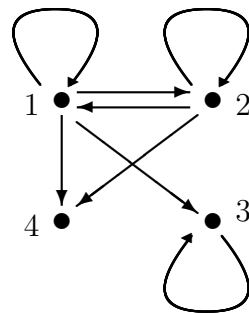
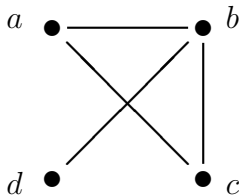
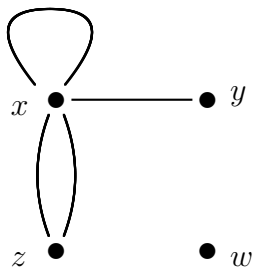


# Graphs

Graphs are drawings with dots and (not necessarily straight) lines or arrows.



The dots are called vertices (or nodes).

The lines or arrows are called edges.

## Different kinds of graphs

Type	Edges	Multiple edges	Loop edges
(simple) graph	undirected	no	no
multigraph	undirected	yes	yes
directed graph	directed	no	yes
...	...	...	...

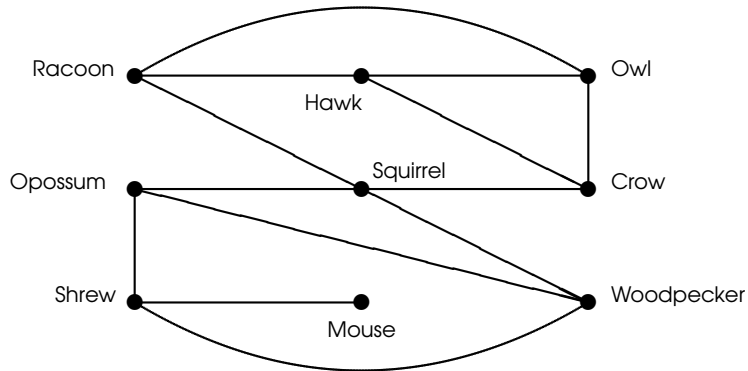
Because graphs have applications in a variety of disciplines, many different terminologies of graph theory have been introduced. You may find different ones in different books, areas, etc.

## Example 1: Niche overlap graphs in ecology

Competitions between species in an ecosystem can be modelled using

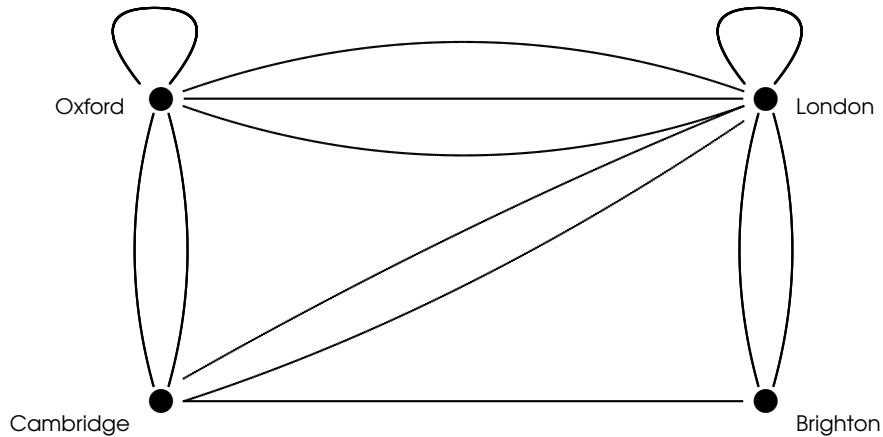
a niche overlap graph:

Each species is represented by a vertex. An edge connects two vertices if the two species represented by these vertices compete (that is, some of the food resources they use are the same).



$\leadsto$  simple graph

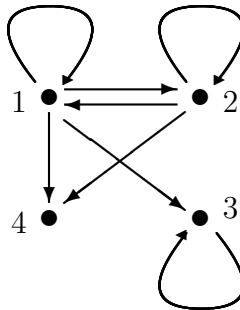
## Example 2: Road networks



$\leadsto$  multigraph

### Example 3: Representing binary relations

$$R = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 1), (2, 2), (2, 4), (3, 3)\}.$$



$\leadsto$  directed graph

## Undirected graphs: basic terminology

If there is an edge  $e$  between vertices  $u$  and  $v$ , we say that

- $u$  and  $v$  are adjacent, and
- $e$  is incident with  $u$  and  $v$ .

The degree of a vertex is the number of edges incident with it.

- A vertex of degree zero is called isolated.

So an isolated vertex is not adjacent to any vertex.

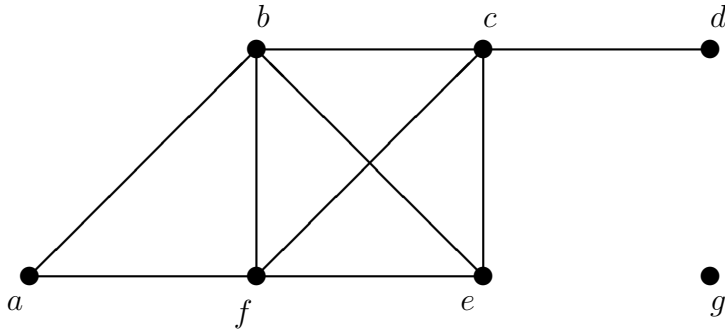
- A vertex of degree one is called pendant.

So a pendant vertex is adjacent to exactly one other vertex.

Handshaking theorem:

$$\text{number of edges} = \frac{\text{sum of the degrees of vertices}}{2}$$

## Degrees of vertices: example 1



- $\text{degree}(a) = 2$ ,
- $\text{degree}(b) = \text{degree}(c) = \text{degree}(f) = 4$ ,
- $\text{degree}(e) = 3$ ,
- $\text{degree}(d) = 1$ , so  $d$  is pendant,
- $\text{degree}(g) = 0$ , so  $g$  is isolated.

## Directed graphs: basic terminology

If there is an edge  $e$  going from vertex  $u$  to  $v$ , we say that

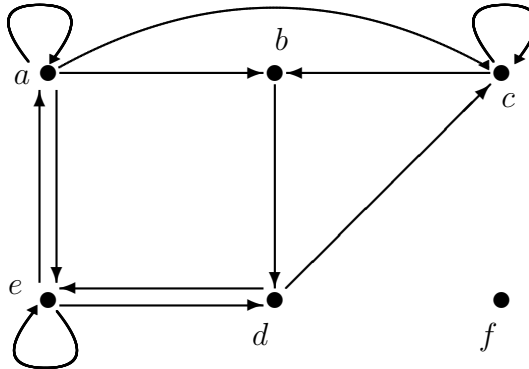
- $u$  is adjacent to  $v$ ,
- $u$  is the initial or start vertex of  $e$ , and
- $v$  is the terminal or end vertex of  $e$ .

The in-degree of a vertex  $v$  is the number of edges with  $v$  as their terminal vertex. The out-degree of a vertex  $v$  is the number of edges with  $v$  as their initial vertex. (A loop at a vertex contributes 1 to both the in-degree and the out-degree of this vertex.)

$$\begin{aligned}\text{number of edges} &= \text{sum of the in-degrees of vertices} \\ &= \text{sum of the out-degrees of vertices.}\end{aligned}$$



## Degrees of vertices: example 2



- $\text{in-degree}(a) = \text{in-degree}(b) = \text{in-degree}(d) = 2$ ,  
 $\text{in-degree}(c) = \text{in-degree}(e) = 3$ ,  
 $\text{in-degree}(f) = 0$ ,
- $\text{out-degree}(a) = 4$ ,  $\text{out-degree}(b) = 1$ ,  
 $\text{out-degree}(c) = \text{out-degree}(d) = 2$ ,  $\text{out-degree}(e) = 3$ ,  
 $\text{out-degree}(f) = 0$ .

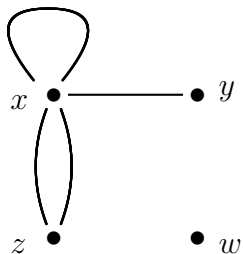
## Representing graphs: adjacency matrix

- List the vertices in some order horizontally from left to right.
- Then, using the same order, list them vertically from top to bottom.
- The entry in the  $i^{\text{th}}$  row and the  $j^{\text{th}}$  column is the number of edges going from vertex  $i$  to vertex  $j$ .

If the graph is undirected, then

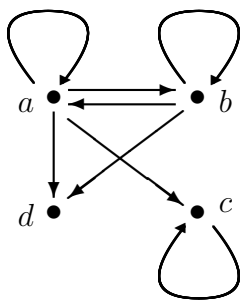
the number in the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column  
= the number in the  $j^{\text{th}}$  row and  $i^{\text{th}}$  column.

## Adjacency matrices: examples



	$x$	$y$	$w$	$z$
$x$	1	1	0	2
$y$	1	0	0	0
$w$	0	0	0	0
$z$	2	0	0	0

	$y$	$x$	$w$	$z$
$y$	0	1	0	0
$x$	1	1	0	2
$w$	0	0	0	0
$z$	0	2	0	0



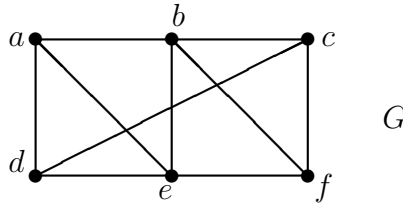
	$a$	$b$	$c$	$d$
$a$	1	1	1	1
$b$	1	1	0	1
$c$	0	0	1	0
$d$	0	0	0	0

	$b$	$c$	$d$	$a$
$b$	1	0	1	1
$c$	0	1	0	0
$d$	0	0	0	0
$a$	1	1	1	1

## Paths in simple graphs

- A **path** is a sequence of vertices travelling along edges.
- The **length** of a path is the number of edges in it.
- A path is called **simple** if it does not contain the same edge twice.
- A **Hamiltonian path** is a simple path passing through every vertex exactly once.

FOR EXAMPLE:



$(a, b, c, f, b, e)$  is a simple path in  $G$  of length 5

$(d, c, a, e)$  is not a path in  $G$  — WHY? no edge between  $c$  and  $a$

$(a, b, e, d, a, b)$  is a path in  $G$  of length 5, but it is not simple  
— WHY? contains the edge between  $a$  and  $b$  twice

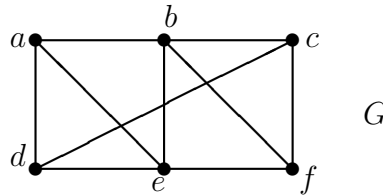
$(d, a, e, b, f, c)$  is a Hamiltonian path in  $G$

$(a, b, c, f, b, e, d)$  is a simple path, but not a Hamiltonian path in  $G$   
— WHY? passes  $b$  more than once

## Cycles in simple graphs

- A **cycle** is a path beginning and ending with the same vertex.
- The **length** of a cycle is the number of edges in it.
- A cycle is called **simple** if it does not contain the same edge twice.
- A **Hamiltonian cycle** is a simple cycle passing through every vertex

FOR EXAMPLE:



exactly once.

$(a, d, c, b, a)$  is a simple cycle in  $G$  of length 4

$(b, e, d, a, e, b)$  is a cycle in  $G$  of length 5, but it is not simple  
— WHY? contains the edge between  $b$  and  $e$  twice

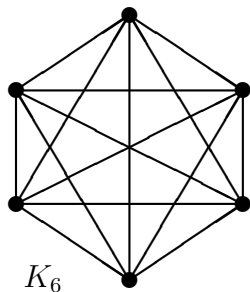
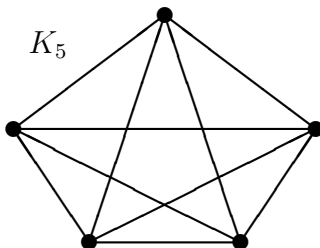
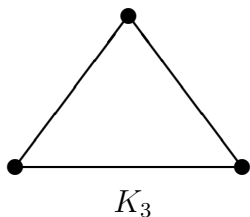
$(c, f, e, d, a, b, c)$  is a Hamiltonian cycle in  $G$

$(e, d, c, f, e, b, a, e)$  is a simple cycle in  $G$ , but not Hamiltonian  
— WHY? passes  $e$  more than once

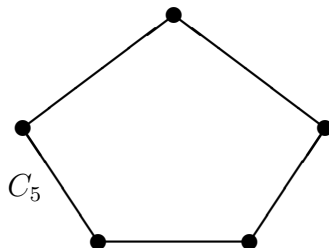
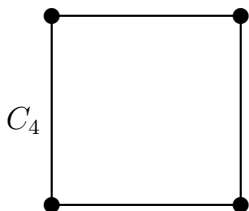
$(a, d, e, f, b, a)$  is a simple cycle in  $G$ , but not Hamiltonian  
— WHY? does not pass  $c$

## Special simple graphs

- The complete graph on  $n$  vertices (or  $n$ -clique), denoted by  $K_n$ , is the simple graph that contains an edge between each pair of distinct vertices.



- The  $n$ -cycle is denoted by  $C_n$ , for  $n \geq 4$ .

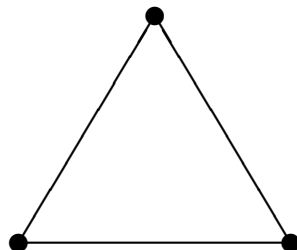
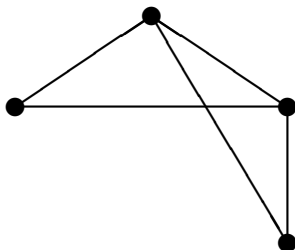
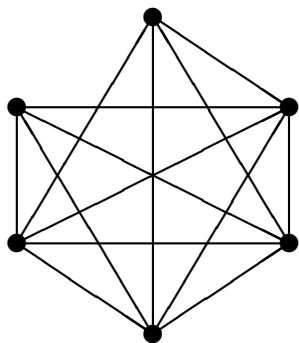


## Subgraphs of graphs

When edges and vertices are removed from a graph, without removing endpoints of any remaining edges, a smaller graph is obtained.

Such a graph is called a subgraph of the original graph.

FOR EXAMPLE: Each of the following 3 graphs

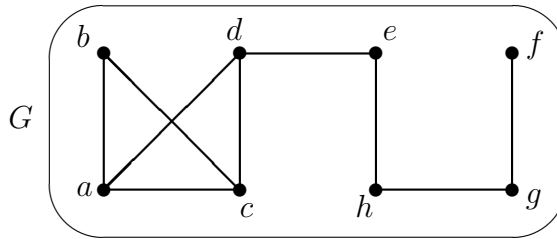


is a subgraph of  $K_6$ .

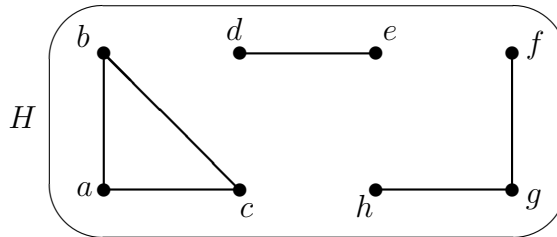
## Connected graphs

A simple graph is called **connected** if there is a path between every pair of distinct vertices.

FOR EXAMPLE:



is connected



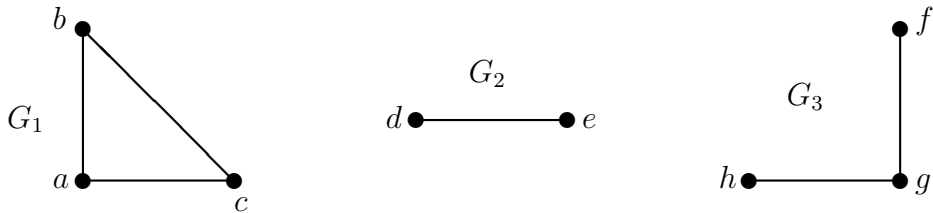
is not connected



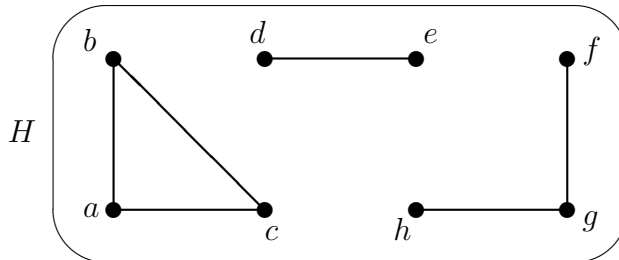
## Connected components of graphs

A connected component of a graph is a maximal connected subgraph.

- If a graph is connected, then it has only 1 connected component, *itself*.
- But if it is not connected, it can have more:



are the 3 connected components of



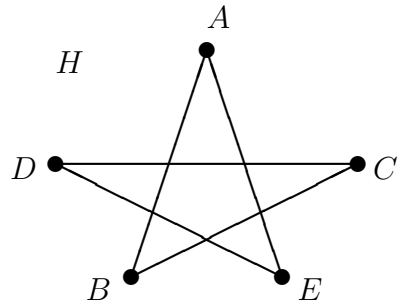
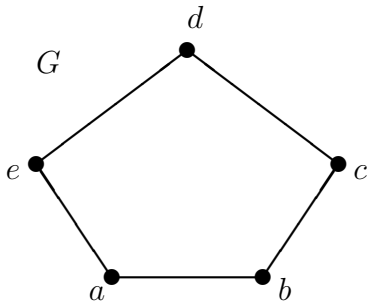
## Isomorphism of graphs: an example

The following instructions were told to two persons:

“Draw and label five vertices with  $a$ ,  $b$ ,  $c$ ,  $d$ , and  $e$ .

Connect  $a$  and  $b$ ,  $b$  and  $c$ ,  $c$  and  $d$ ,  $d$  and  $e$ , and  $a$  and  $e$ .”

They drew the graphs:



Surely, these drawings describe the same situation, though the graphs  $G$  and  $H$  appear dissimilar.

## Isomorphism of graphs

Graphs  $G$  and  $H$  are **isomorphic** if there is an **isomorphism** between them:

A function  $f$  from the vertices of  $G$  to the vertices of  $H$  such that

- $f$  is a bijection (one-to-one and onto)

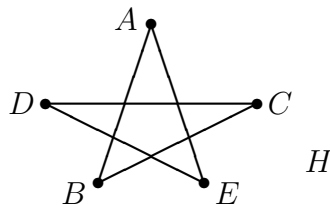
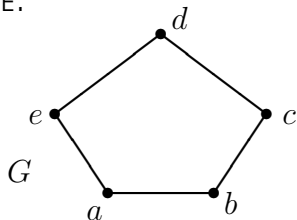
- $f$  'takes' edges to edges:

for all vertices  $x, y$  in  $G$ , if  $\begin{array}{c} y \\ | \\ x \end{array}$  then  $\begin{array}{c} f(y) \\ | \\ f(x) \end{array}$  in  $H$

- $f$  'takes' non-edges to non-edges:

for all vertices  $x, y$  in  $G$ , if  $\begin{array}{c} y \\ \bullet \\ x \end{array}$  then  $\begin{array}{c} f(y) \\ \bullet \\ f(x) \end{array}$  in  $H$

FOR EXAMPLE:

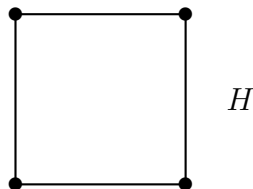
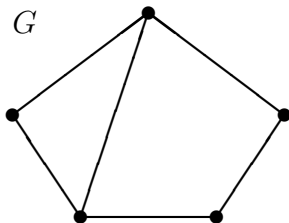


The function  $f$  defined by taking

$$f(a) = A, \quad f(b) = B, \quad f(c) = C, \quad f(d) = D, \quad f(e) = E$$

is an isomorphism, showing that graphs  $G$  and  $H$  are isomorphic.

## Isomorphic or not — how can we decide?



TASK: Determine whether two graphs  $G$  and  $H$  are isomorphic or not.

isomorphic = there is an isomorphism

not isomorphic = there is no isomorphism

- We can try **all possible functions** from  $G$  to  $H$ , and check whether any of them is an isomorphism.
- But this might take a lot of time: there are  $4^5 = 1024$  possible functions even for this simple example (see lecture slide 119).  
So for larger graphs it is kind of hopeless.

Is there some quicker way?

## Isomorphism of graphs: invariants

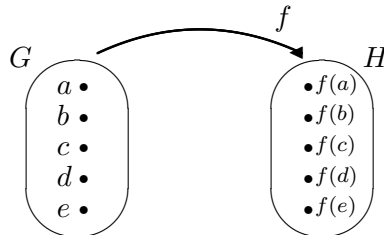
A property  $\mathcal{P}$  of graphs is called an **invariant** if it is 'preserved under isomorphisms': If  $G$  and  $H$  are isomorphic graphs, and  $G$  has property  $\mathcal{P}$ , then  $H$  has property  $\mathcal{P}$  as well.

FOR EXAMPLE: "Having 5 vertices" is an invariant:

If  $G$  and  $H$  are isomorphic graphs, and  $G$  has 5 vertices,  
then  $H$  has 5 vertices as well.

WHY? If  $G$  and  $H$  are isomorphic, then there is an isomorphism  $f$  between them.

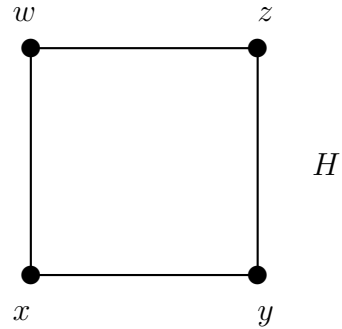
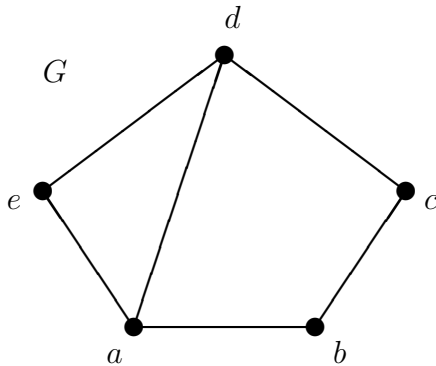
- $f$  is a bijection from the vertices of  $G$  (domain) to the vertices of  $H$  (codomain).



- As  $f$  is one-to-one,  $H$  has at least 5 vertices.
- And as  $f$  is onto,  $H$  has at most 5 vertices.

## Exercise 6.1

Determine whether  $G$  and  $H$  are isomorphic or not:



SOLUTION: We've just seen that the property "having 5 vertices" is an invariant. This property holds for  $G$ , but not for  $H$ . Therefore,  $G$  and  $H$  are not isomorphic.

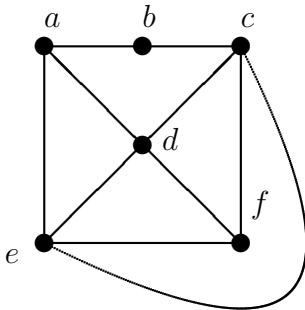
## Some more examples of invariants

- the number of edges
- the number of vertices of each degree
- containing a triangle ( $K_3$ ) as a subgraph
- containing two  $K_4$ s as disjoint subgraphs
- containing a simple cycle of length 4
- having a path of length 2 between two vertices of degree 2
- ...

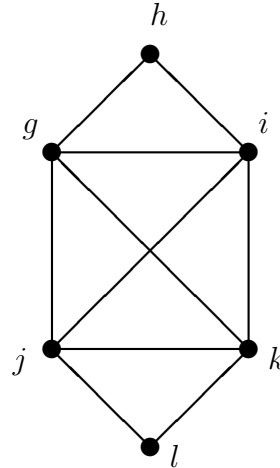
## Exercise 6.2

Determine whether  $G$  and  $G'$  are isomorphic or not:

$G$



$G'$



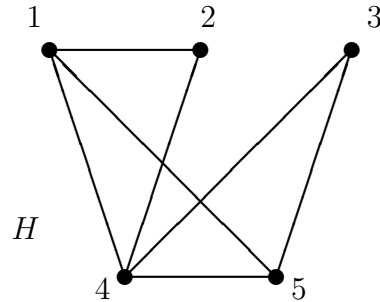
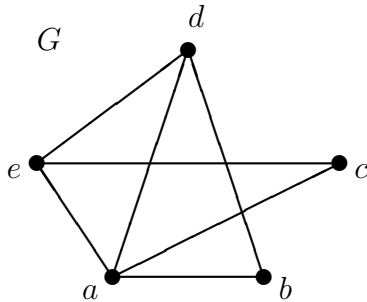
SOLUTION: Now it is a bit harder to find an invariant. Both graphs have 6 vertices and 10 edges. But, say, “having a vertex of degree 3” is an **invariant**.

This property holds in  $G$  (say,  $\text{degree}(a) = 3$ ), but does not hold in  $G'$ , showing that  $G$  and  $G'$  are not isomorphic.



## Exercise 6.3

Determine whether  $G$  and  $H$  are isomorphic or not:



SOLUTION: The function  $f$  defined by taking

$$f(a) = 4, \quad f(b) = 2, \quad f(c) = 3, \quad f(d) = 1, \quad f(e) = 5$$

is an isomorphism (because  $f$  is a bijection, takes edges to edges, and non-edges to non-edges). This shows that graphs  $G$  and  $H$  are isomorphic.

HOW TO FIND AN ISOMORPHISM?

**Hint:** Always keep in mind that if  $h$  is an isomorphism between graphs  $G$  and  $H$ , then for every vertex  $x$  in  $G$ , the degrees of  $x$  and  $h(x)$  must be the same.

## Isomorphic or not — how to decide?

TASK: Determine whether two graphs  $G$  and  $H$  are isomorphic or not.

SOLUTION: There is no quick and easy way, without thinking.

The computational complexity of the *graph isomorphism problem* is one of the most famous unsolved problems in computer science.

It is **not known**

- whether the problem is *solvable in polynomial time* (that is, whether there exists a general method that is always quicker than trying out all possible functions), and
- whether the problem is *definitely NOT solvable in polynomial time*. (For many other well-known computational problems, there are known PROOFS showing that they cannot be solved quicker than trying out all options.)

## Isomorphic or not — so how can we decide?

TASK: Determine whether two graphs  $G$  and  $H$  are isomorphic or not.

SOLUTION: We have to try in parallel:

- To describe a bijection between the vertices of  $G$  and  $H$  that ‘takes’ edges to edges, non-edges to non-edges.  
     $\leadsto$  If we succeed, the answer is YES
- To find an invariant  $\mathcal{P}$   
    and show that  $G$  has  $\mathcal{P}$  but  $H$  doesn’t, or the other way round.  
     $\leadsto$  If we succeed, the answer is NO

It would be nice to have a list of easily checkable invariants that isomorphic graphs and only isomorphic graphs share. Then we would just have to check those. BTW having such a list would solve the *graph isomorphism problem*.

Unfortunately, no one has yet succeeded in finding such a list of invariants, so determining whether two graphs are isomorphic or not might require some

**creative thinking.**