $\mathbf{A}^n$ equals the number of walks of length *n* connecting the *i*-th and the *j*-th vertices of *G*.

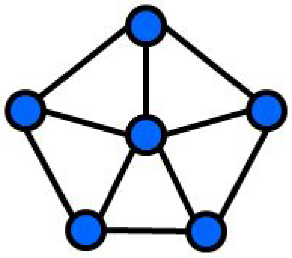
Theorem 10.3.2

If *G* is a graph with vertices $v_1, v_2, \ldots, v_m$ and **A** is the adjacency matrix of *G*, then for each positive integer *n* and for all integers $i, j = 1, 2, \ldots, m$,
the *ij*-th entry of $\mathbf{A}^n$ = the number of walks of length *n* from v_i to v_j.

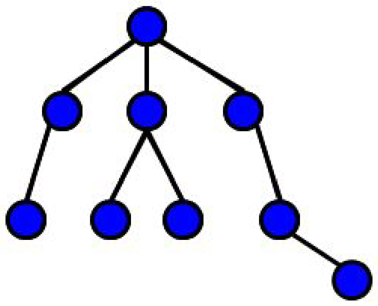
10.4 Planar Graphs



n=5, m=5, f=2



n=6, m=10, f=6



n=9, m=8, f=1

Isomorphisms of Graphs

The two drawings shown in Figure 10.4.1 both represent the same graph: Their vertex and edge sets are identical, and their edge-endpoint functions are the same. Call this graph *G*.

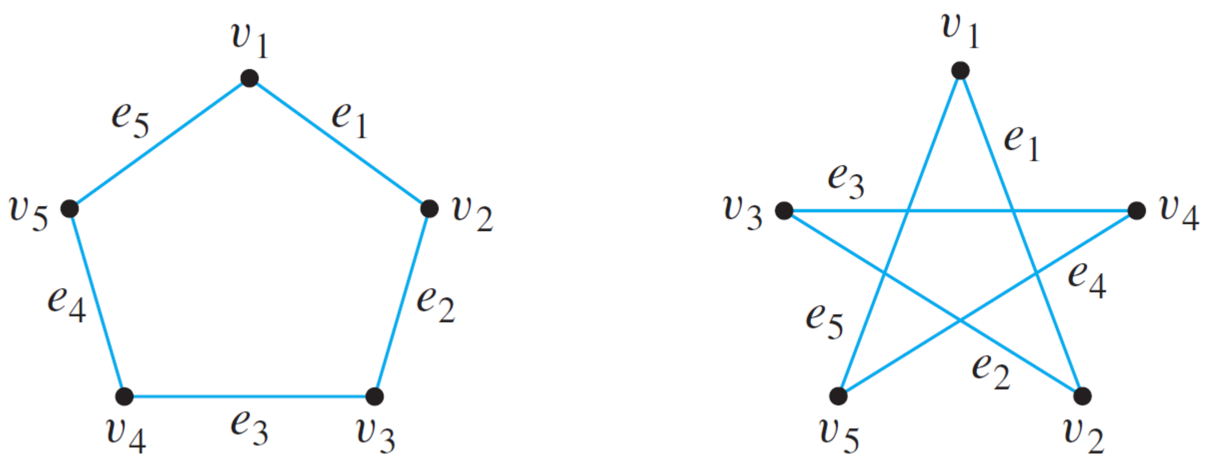


Figure 10.4.1

Now consider the graph G' represented in Figure 10.4.2.

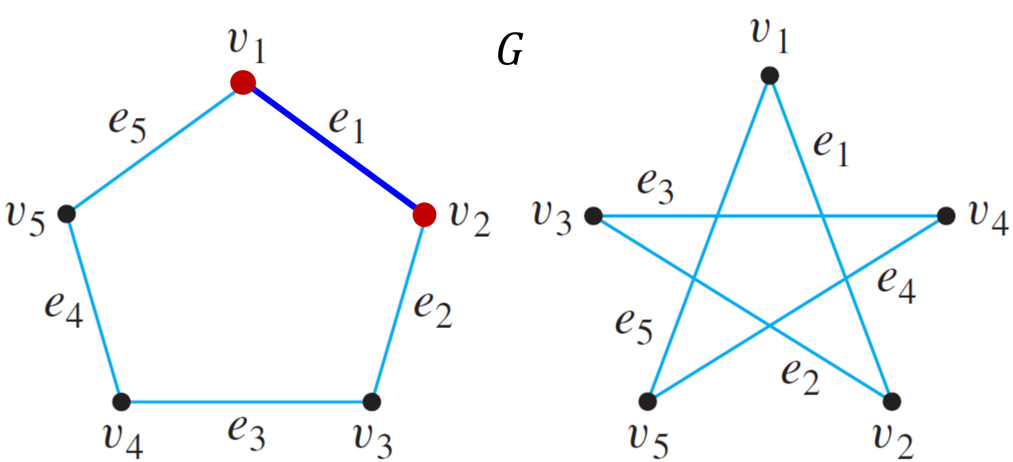


Figure 10.4.1

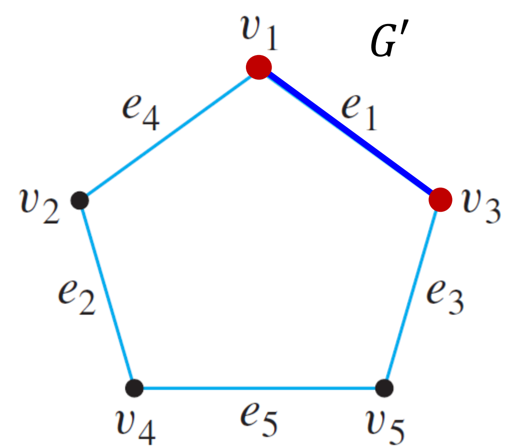
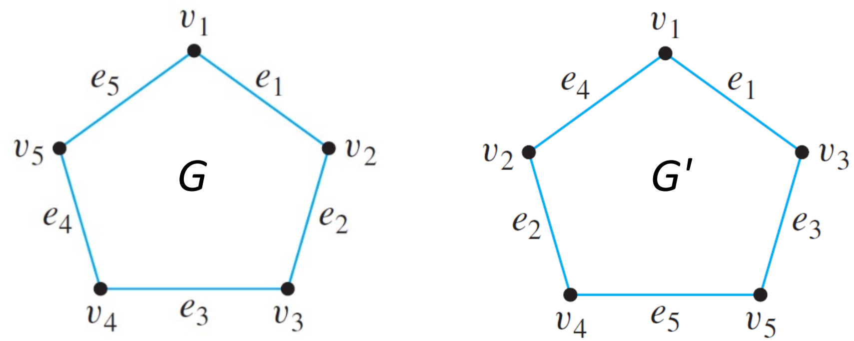


Figure 10.4.2

Observe that G' is a "different graph" from G in terms of the labelling of the vertices and edges (for instance, in G the endpoints of e_1 are v_1 and v_2, whereas in G' the endpoints of e_1 are v_1 and v_3).

Isomorphisms of Graphs

Yet G' is certainly very similar to G. In fact, if the vertices and edges of G' are relabeled by the functions shown in Figure 10.4.3, then G' becomes the same as G.



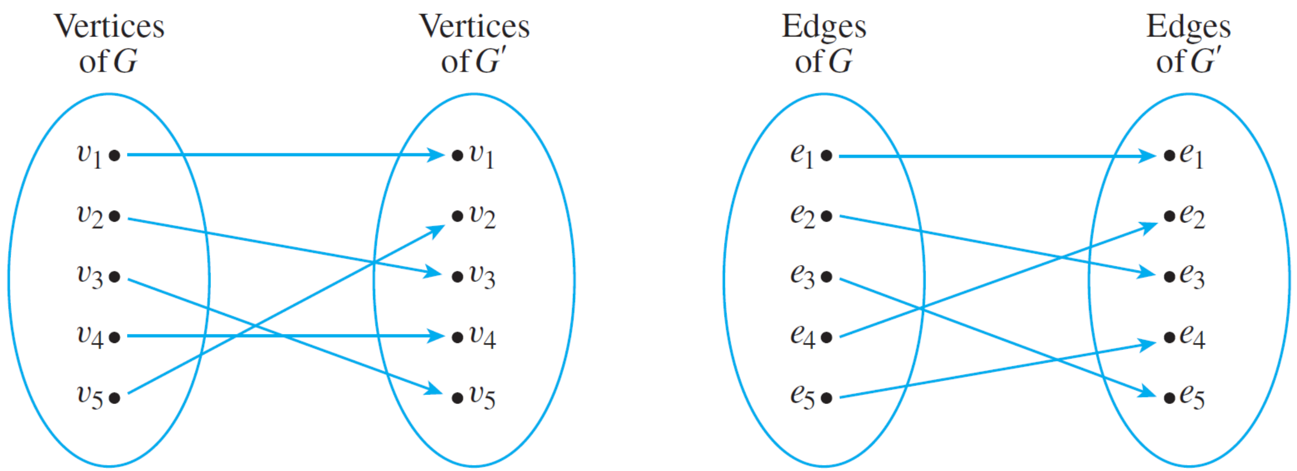


Figure 10.4.3

Note that these relabeling functions are **bijective**.

Two graphs G and G' that are the same except for the labeling of their vertices and edges are called *isomorphic*. In other words, there exists matching between the vertices such that two vertices are connected by an edge in G if and only if corresponding vertices are connected by an edge in G'.

Definition: Isomorphic Graph

Let $G = (V_G, E_G)$ and $G' = (V_{G'}, E_{G'})$ be two graphs.

G* is isomorphic to *G', denoted $G \cong G'$, if and only if there exist bijections $g: V_G \to V_{G'}$ and $h: E_G \to E_{G'}$ that preserve the edge-endpoint functions of G and G' in the sense that for all $v \in V_G$ and $e \in E_G$,

$$v \text{ is an endpoint of } e \Leftrightarrow g(v) \text{ is an endpoint of } h(e).$$

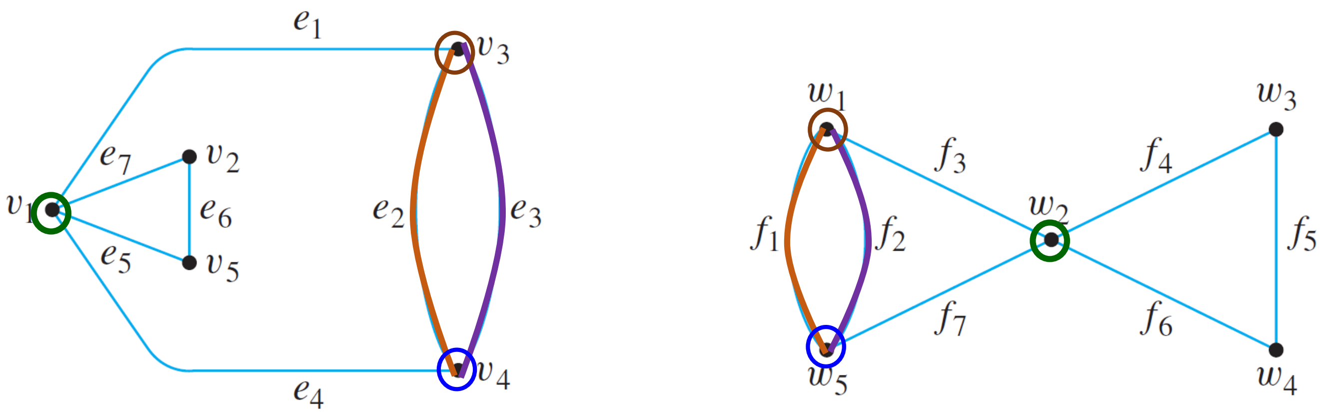
Alternative definition

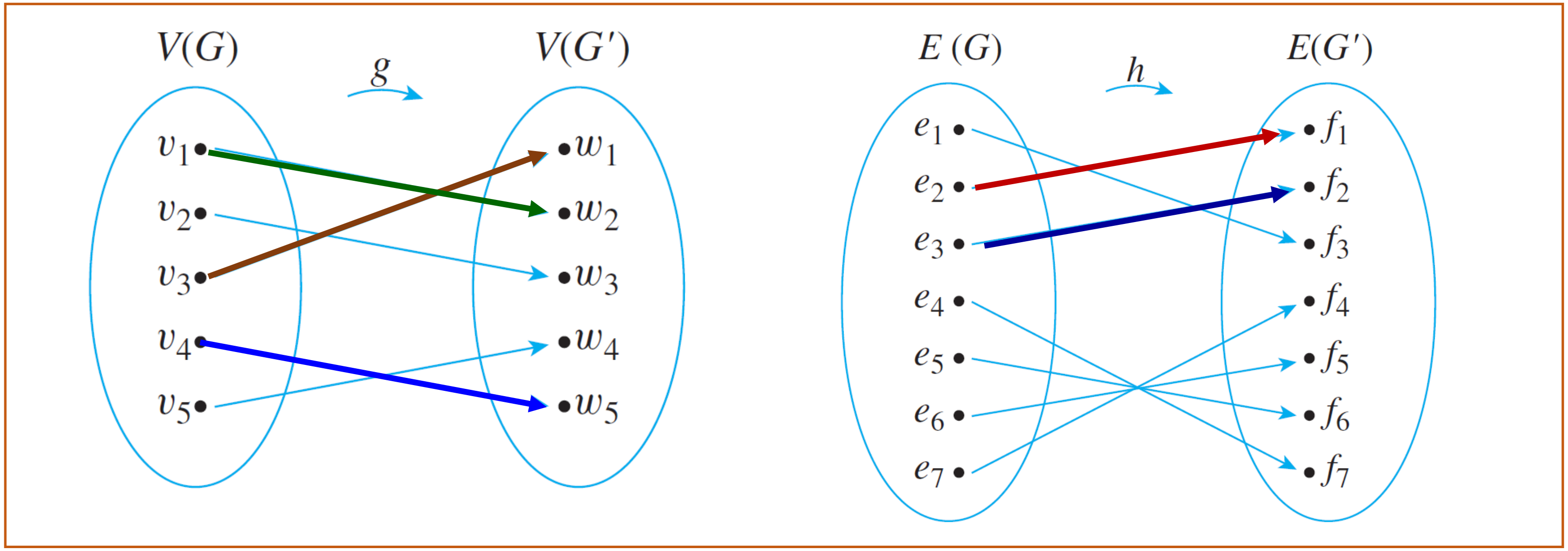
Let $G = (V_G, E_G)$ and $G' = (V_{G'}, E_{G'})$ be two graphs.

G* is isomorphic to *G' if and only if there exists a permutation $\pi: V_G \to V_{G'}$ such that $\{u, v\} \in E_G \Leftrightarrow \{\pi(u), \pi(v)\} \in E_{G'}$.

Isomorphisms of Graphs

Example: Show that the following two graphs are isomorphic.





It is not hard to show that graph isomorphism is an equivalence relation on a set of graphs; in other words, it is reflexive, symmetric, and transitive.

Theorem 10.4.1 Graph Isomorphism is an Equivalence Relation

Let S be a set of graphs and let $\cong$ be the relation of graph isomorphism on S. Then $\cong$ is an equivalence relation on S.

Exercise: Prove that graph isomorphism $\cong$ is an equivalence relation.

Definition: Planar Graph

A **planar graph** is a graph that can be drawn on a (two-dimensional) plane without edges crossing.

Is Figure 10.4.4 a planar graph?

Yes, it is a planar graph.

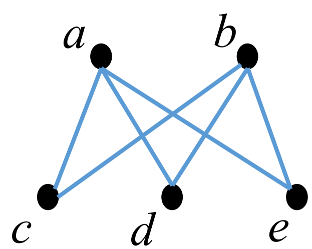
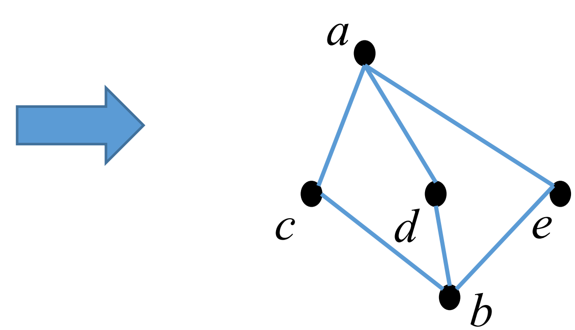


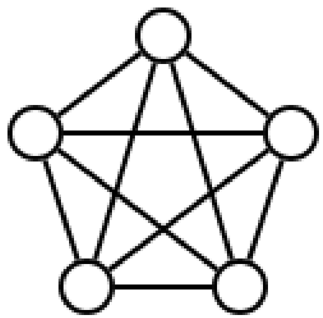
Figure 10.4.4



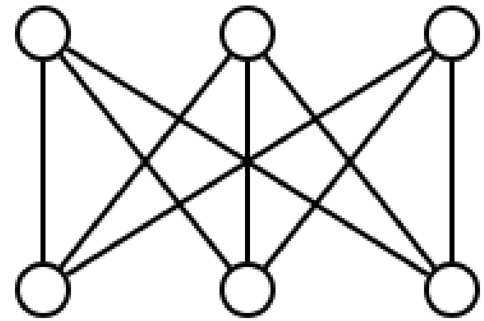
Non-planar representation of the graph

Planar representation of the graph

Examples of non-planar graphs



K_5



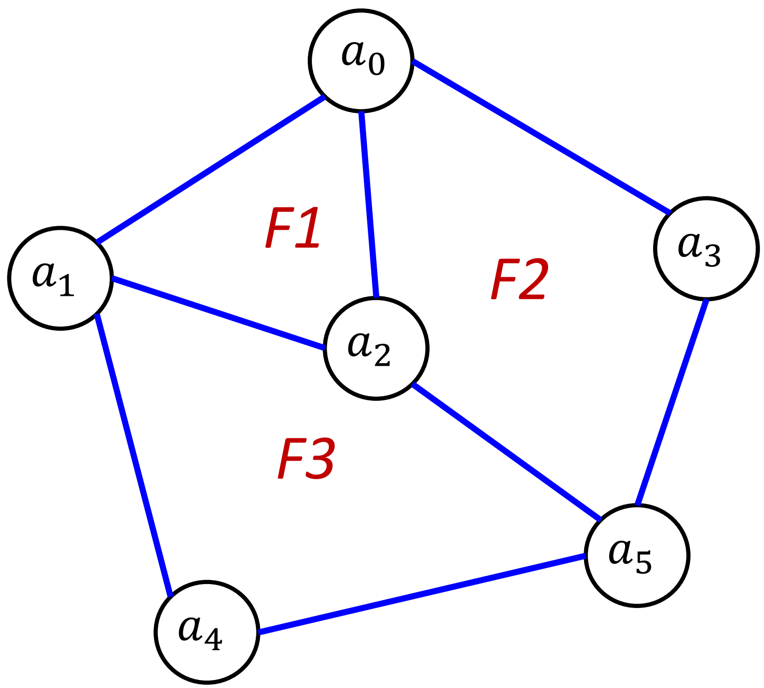
$K_{3,3}$

Kuratowski's Theorem:

A finite graph is planar if and only if it does not contain a subgraph that is a subdivision of the complete graph K_5 or the complete bipartite graph $K_{3,3}$.

Euler's Formula

When we draw a planar representation of a planar graph, it divides the plane up into regions or faces.



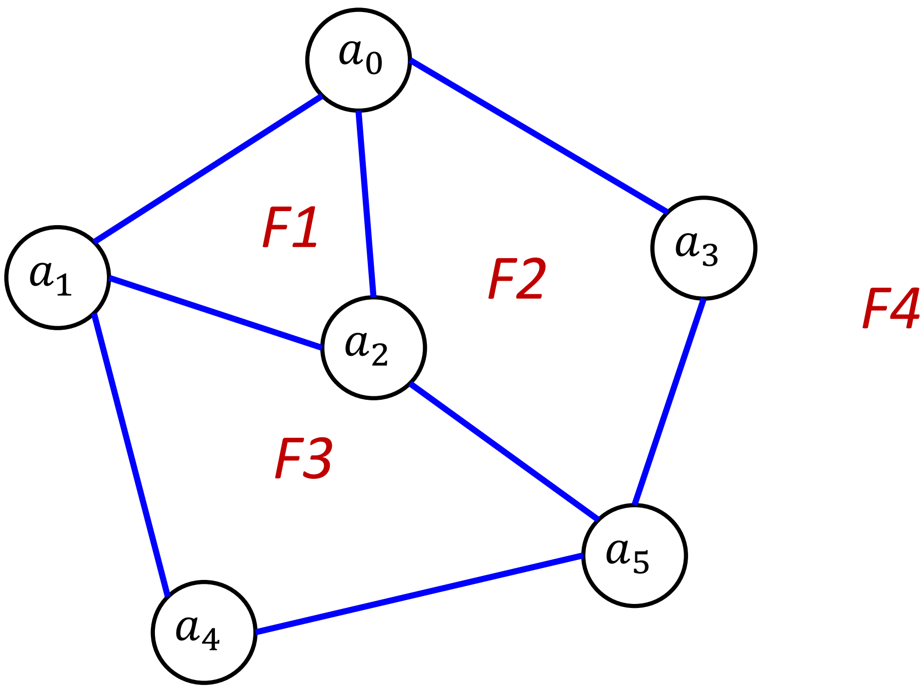
F4

(*F4* = 'outside' the planar graph)

Euler's Formula

For a connected planar simple graph $G = (V, E)$ with $e = |E|$ and $v = |V|$, if we let f be the number of faces, then

$$f = e - v + 2$$



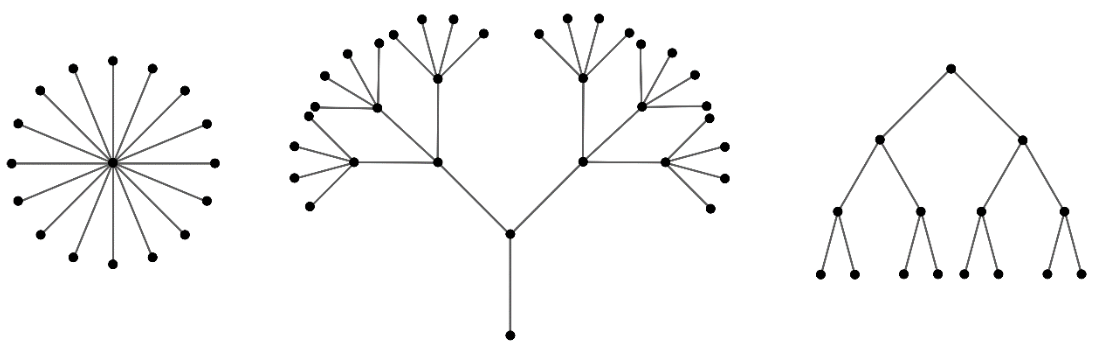
$e = 8$

$v = 6$

$f = 8 - 6 + 2 = 4$

Next week's lectures

Trees



END OF FILE