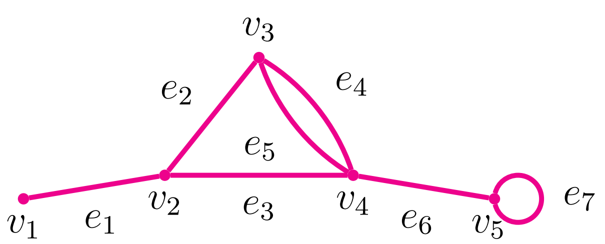
§5 Graph Theory

- Loosely speaking, a *graph* is a set of dots and dot-connecting lines.
  - The dots are called *vertices* and the lines are called *edges*.
- Formally, a *(finite) graph* G consists of
  - A finite set V whose elements are called the *vertices* of G;
  - A finite set E whose elements are called the *edges* of G;
  - A function that assigns to each edge $e \in E$ an *unordered* pair of vertices called the *endpoints* of e.
    This function is called the *edge-endpoint function*.
- Note that these graphs are **not** related to graphs of functions.
- Graphs can be used as mathematical models for networks such roads, airline routes, electrical systems, social networks, biological systems and so on.
- *Graph theory* is the study of graphs as mathematical objects.

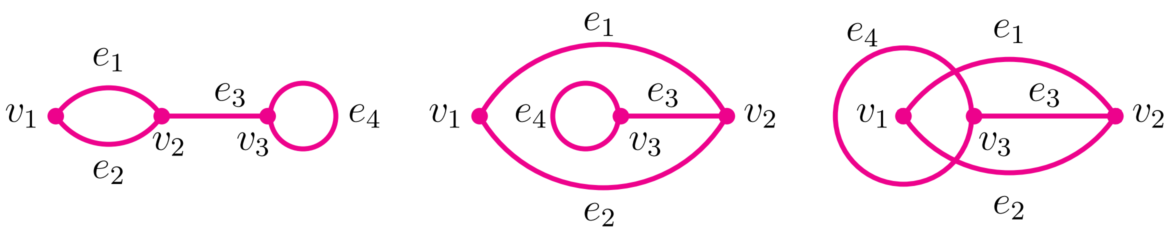
Example. Consider the following graph G with vertices and edges

$$V = \{v_1, v_2, v_3, v_4, v_5\} \quad \text{and} \quad E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\} \;:$$



Edge	Endpoints
e_1	$\{v_1, v_2\}$
e_2	$\{v_2, v_3\}$
e_3	$\{v_2, v_4\}$
e_4	$\{v_3, v_4\}$
e_5	$\{v_3, v_4\}$
e_6	$\{v_4, v_5\}$
e_7	$\{v_5\}$

Example. Below are 3 different pictorial representations of another graph. This is the same graph in 3 different layouts.

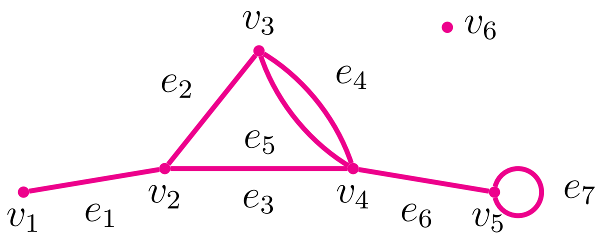


The edge-endpoint function of this graph is the same for each representation and is as follows:

Edge	Endpoints
e_1	
e_2	
e_3	
e_4	

- If the edge $e \in E$ has endpoints $v, w \in V$, then we say that
  - the edge e *connects* the vertices v and w;
  - the edge e is *incident* with the vertices v and w;
  - the vertices v and w are the *endpoints* of the edge e;
  - the vertices v and w are *adjacent*;
  - the vertices v and w are *neighbours*.
- Two edges with the same endpoints are *multiple* or *parallel*.
- A *loop* is an edge that connects a vertex to the same vertex.
- The *degree* of a vertex v, denoted by $\deg(v)$,
  is the number of edges incident with v, counting any loops twice.
- An *isolated vertex* is one with degree 0,
  and a *pendant vertex* is one with degree 1.

Exercise.



Vertex	Degree
v_1	
v_2	
v_3	
v_4	
v_5	
v_6	

In the diagram,

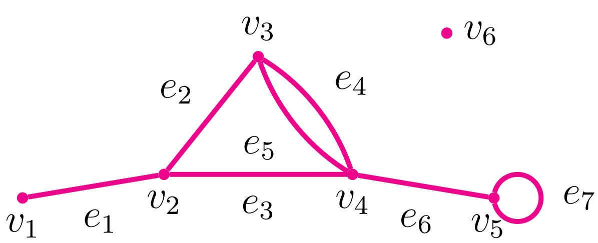
- e_3 connects vertices
- v_2 and v_3 are
- e_7 is a
- e_4 and e_5 are
- v_1 is a
- v_6 is an

- **The Handshaking Theorem.**
  The total degree of a graph is twice the number of edges:
  $$2|E| = \sum_{v \in V} \deg(v)\,.$$

Proof.

Each edge has two endpoints and must contribute 2 to the sum of degrees, which is why we count a loop twice.

Example.



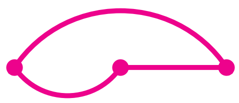
$$\begin{aligned} 2|E| &= 2 \cdot 7 = 14 \\ \sum_{v \in V} \deg(v) &= 1 + 3 + 3 + 4 + 3 + 0 = 14 \end{aligned}$$

- By the Handshaking Theorem, the total degree of a graph must be even and the number of odd vertices must be even.

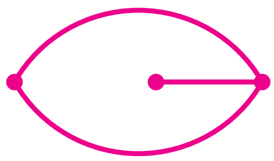
Example. No graph can have vertex degrees 3,3,3,2,2.
Why?

- A *simple graph* is a graph with **no** loops or parallel edges.

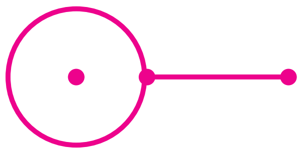
Example.



A simple graph



Not a simple graph



Not a simple graph

- Note: each vertex in a simple graph on n vertices has degree at most $n-1$.

Why? Let v be a vertex. There is no loop at v.

No parallel edges $\Longrightarrow$ at most 1 edge connects v to each of the other $n-1$ vertices.

In total, there are at most $n-1$ edges incident on v.

Exercise. Prove that no simple graph can have the following vertex degrees:

- 5,4,3,2,2;
- 4,3,3,1,1.

Answer Proof by contradiction. Suppose there is a simple graph with vertex degrees $4, 3, 3, 1, 1$. Label the corresponding vertices v_1, v_2, v_3, v_4, v_5.

$\deg(v_1) = 4 \Longrightarrow v_1$ is adjacent to all the other 4 vertices.

$\deg(v_2) = 3 \Longrightarrow v_2$ is adjacent to either v_4 or v_5. Without loss of generality, suppose there's an edge connecting v_2 to v_4.

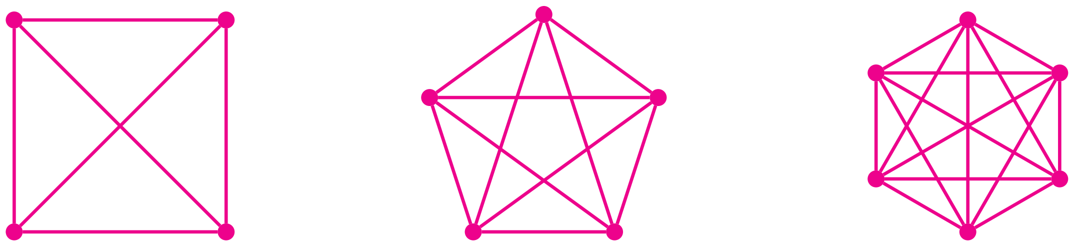
Then v_4

SOME NAMED GRAPHS

1. The complete graph K_n ($n \geq 1$) is a simple graph with
   - n vertices;
   - exactly one edge between each pair of distinct vertices.

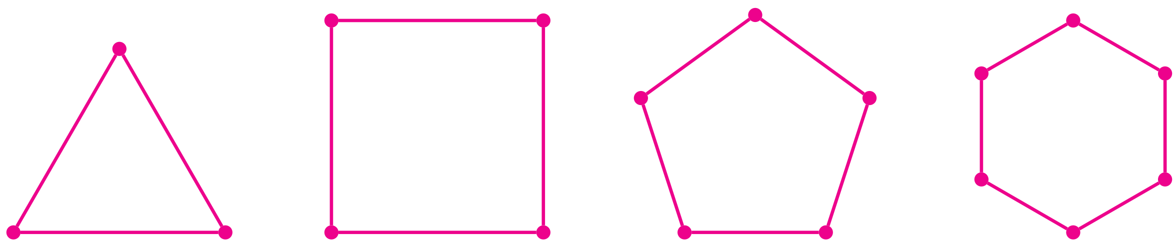
Hence K_n has $C(n, 2)$ edges.

K_1 K_2 K_3



K_4 K_5 K_6

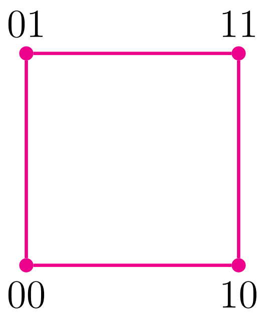
2. The cyclic graph C_n ($n \geq 3$) consists of
   - n vertices $v_1, v_2, \ldots, v_n$;
   - n edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.



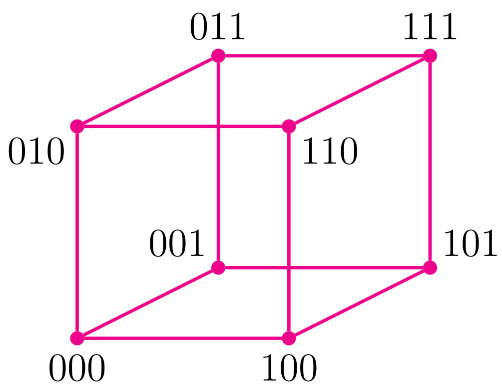
C_3 C_4 C_5 C_6

3. The n-cube Q_n is the simple graph with
   - vertices for each bit string $a_1 a_2 \cdots a_n$ of length n, where $a_i \in \{0, 1\}$;
   - an edge between vertices $a_1 a_2 \cdots a_n$ and $b_1 b_2 \cdots b_n$ if and only if $a_j \neq b_j$ for exactly one $j \in \{1, \ldots, n\}$.

     $\star$ Two vertices are adjacent if and only if they differ by one bit.



Q_2

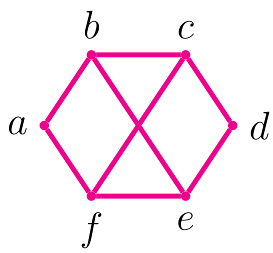


Q_3

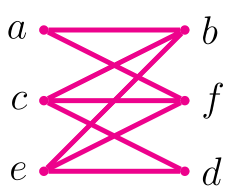
BIPARTITE GRAPHS

- A graph is *bipartite* iff its vertex set V can be partitioned into subsets V_1, V_2 so that every edge has an endpoint in V_1 and an endpoint in V_2.
  - In other words, no vertex is adjacent to any vertex in the same subset.

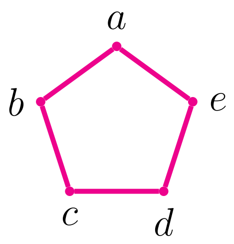
Example. Is the following graph bipartite?



Yes: let $V_1 = \{a, c, e\}$ and $V_2 = \{b, d, f\}$;
then each edge has a vertex in V_1 and a vertex in V_2,
as we can see by redrawing the graph, for instance as follows:



Example. Is the cycle C_5 bipartite?



No.

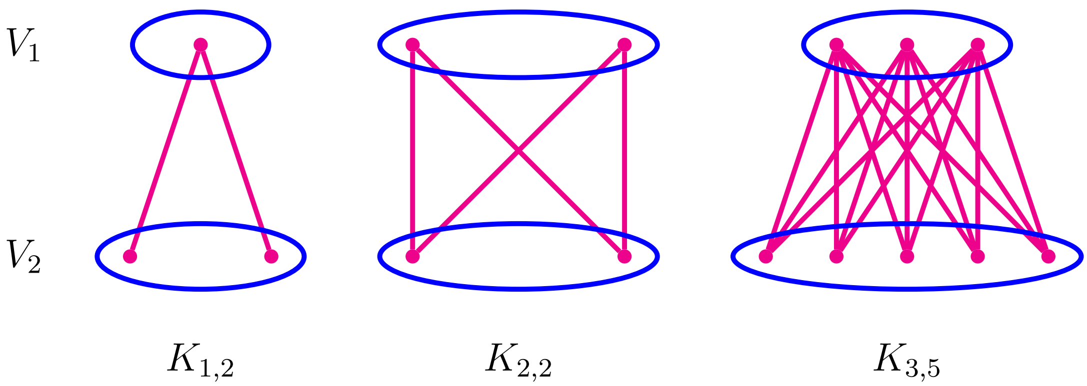
We give a proof by contradiction:
Suppose that C_5 is bipartite with vertices partitioned into subsets V_1 and V_2.
Let $a \in V_1$. Since b and e are adjacent to a, it follows that $b, e \in V_2$.
Similarly, c is adjacent to b, so $c \in V_1$, and d is adjacent to e, so $d \in V_1$.
But then both c and d are in V_1 despite being adjacent;
this contradicts the definition of a bipartite graph.
Therefore, C_5 is not bipartite.

A similar proof shows C_n is not bipartite whenever n is odd.

- The *complete bipartite graph* $K_{m,n}$ is the simple bipartite graph with
  - V_1 containing m vertices and
    V_2 containing n vertices;
  - edges between **every** vertex in V_1 and **every** vertex in V_2.

  $K_{m,n}$ has $m + n$ vertices and mn edges.

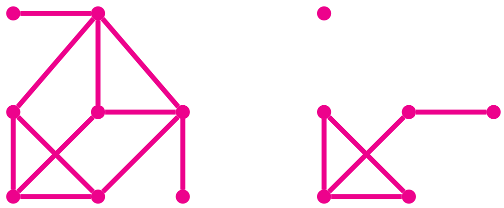
Example.



- Let G_1 and G_2 be two graphs
  with vertex sets V_1 and V_2 and edge sets E_1 and E_2, respectively.
  Then G_1 is a *subgraph* of G_2, and we write $G_1 \subseteq G_2$, iff
  - $V_1 \subseteq V_2$;
  - $E_1 \subseteq E_2$;
  - each edge in G_1 has the same endpoints as in G_2.

  Pictorially, a graph obtained by deleting somes edges and/or vertices is a subgraph. If a vertex is deleted, then so must all edges incident with it.

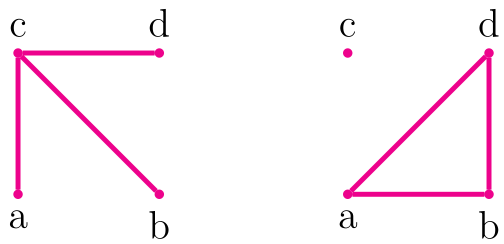
Example. $G_1 \subseteq G_2$.



G_2 $\qquad$ G_1

- Let G be a simple graph.
  The *complementary graph* $\overline{G}$ of G is a simple graph with
  - the same vertex set as G;
  - an edge joining two vertices if and only if they are **not** adjacent in G.

Example.



G $\qquad$ $\overline{G}$

Problem Set 5, Problem 10.
If a simple graph G has n vertices and m edges, then how many edges does $\overline{G}$ have?

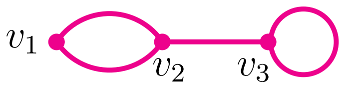
ADJACENCY MATRIX

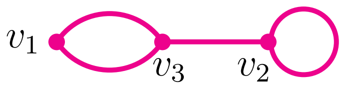
- Let G be a graph with an ordered listing of vertices $v_1, v_2, \ldots, v_n$.
  The *adjacency matrix* of G is the $n \times n$ matrix $A = [a_{ij}]$ with

  $a_{ij} = \#$ edges connecting v_i and v_j.

- The entries a_{ij} depend on the order in which the vertices have been numbered.
  - Changing the vertex order corresponds to permuting rows and columns.
- The adjacency matrix A is symmetric, i.e., $A = A^T$.

Example.





$$A = \begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 2 \\ 0 & 1 & 1 \\ 2 & 1 & 0 \end{bmatrix}$$

Exercise. Is the graph G with adjacency matrix A below simple?

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Answer. 0's on the diagonal of $A \Longrightarrow$

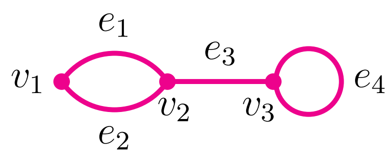
All off-diagonal entries $\leq 1 \Longrightarrow$

- A *walk* in a graph G is an alternating sequence of vertices v_i and edges e_i in G

$$v_0 e_1 v_1 e_2 v_2 \ldots v_{n-1} e_n v_n$$

  where v_{i-1} and v_i are the endpoints of edge e_i for all i.
- The *length* of the walk is the number of edges involved (n above).
- A *closed* walk is one that starts and ends in the same vertex.

Example.

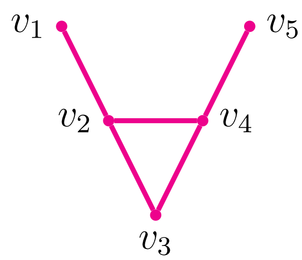


$v_1 e_1 v_2 e_3 v_3 e_4 v_3$ is a walk of length 3 from v_1 to v_3.

$v_1 e_1 v_2 e_2 v_1$ is a closed walk of length 2 from v_1 to v_1.

- In a *simple graph*, a walk can be specified by stating the vertices alone.
- A *path* is a walk with **no** repeated edges.
- A *circuit* is a path whose first and last vertices are the same (a closed path).

Example. Consider a simple graph



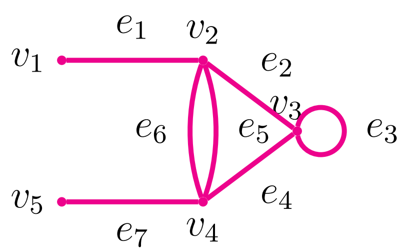
$v_1 v_2 v_4 v_3 v_2$ is a walk of length 4 from v_1 to v_2. It is also a path.

$v_1 v_2 v_4 v_3 v_2 v_4 v_5$ is a walk but not a path because edge $v_2 v_4$ is repeated.

$v_2 v_3 v_4 v_2$ is a circuit.

- A path $v_0e_1v_1e_2v_2\ldots v_{n-1}e_nv_n$ is *simple* iff all the v_i are distinct i.e. there are no repeated vertices.
- A circuit $v_0e_1v_1e_2v_2\ldots v_{n-1}e_nv_n$ is *simple* iff $v_1,\ldots,v_n$ are distinct (but $v_0 = v_n$ of course).

Example.



$v_5e_7v_4e_6v_2e_2v_3e_4v_4e_5v_2$ is a path of length 5 from v_5 to v_2. It's not simple.

$v_2e_6v_4e_4v_3e_2v_2$ is a simple circuit of length 3.

- **Theorem.** Let a and b be vertices in a graph.
If there is a walk from a to b, then there is a simple path from a to b.

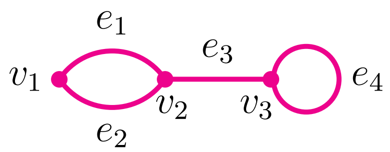
Proof. Suppose that there is at least one walk from a to b.
Then there is a shortest walk length from a to b.
Let $W = v_0e_1v_1\cdots v_n$ be a walk from a to b with this shortest length.
Suppose that W is not a simple path.
Then some vertex occurs twice, say $v_i = v_j$ for some $i < j$.
By removing the walk $v_ie_{i+1}\cdots v_j$ from W, we get the walk

$$v_0e_1v_1\cdots e_iv_ie_{j+1}v_{j+1}\cdots v_n\,.$$

This is a walk from a to b with shorter length than W.
But W had the shortest length, so we have a contradiction.
Therefore, W must be a simple path.

- **Theorem.** If A is the adjacency matrix for G with ordered vertices $v_1,\ldots,v_n$, then the number of walks of length k from v_i to v_j in G is given by the entry in the ith row and jth column of A^k.

Example.



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

From A^2 we can see that there are

- 2 walks of length 2 from v_1 to v_3, namely

  ($v_1 e_1 v_2 e_3 v_3$ and $v_1 e_2 v_2 e_3 v_3$);
- 4 walks of length 2 from v_1 to v_1, namely

Proof. We will use induction:
Let $a_{ij}^{(k)}$ be the entry in the ith row and jth column of A^k.

- Basis step: $a_{ij}^{(1)} = a_{ij}$ is the number of edges with endpoints v_i and v_j; that is, $a_{ij}^{(1)}$ is the number of walks of length 1 from v_i to v_j.
  Hence, the theorem is true for $k = 1$.

- Assume that the theorem is true for some $k \geq 1$.

- Then, since $A^{k+1} = A^k A$,

$$\begin{aligned} a_{ij}^{(k+1)} = \sum_{l=1}^{n} a_{il}^{(k)} a_{lj}^{(1)} &= \sum_{l=1}^{n} (\text{\# of walks of length } k \text{ from } v_i \text{ to } v_l) \\ &\quad \times (\text{\# of walks of length 1 from } v_l \text{ to } v_j) \\ &= \sum_{l=1}^{n} (\text{\# of walks of length } k+1 \text{ from } v_i \text{ to } v_j \text{ via } v_l) \\ &= \text{\# of walks of length } k+1 \text{ from } v_i \text{ to } v_j \end{aligned}$$

Thus, the theorem is true for $k+1$, so by induction, the theorem is true for all k.

Example.
Given two different vertices x, y in the complete graph K_{100}, how many walks of length n are there from x to y?

Answer
Let v be the second last vertex in a walk from x to y.
Since K_{100} has no loops, v cannot equal y.
However, once the second last vertex is known, there is only one way to take the last step to y.
So, to count the walks of length n from x to y we need to count the total number of walks of length $n-1$ from x to all vertices $v \neq y$.

If v is different from x we can do this recursively.

If $v = x$ we cannot; but then the third last vertex u cannot be x, and we count walks from x to u to x to y.

Let a_n be the required number of walks.

First we shall count walks $x \to \cdots \to v \to y$ with $v \neq x, y$.

1. Choose v in .... 98 ways
2. Choose a walk of length $n-1$ from x to v in .... a_{n-1} ways

Next count walks $x \to \cdots \to u \to x \to y$ with $u \neq x$.

1. Choose u in .... 99 ways
2. Choose a walk of length $n-2$ from x to u in .... a_{n-2} ways

Putting all this together, we obtain the recurrence

$$a_n = 98a_{n-1} + 99a_{n-2} \ .$$

As this is of second order we require two initial conditions:

$$a_1 = 1 \quad \text{and} \quad a_2 = 98 \ .$$

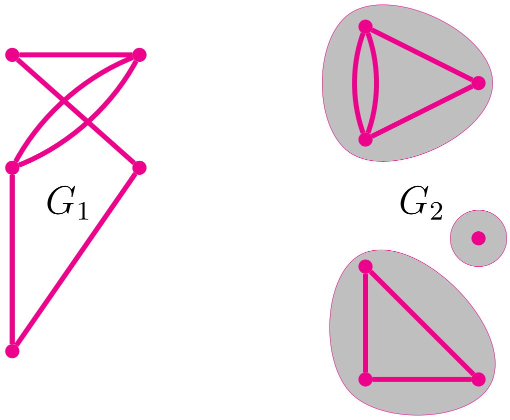
Exercise.

- Solve this recurrence to show that
$$a_n = \frac{99^n - (-1)^n}{100} .$$
- Given two different vertices x and y in K_m, where m is a fixed number, find by similar methods the number of walks of length n from x to y.
- Given a vertex x in K_m, how many walks of length n are there starting at x and finishing at x?

CONNECTIVITY

- Vertices a, b of a graph G are *connected* in G iff there is a walk from a to b.
- A graph G is *connected* iff every pair of distinct vertices is connected in G.
- Let G be a graph with vertex set V.
  The relation $\sim$ on V defined by
  $$v_i \sim v_j \quad \text{if and only if} \quad v_i \text{ is connected to } v_j \text{ in } G$$
  is an equivalence relation.
- The equivalence classes of this relation are the *connected components of G*.
  Two vertices are in the same connected component
  if and only if they are connected in G.

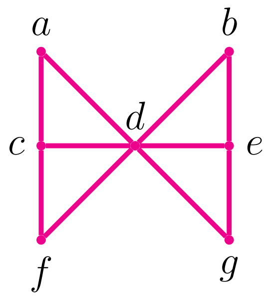
Example. The graph G_1 below is connected, but G_2 is disconnected. The 3 connected components of G_2 have been shaded.



EULER & HAMILTON PATHS & CIRCUITS

- Let G be a graph.
- An *Euler path* in G is a path that includes every edge of G exactly once.
- A *Hamilton path* in G is a path that includes every vertex of G exactly once.
- An *Euler/Hamilton circuit* in G is an Euler/Hamilton path that is a circuit.

Example.



$cfdacdgebde$ is an Euler path;
$acfdbeg$ is a Hamilton path.

Example. A network of roads is to be snow-ploughed. An Euler circuit will ensure that all roads get ploughed without going over a road already cleared.

Example. Similarly for a postman delivering mail.

Example. A salesperson wants to visit some towns using a network of roads. He wants a Hamilton circuit so that each town is visited without backtracking through a town already visited.

- **Theorem.** Let G be a connected graph.

  An Euler circuit exists if and only if G has even vertex degrees i.e. there are no vertices with odd degree.

Proof. ($\Longrightarrow$).
Suppose that an Euler circuit C exists in G.
Each time C passes through a vertex v, it uses up 2 distinct edges, one in and one out.
Every edge is used exactly once, so $\deg(v)$ is twice the number of times C passes through v.
Therefore, $\deg(v)$ is even.

($\Longleftarrow$). Conversely, suppose that G has no vertex of odd degree. We need

- **Lemma** Let G' be a graph with even vertex degrees. Then any path $v_0e_1v_1 \ldots e_nv_n$ with $v_0 \neq v_n$ can be extended to a circuit $v_0e_1v_1 \ldots e_nv_ne_{n+1} \ldots v_m$ with $v_m = v_0$.

Why? Path $v_0e_1v_1 \ldots e_nv_n$ uses up an odd number of edges incident on v_n.
$\deg(v_n)$ even $\Longrightarrow$ can find unused edge e_{n+1} and

extend path to $v_0e_1v_1 \ldots e_nv_ne_{n+1}v_{n+1}$.
We can continue inductively until the circuit is found.

We now return to the proof of ($\Longleftarrow$) and use the following algorithm to produce an Euler circuit C:

Euler Circuit

Choose a vertex v_0 in G. Start with length 0 circuit $C := v_0$ which we extend to an Euler circuit as follows.

while there exists an edge not in C do

- choose a vertex $v \in C$ and an edge $e \notin C$ that is incident with v;
(possible because G is connected).

- choose a circuit C' starting and finishing at v

  that contains e but not any edge in C; (possible by applying Lemma to subgraph G' of G obtained by deleting all the edges of C)

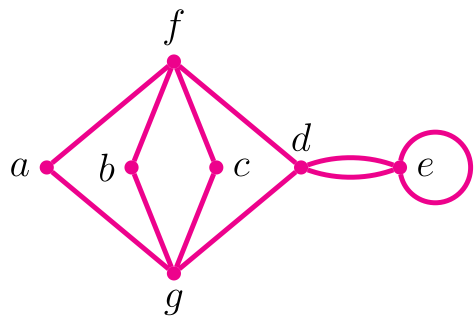
- Replace one of the 'v's in C by C';

end do

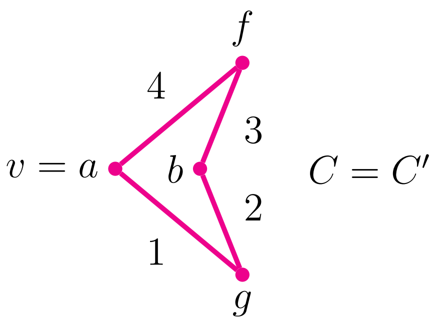
The algorithm terminates because G has only finitely many edges, and every step adds at least one edge to C.
By construction, C is an Euler circuit.

This proves the theorem.

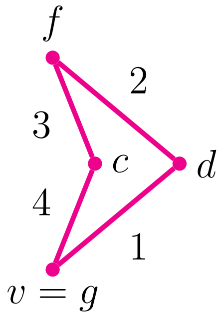
Example. Apply **Euler Circuit** to construct an Euler circuit below:



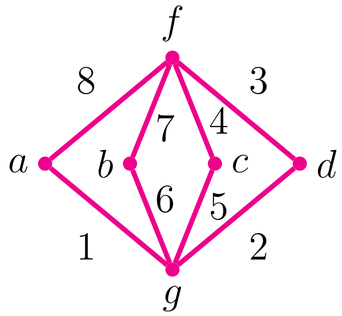
First note that the degree of each vertex is even.
If we initially take $v_0 := a$, then at the first step of the algorithm, one possible choice for C' is as follows:



At the second step, we may take $v := g$ and choose C' as follows:

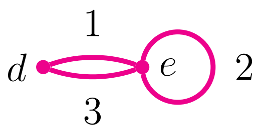


C'

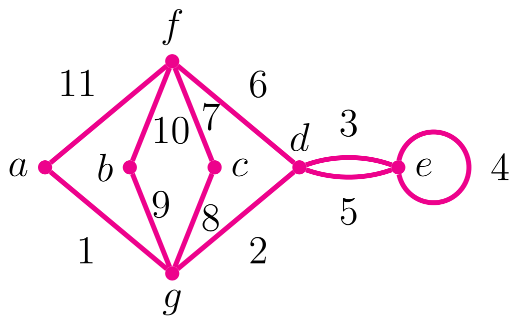


C

At the third step, we must take $v := d$, and can choose C' as follows:



C'



C

We now have an Euler path C.

Theorem. Let G be a connected graph.

An Euler path which is not a circuit exists if and only if G has exactly two vertices of odd degree.

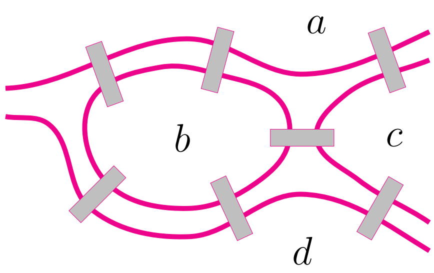
Proof.

($\Longrightarrow$) Assume that G has an Euler path $v_0 e_1 v_1 \ldots e_n v_n$ (with $v_0 \neq v_n$), and consider the graph G' formed from G by adding a new edge e' that connects v_0 and v_n.
Since $v_0 e_1 v_1 \ldots e_n v_n e' v_0$ is an Euler circuit in G',
we know that each vertex in G' has even degree.
Hence, G has exactly two vertices of odd degree, namely, v_0 and v_n.

($\Longleftarrow$) Conversely, suppose that G has exactly two vertices of odd degree, say a and b.
We form G' by connecting a and b with a new edge e',
so that every vertex in G' has even degree.
Hence, G' has an Euler circuit.
Removing the new edge e' from G' again gives an Euler path for G.

The Königsberg Bridge Problem.

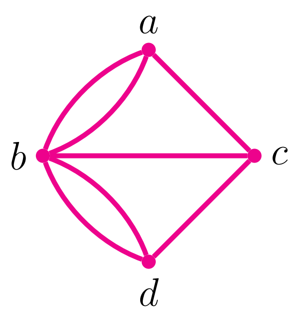
In the 18 century, Kaliningrad in the Russian Republic was called Königsberg, and was part of Prussia. At that time, seven bridges connected the four different parts of the town, as depicted in the diagram below.



The question arose as to whether it was possible to start from one part of the town, cross every bridge exactly once, and return to the starting point.

In the first ever paper on graph theory, in 1736, Leonhard Euler explained why this could not be done and proved the theorems about when an Euler circuit or Euler path exist.

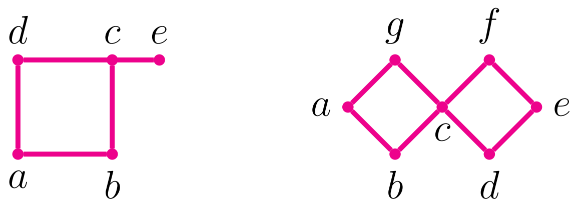
We now know that the problem is equivalent to finding an Euler circuit in the following graph.



Here, $\deg(a) = 3$, $\deg(b) = 5$, $\deg(c) = 3$, and $\deg(d) = 3$,
so there are 4 vertices with odd degree, and hence no Euler circuit (or path) exists.

- No simple necessary and sufficient criteria are known that determine whether a graph has a Hamilton circuit or path.
- Note that a graph with a vertex of degree 1 cannot have a Hamilton circuit.
- If a graph G has a Hamilton circuit,
  then the circuit must include all edges incident with vertices of degree 2.
- A Hamilton path or circuit uses at most 2 edges incident with any one vertex.

Example. The following graphs have no Hamilton circuits:



To see that the second graph has no Hamilton circuit,
note that b, d, f, g each have degree 2,
so any Hamilton circuit would have to use all edges incident with these vertices,
including the four edges incident with c.

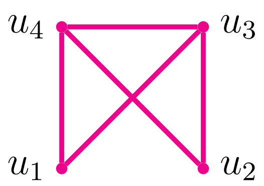
- **Theorem.** (Dirac 1952)
  If G is a connected and simple graph with $n \geq 3$ vertices,
  and each vertex has degree at least $n/2$, then G has a Hamilton circuit.
- **Example.** The complete graph K_n is connected and simple, and has n vertices.
  Each vertex is adjacent to every other vertex, so each vertex has degree $n-1$.
  By the theorem, K_n has a Hamilton circuit when $n \geq 3$.

- The above theorem does **not** give a *necessary* condition.
  Some graphs have Hamilton circuits but do not satisfy the theorem's condition.
- **Example.** The cyclic graph C_n has a Hamilton circuit for all $n \geq 3$.
  Each vertex has degree 2 which is smaller than $n/2$ when $n \geq 5$.

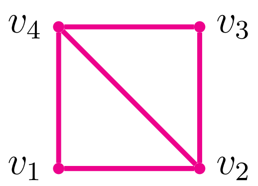
- **Theorem.** (Ore 1960)
  If G is a simple connected graph with $n \geq 3$ vertices and
  for every pair v_1 and v_2 of non-adjacent vertices $\deg(v_1) + \deg(v_2) \geq n$,
  then g has a Hamilton circuit.

- Let G and G' be graphs with vertices V, resp. V', and edges E, resp. E'. Then G is *isomorphic* to G', and we write $G \simeq G'$, iff there are two bijections $f : V \to V'$ and $g : E \to E'$, such that e is incident with v in G if and only if $g(e)$ is incident with $f(v)$ in G'.
  - Roughly speaking, two graphs are *isomorphic* iff they are the same except for edge and vertex labelings.
  - In this case, $\deg(v) = \deg(f(v))$.
- Two simple graphs G and G' are isomorphic iff there is a bijection $f : V \to V'$ such that for all $v_1, v_2 \in V$, v_1 and v_2 are adjacent in G if and only if $f(v_1)$ and $f(v_2)$ are adjacent in G'.

Exercise. Are the following simple graphs isomorphic?



G



G'

v	$f(v)$
u_1	v_1
u_2	v_3
u_3	v_4
u_4	v_2

Yes; an isomorphism is given by the table.
Note that u_i and u_j are adjacent if and only if $f(u_i)$ and $f(u_j)$ are adjacent.

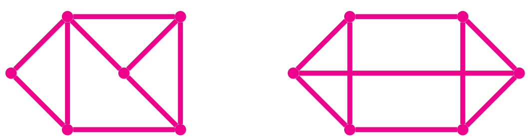
- A property of a graph G is an *invariant* iff G' also has this property whenever $G' \simeq G$.

Example. Some graph invariants are

- the number of vertices;
- the number of edges;
- the total degree;
- the number of vertices of a given degree;
- bipartiteness, number of connected components, connectedness;
- having a vertex of some degree n adjacent to a vertex of some degree m;
- the number of circuits of a given length;
- the existence of an Euler circuit;
- the existence of a Hamilton circuit.

- The easiest way to show that graphs G and G' are **not** isomorphic ($G \not\simeq G'$) is to find an invariant property that holds for G but not for G'.
- To prove that simple graphs G and G' **are** isomorphic ($G \simeq G'$), we need to find an isomorphism between them; that is, a bijection $f : V \to V'$ satisfying the condition for isomorphism.
- If G (and G') has n vertices, then there are $n!$ bijections from V to V'. If n is large, then it is very hard to find an isomorphism among all $n!$ bijections.

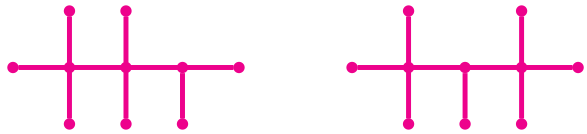
Example. Are these two simple graphs isomorphic?



G $\qquad$ G'

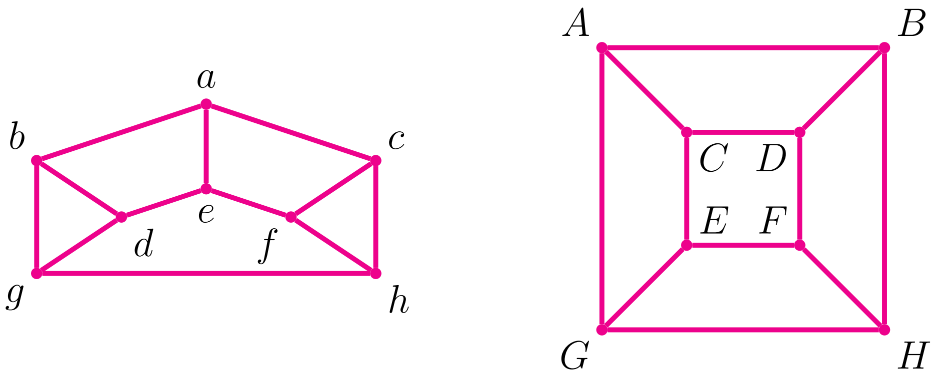
No:
G and G' are not isomorphic since G has a vertex of degree 4 but G' does not.

Exercise. Are these two simple graphs isomorphic?



G $\qquad$ G'

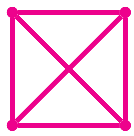
Exercise. Are these two simple graphs isomorphic?



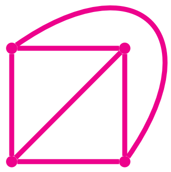
G_1 $\qquad$ G_2

- A graph G is *planar*
  iff it **can be** drawn in the plane so that no edge crosses another.
- Such a drawing is called a *planar map* or *planar representation* of G.

Example. The complete graph K_4 is planar:



not a planar map of K_4



a planar map of K_4

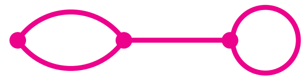
Exercise. Prove that the complete bipartite graph $K_{2,3}$ is planar.

Note. We will see later that K_5 and $K_{3,3}$ are not planar.

- A planar map divides the plane into a finite number of *regions*.
  Exactly one of these regions is unbounded.
- A planar graph can have different planar representations (or maps),
  but the *number of regions* is the **same** for all planar representations.
  This number depends only on the number of edges and vertices of the graph:
- **Euler's Formula.**
  If G is a connected planar graph with e edges and v vertices,
  and if r is the number of regions in a planar representation of G,
  then

$$v - e + r = 2\,.$$

Example. Consider the following planar map:



The map has $v = 3$ vertices, $e = 4$ edges, and $r = 3$ regions.
Therefore, $v - e + r = 2$.

Exercise. If G is a connected planar graph with 8 vertices each with degree 3 (see page 46 for some examples), then how many regions are there in a planar representation for G?

- **Lemma.** Let G be a connected graph with more than one vertex. If G has no circuit, then it has a pendant vertex.

Proof. Suppose that G has no pendant vertices.

Since G is connected and has at least two vertices, it follows that there are no isolated vertices so every vertex of G has degree at least 2.
Choose a vertex v_0 and find a simple path $v_0 e_1 v_1 \cdots e_n v_n$ that cannot be extended to a longer simple path.
Since $\deg(v_n) \geq 2$, there is an edge $e_{n+1} \neq e_n$ incident with v_n.
Let v_{n+1} be the other endpoint of e_{n+1}.
The walk $v_0 e_1 v_1 \ldots e_n v_n e_{n+1} v_{n+1}$ cannot be a simple path, so $v_{n+1} = v_i$ for some $i \in \{1, \ldots, n\}$.
But then $v_i e_{i+1} v_{i+1} \ldots e_{n+1} v_{n+1}$ is a circuit.

Proof of Euler's Formula. We will use (informal) induction on e:

- If $e = 0$, then $v = 1$ and $r = 1$ because the map is connected.
  Thus, $v - e + r = 1 - 0 + 1 = 2$ as required.

- Assume for some $e \geq 0$ that the formula holds for all connected planar map with e edges. Let G be a connected planar map with $e+1$ edges, v vertices, and r regions.
  We must prove that $v - (e+1) + r = 2$.

  If G has no circuits, then by the lemma there exists a pendant vertex.
  Deleting this vertex and the edge incident with it,
  we obtain a connected planar map G' with e edges and $v-1$ vertices.
  Note that G' also divides the plane into r regions.
  Hence, by the inductive assumption, $(v-1)-e+r = 2$, so $v-(e+1)+r = 2$.

  If, on the other hand, G has a circuit,
  then delete an edge from that circuit to get a connected planar map G' with v vertices and e edges.
  Note that G' divides the plane into $r-1$ regions.
  So, by the inductive assumption, $v-e+(r-1) = 2$, so $v-(e+1)+r = 2$.

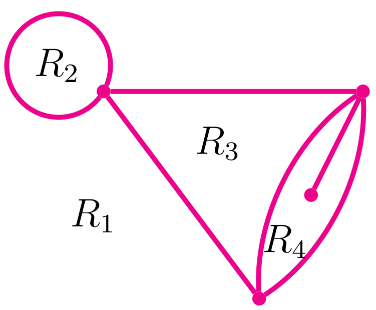
By induction, Euler's Formula holds for all connected planar maps.

- The *degree* of a region R in a planar representation is the number of edges (counting repetitions) traversed in going round the boundary of R.
- In summing all region degrees, each edge contributes twice, so

$$\boxed{2|E| = \text{the sum of region degrees.}}$$

- By the Handshake Lemma, it follows that
  - the sum of region degrees equals the sum of vertex degrees.

Example.



$\deg(R_1) = 4$

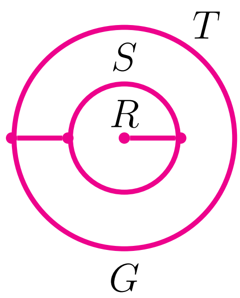
$\deg(R_2) = 1$

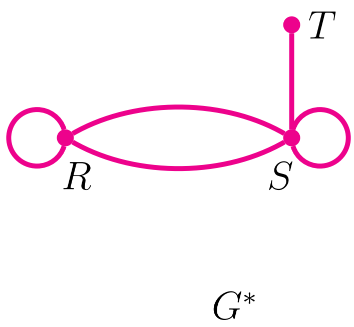
$\deg(R_3) = 3$

$\deg(R_4) = 4$

- The *dual* of a planar map G is a planar map G^* given as follows:
  - for each region R_i of G, there is an associated vertex v_i^* in G^*;
  - for each edge e in G that is surrounded by one region R_i, there is an associated loop in G^* at vertex v_i^*.
  - for each edge e of G that separates two regions R_1 and R_2, there is an edge e^* in G^* that connects vertices v_1^* and v_2^* corresponding to R_1 and R_2, respectively;

Example.





- If G is a simple connected planar graph with at least 3 vertices, then every region degree is at least 3.
  - To have a region of degree 1, G must have a loop.
  - To have a region of degree 2, G must have parallel edges.
- **Theorem.**
  If G is a connected planar simple graph with e edges and $v \geq 3$ vertices, then
  - $e \leq 3v - 6$;
  - $e \leq 2v - 4$ if G has no circuits of length 3.
- This theorem is useful for proving that some graphs are **not** planar.
- This theorem is an example of the principle that, the more edges a graph has, the harder for it to be planar.

Proof. Let r denote the number of regions of G.

- Since G is simple, connected, and planar,
  the r regions of G each have degree at least 3.
  By Euler's Formula,

$$2e = \sum_{\text{all regions } R} \deg(R) \geq 3r = 3(e - v + 2) = 3e - 3v + 6\,,$$

  so $e \leq 3v - 6$. Also note that $e \geq \frac{3}{2}r$.

- If G has no circuits of length 3, then each region has degree at least 4.
  Thus, as above,

$$2e = \sum_{\text{all regions } R} \deg(R) \geq 4r = 4(e - v + 2)\ n = 4e - 4v + 8\,,$$

  so $e \leq 2v - 4$.

Example. Prove that the complete graph K_5 is not planar.
K_5 is connected and simple,
but $e = C(5, 2) = 10$ and $3v - 6 = 3 \times 5 - 6 = 9$.
By the theorem, K_5 is not planar.

Example. Prove that the complete bipartite graph $K_{3,3}$ is not planar.

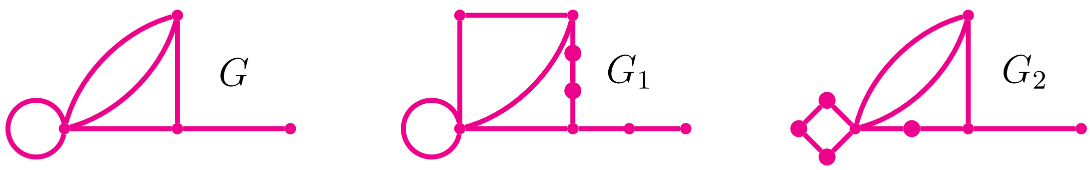
Note that a path of length 3 starting in one vertex set must end in the other.
Therefore, $K_{3,3}$ has no circuits of length 3.
Since $2v - 4 = 2 \times 6 - 4 = 8 < 9 = e$, $K_{3,3}$ cannot be planar.

Note. $3v - 6 = 3 \times 6 - 6 = 12$,
so we could not use the inequality $e \leq 3v - 6$.

- Suppose that G has an edge e with endpoints v and w.
  Let G' be the graph obtained from G by replacing e by a path $ve'v'e''w$.
- Such an operation is called an *elementary subdivision*.

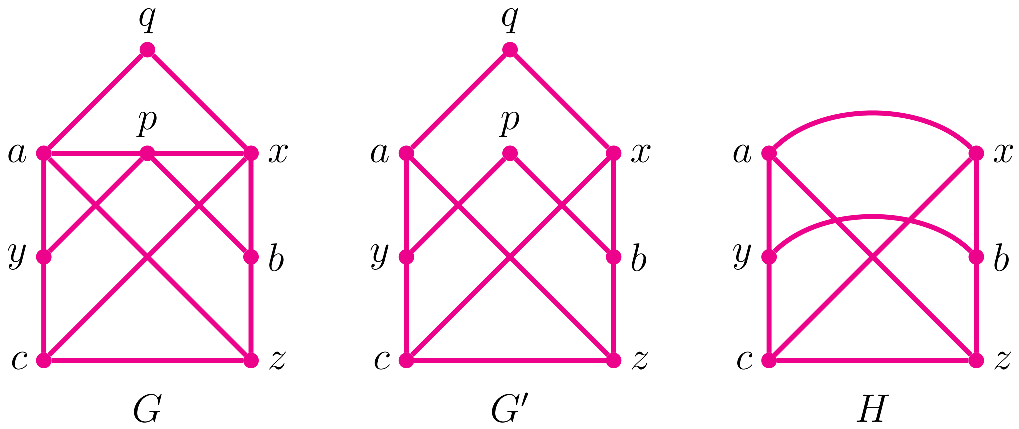
- If G is planar, then so is G'.
- Two graphs are *homeomorphic*
  iff each can be obtained from a common graph by elementary subdivisions.
  - If G is planar, then so is any graph homeomorphic to G.

Example. Consider the graphs G, G_1, and G_2 below.
Since G_1 and G_2 each are obtained from G by elementary subdivisions,
it follows that G_1 and G_2 are homeomorphic.



- **Kuratowski's Theorem.** (1930) A graph is planar if and only if
  it does not contain a subgraph homeomorphic to $K_{3,3}$ or K_5.

Example. Show that the graph G below is not planar.

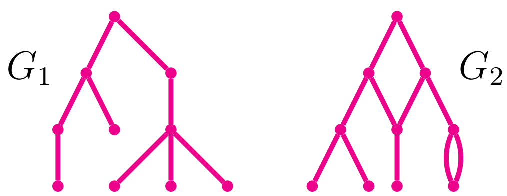


We can't use the first theorem on planarity,
because $e = 13$ and $3v - 6 = 18$, and G does have circuits of length 3.
However, we can use Kuratowski's Theorem:
the subgraph G' obtained by deleting edges ap and px is homeomorphic to H (by subdivision of edges ax and yb in H),
and H is isomorphic to $K_{3,3}$
(as can be seen by partitioning the vertices of H into sets $\{a, b, c\}$ and $\{x, y, z\}$).

TREES

- A *tree* is a connected graph with no circuits.
- It has no loops or multiple edges so is simple.

Example. The graph G_1 is a tree but G_2 is not (it has two simple circuits).



- **Theorem.** Any tree T is planar.

Why? The theorem is true if T has 1 vertex so suppose it has more than one.
It has no circuits so it has a pendant vertex v with connecting edge e, say.
By induction, we can draw a planar representation of T with v, e removed.
Hence we can also draw a planar representation of T.

- A *spanning tree* in a graph G is

  a subgraph that is a tree and contains every vertex of G.

Example. The subgraph G' below is a spanning tree for the graph G:

G $\quad$ G'

Example. Given a network of dirt roads connecting various towns, we may want to pave a minimal subset of the roads so as to ensure that every pair of towns is connected by a paved route.
To do this, we should pick a spanning tree for the network.

- **Theorem.** Every connected graph contains a spanning tree.

Proof. If a connected graph G is not itself a tree,
then remove edges from circuits until no circuit remains.
The result is a tree that contains all vertices of G; that is, a spanning tree of G.

- **Theorem.** A connected graph with n vertices is a tree
  if and only if it has exactly $n-1$ edges.

Proof. Let G be a connected graph with n vertices.
Suppose that G is a tree; then G is a planar graph with $r = 1$ region.
By Euler's Formula, $2 = v - e + r = n - e + 1$, so G has $e = n - 1$ edges.

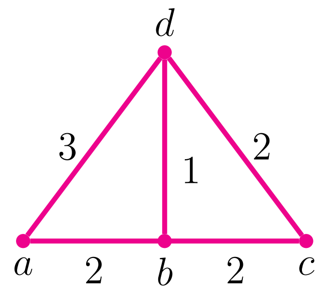
Conversely, suppose that G has $n - 1$ edges.
Since G is connected, it contains a spanning tree.
Since this tree contains n vertices, the first part of the proof implies that it has $n - 1$ edges and must in fact be the graph G.
Hence, G is a tree.

Exercise. Find an easy induction proof of this result.

Exercise.
Prove that the spanning trees of a graph all have the same number of edges.

- A *weighted graph* is a graph whose edges have been given numbers called *weights*. The *weight of an edge* e is denoted by $w(e)$.
- The *weight of a subgraph* in a weighted graph G is the sum of the weights of the edges in the subgraph.
- These numbers often represent lengths, travel time, costs, flow capacity, etc.
- A *minimal spanning tree* in a weighted graph G is a spanning tree whose weight is less than or equal to the weight of any other spanning tree.
- There can be more than one minimal spanning tree in a graph.

Example. Consider the following weighted graph.

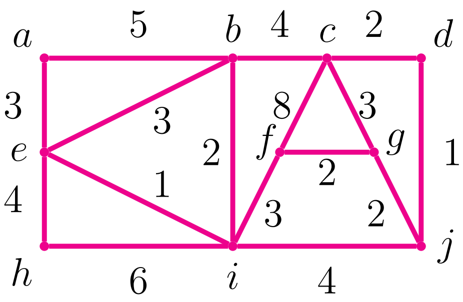


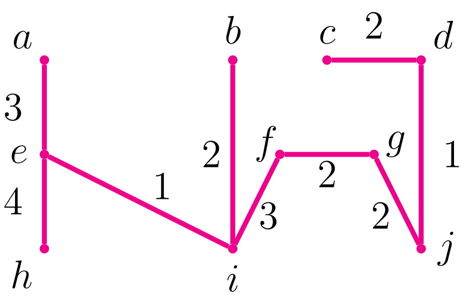
The edges da, db, and dc form a spanning tree T of total weight 6.
This tree T is not minimal
since edges ab, bd, and bc form a spanning tree T' of total weight 5.
It is not hard to see that T' is minimal.

The edges ab, bd, and dc form another minimal spanning tree T''.

- Minimal spanning trees of weighted graphs G on n vertices are found using **Kruskal's Algorithm** (1928):
  - Start with the tree $T := \varnothing$.
  - Sort the edges of G into increasing order of weight.
  - Going down the list, add an edge to T if and only if it does not form a circuit with edges already in T.
  - Continue this process until T has $n-1$ edges.
  - Then T is a minimal spanning tree for G.

Example. Find a minimal spanning tree for the following weighted graph.

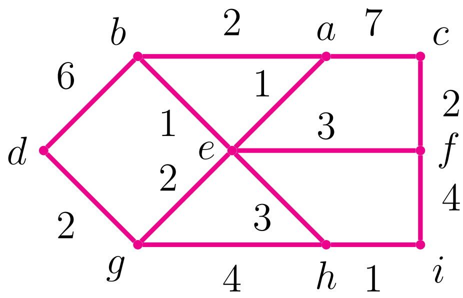




Edge	Weight	Chosen?
dj	1	Y
ei	1	Y
bi	2	Y
cd	2	Y
fg	2	Y
gj	2	Y
ae	3	Y
be	3	N
cg	3	N
fi	3	Y
bc	4	N
eh	4	Y, stop
ij	4	
ab	5	
hi	6	
cf	8	

- Given a connected weighted graph G and a particular vertex v_0, we want to find a *shortest path from v_0 to v* for each vertex v in G (here, a *shortest path* is one with minimal total weight).
- The union of these paths forms a *minimal v_0-path spanning tree* for G.
- **Dijkstra's Algorithm** (1959)
  - Start with the subgraph T consisting of v_0 only.
  - Consider all edges e with one vertex in T and the other vertex v not in T.
  - Of these edges, choose an edge e giving a shortest path from v_0 to v.
  - Add this edge e and vertex v to T.
  - Continue this process until T contains all the vertices of G.
  - Then T is a minimal v_0-path spanning tree for G.

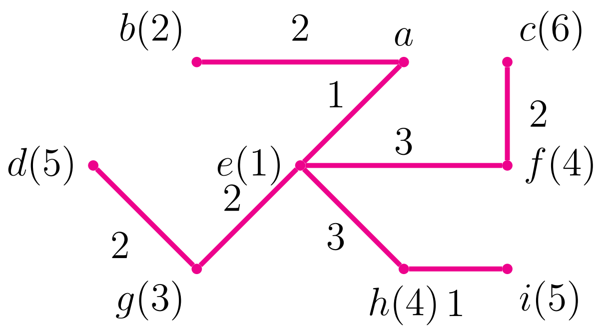
Example. Find the shortest paths from the vertex a to each of the other vertices in the weighted graph shown below (i.e., find a minimal a-path spanning tree).



There are nine vertices, so DIJKSTRA'S ALGORITHM has eight steps, given by the following table, where the total distances from a are in brackets.

Edge candidates	Next edge	Next vertex
$ab(2)$, $ae(1)$, $ac(7)$	$ae(1)$	$e(1)$
$ab(2)$, $ac(7)$, $eb(2)$, $eg(3)$, $eh(4)$, $ef(4)$	$ab(2)$	$b(2)$
$ac(7)$, $bd(8)$, $eg(3)$, $eh(4)$, $ef(4)$	$eg(3)$	$g(3)$
$ac(7)$, $bd(8)$, $eh(4)$, $ef(4)$, $gd(5)$, $gh(7)$	$eh(4)$	$h(4)$
$ac(7)$, $bd(8)$, $ef(4)$, $gd(5)$, $hi(5)$	$ef(4)$	$f(4)$
$ac(7)$, $bd(8)$, $gd(5)$, $hi(5)$, $fc(6)$, $fi(8)$	$gd(5)$	$d(5)$
$ac(7)$, $hi(5)$, $fc(6)$, $fi(8)$	$hi(5)$	$i(5)$
$ac(7)$, $fc(6)$	$fc(6)$	$c(6)$

The tree produced by the algorithm is shown below.

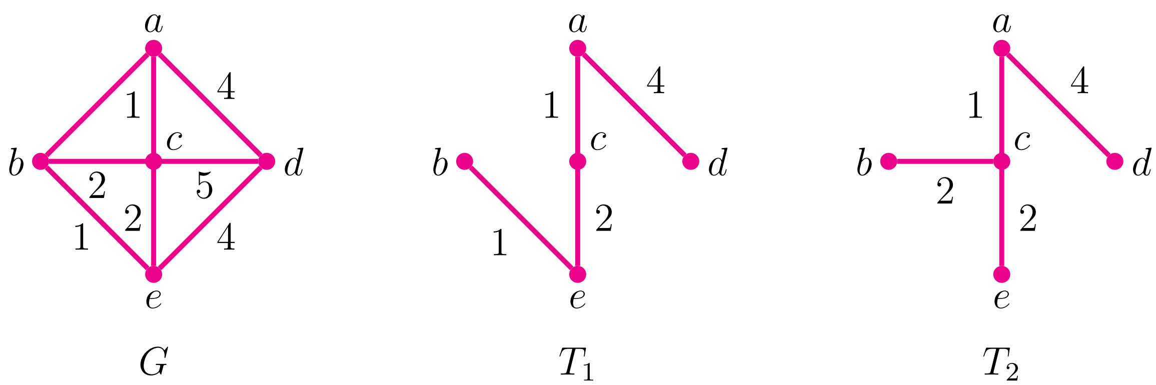


Note. At the second step, we could have chosen eb instead of ab.

- The minimal a-path spanning tree is **not** generally a minimal spanning tree.

Example.

The tree T_1 below is a minimal spanning tree in G (total weight $= 8$) whereas T_2 is a minimal a-path spanning tree in G (total weight $= 9$).

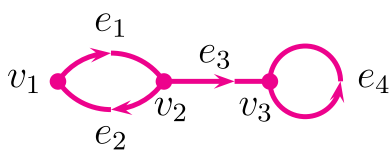


EXTRA MATERIAL NOT EXAMINED BUT IN SYLLABUS

DIRECTED GRAPHS

- Loosely speaking, a *directed graph* is a set of dots and dot-connecting arrows; in other words, a graph whose edges have been pointed one way or the other.
- Formally, a *directed graph* D consists of
  - A set V whose elements are called the *vertices* of D;
  - A set E whose elements are called the *(directed) edges* of D;
  - An *edge-endpoint function* that assigns to each (directed) edge $e \in E$ an *ordered* pair of vertices called the *endpoints* of e.
    - The first vertex of the pair is called the *initial* or *start vertex*; the second vertex is called the *final* or *finish vertex*.

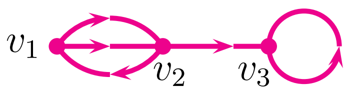
Example. Internet pages and links form the vertices and edges of a directed graph.
Exercise.



Edge	Endpoints
e_1	
e_2	
e_3	
e_4	

ADJACENCY MATRIX OF A DIRECTED GRAPH

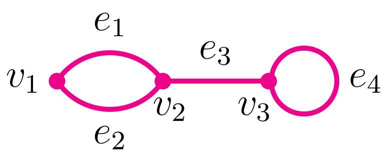
- Let G be a directed graph with an ordered listing of vertices $v_1, v_2, \ldots, v_n$. The *adjacency matrix* of G is the $n \times n$ matrix $A = [a_{ij}]$ with
  - $a_{ij} = \#$ edges with start point v_i and final point v_j,
- The entries a_{ij} depend on the order in which the vertices have been numbered.
  - Changing the vertex order corresponds to permuting rows and columns.
- If G is directed, then the adjacency matrix A need not be symmetric.



$$A = \begin{bmatrix} 0 & 2 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

- Let G be a graph with vertices $v_1, v_2, \ldots, v_n$ and edges $e_1, e_2, \ldots, e_m$. The *incidence matrix* of G is the $n \times m$ matrix $B = [b_{ij}]$ with
  - $b_{ij} = \begin{cases} 1 & \text{, if edge } e_j \text{ is incident with vertex } v_i, \\ 0 & \text{, otherwise.} \end{cases}$

Example.



$$B = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Exercise. How can you tell from the incidence matrix whether a graph has parallel edges or loops?