Graphs

Laura Kovács

Graphs – Definition

An undirected graph (ungerichteter Graph), or simply a graph $G = (V, E)$ consists of:

- a set V of nodes / vertices (Knoten),

and

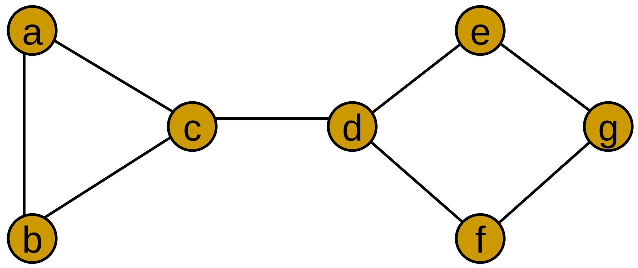
- a set E of edges (Kanten), connecting two distinct nodes: $E = \{ \{u,v\} \mid u,v \in V\}$.

Note: Unlike trees, graphs have no restrictions on edges connecting nodes!

A tree can be viewed as a special kind of graph.

Example of a graph:

(it is NOT a tree!)



V = {a,b,c,d,e,f,g}

E = { {a,b}, {a,c}, {b,c}, {c,d}, {d,e}, {d,f}, {e,g}, {f,g} }

No difference between edge {a,b} or {b,a} in the undirected graph!

{a, b} indicates that nodes a and b are connected by edge {a,b}.

Graphs – Adjacent and Incident Nodes

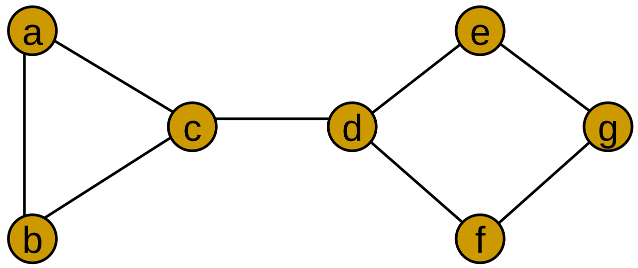
Consider a graph $G = (V, E)$.

If $\{u,v\} \in E$ ({u,v} is an edge in G), then:

nodes u and v are said to be adjacent / neighbors (adjazent).

A node $u \in V$ is called incident (inzident) to an edge that contains u.

Example:



- **a and b are adjacent**
- **a and f are not adjacent**
- **a is incident to {a,b}, and {a,c}**
- **a is not incident to {d,f}**

Graphs – Representing Graphs via Adjacency Matrix

Consider a graph G = (V, E), where V has n nodes.

The adjacency matrix (adjacency list, Adjazenzmatrix) of G is an

n × n matrix A (that is, A has n rows and n columns) such that

$\mathbf{A_{u\,v} = 1 \text{ if } \{u,v\} \in E}$ and $\mathbf{A_{u\,v} = 0 \text{ if } \{u,v\} \notin E}$

Example:

	a	b	c	d	e	f	g
a	0	1	1	0	0	0	0
b	1	0	1	0	0	0	0
c	1	1	0	1	0	0	0
d	0	0	1	0	1	1	0
e	0	0	0	1	0	0	1
f	0	0	0	1	0	0	1
g	0	0	0	0	1	1	0

Adjacency matrix (list)

Graphs – Representing Graphs via Adjacency Matrix

Consider a graph G = (V, E), where V has n nodes.

The adjacency matrix (adjacency list, Adjazenzmatrix) of G is an

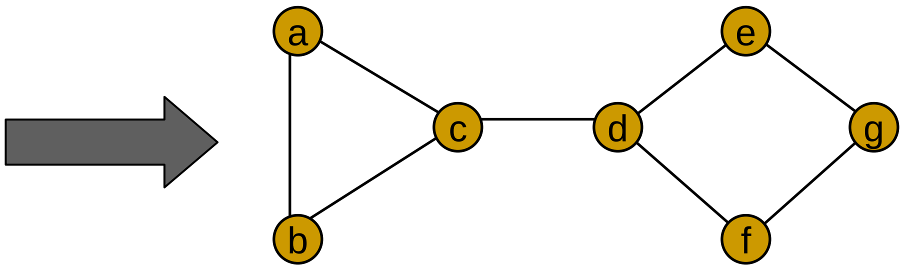
n × n matrix A (that is, A has n rows and n columns) such that

$A_{u\,v}$= 1 if {u,v} ∈ E and **$A_{u\,v}$= 0 if {u,v} ∉ E**

Example:

	a	b	c	d	e	f	g
a	0	1	1	0	0	0	0
b	1	0	1	0	0	0	0
c	1	1	0	1	0	0	0
d	0	0	1	0	1	1	0
e	0	0	0	1	0	0	1
f	0	0	0	1	0	0	1
g	0	0	0	0	1	1	0

Adjacency matrix (list)



Graph

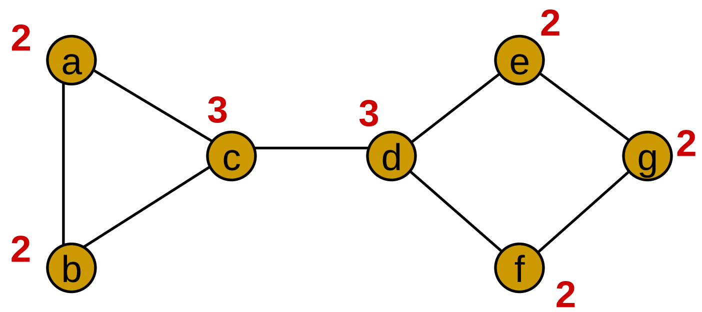
Graphs – Degree of a Node

Consider a graph G = (V, E).

The degree (grad) of a node u ∈ V is the number of edges to which u incident is.

Example:

- degree of a is 2



Graphs and Binary Relations

A graph G = (V, E) consists of a set of nodes V and a *binary relation* E $\subseteq$ V $\times$V.

- If {u,v}$\in$E, that is *u E v*, then there is an edge {u,v} in the graph.

Graphs and Binary Relations

A graph G = (V, E) consists of a set of nodes V and a *binary relation* $E \subseteq V \times V$.

- If $\{u,v\} \in E$, that is *u E v*, then there is an edge {u,v} in the graph.

For a (undirected) graph G = (V, E), the binary relation $E \subseteq V \times V$ is symmetric.

- If {u,v} is an edge in G, so is {v,u} an edge in G.

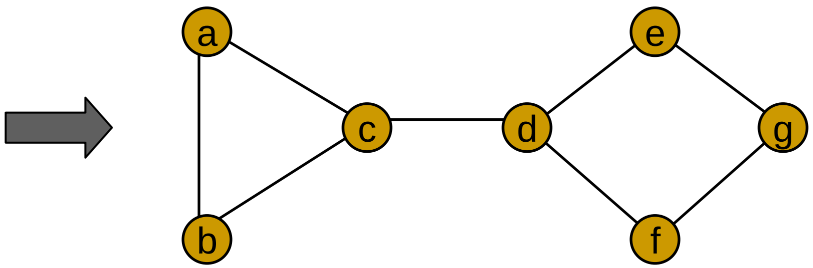
Example:

Binary relation $E \subseteq V \times V$, where:

V={a,b,c,d,e,f,g}

E= { {a,b}, {a,c}, {b,c}, {c,d}, {d,e}, {d,f}, {e,g}, {f,g},

{b,a}, {c,a}, {c,b}, {d,c}, {e,d}, {f,d}, {g,e}, {g,f} }



Graphs – Directed Graphs

A graph G = (V, E) is called directed (gerichtet) if its edges give *directions* (Orientierung) from one node to another.

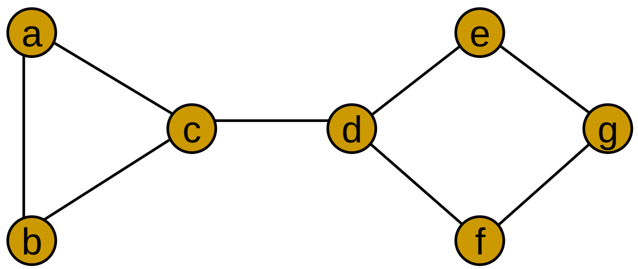
For an edge {u,v} $\in$ E in a *directed graph*, we say that:

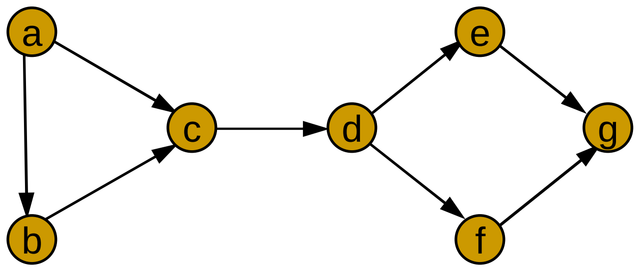
- {u,v} is directed (orientiert) from u to v;

- u is the head (Kopf) of edge {u,v}.
- v is the tail (Ende) of the edge {u,v}.

A directed graph is shortly called Digraph.

Example:



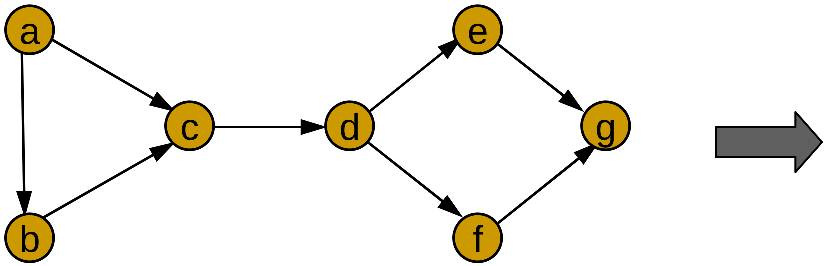


Undirected graph	**Directed graph**
{a,b} is the same as {b,a}	**{a,b} is NOT the same as {b,a}**
{a,b} is an edge, and so is {b,a}	**{a,b} is an edge, but {b,a} is NOT.**

Directed Graphs and Binary Relations

A directed graph G = (V, E) is the binary relation $E \subseteq V \times V$ over the set of *nodes* V.

Example:



Directed graph

Binary relation $E \subseteq V \times V$, where:

V={a,b,c,d,e,f,g}

E= { {a,b}, {a,c}, {b,c}, {c,d}, {d,e}, {d,f}, {e,g}, {f,g} }

Directed Graphs and Binary Relations

A directed graph G = (V, E) is the binary relation $E \subseteq V \times V$ over the set of *nodes* V.

A binary relation $E \subseteq V \times V$ over the set of *objects* V defines a directed graph G=(V,E).

Example:

Binary relation $E \subseteq V \times V$, where:

V={a,b,c,d,e,f,g}

E= { {a,b}, {a,c}, {b,c}, {c,d}, {d,e}, {d,f}, {e,g}, {f,g} }

Directed graph

Directed Graphs and Binary Relations

A directed graph G = (V, E) is the binary relation E ⊆ V ×V over the set of *nodes* V.

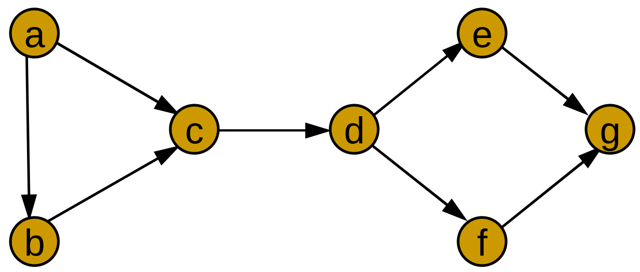
A binary relation E ⊆ V ×V over the set of *objects* V defines a directed graph G=(V,E).

Example:

Binary relation E ⊆ V ×V, where:

V={a,b,c,d,e,f,g}

E= { {a,b}, {a,c}, {b,c}, {c,d}, {d,e}, {d,f}, {e,g}, {f,g} }



Directed graph

Directed Graphs and Binary Relations

A directed graph G = (V, E) is the binary relation $E \subseteq V \times V$ over the set of *nodes* V.

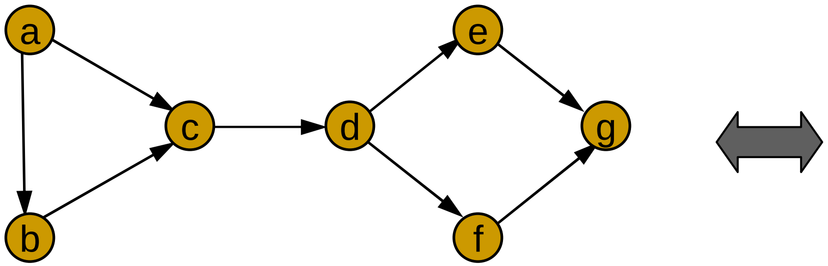
A binary relation $E \subseteq V \times V$ over the set of *objects* V defines a directed graph G=(V,E).

Note: If $E \subseteq V \times V$ is a symmetric relation,

then the undirected graph G'=(V,E) and directed graph G=(V,E) are the same.

Only directed graphs can model antisymmetric /asymmetric/non-symmetric/ partial order relations!

Example:



Binary relation $E \subseteq V \times V$, where:

V={a,b,c,d,e,f,g}

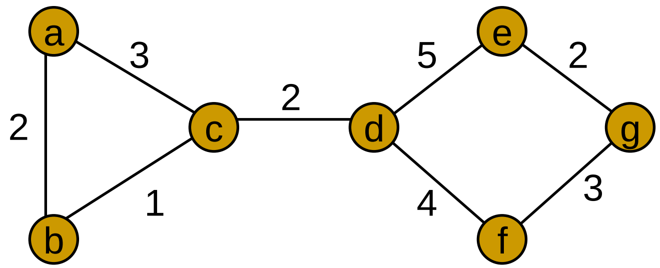
E= { {a,b}, {a,c}, {b,c}, {c,d}, {d,e}, {d,f}, {e,g}, {f,g} }

Directed graph

Graphs – Weighted Graphs

A graph G = (V, E) is called **weighted** (gewichtet) when
a **weight/label** (Gewicht/Attribut) is associated with every edge in the graph.

Example:

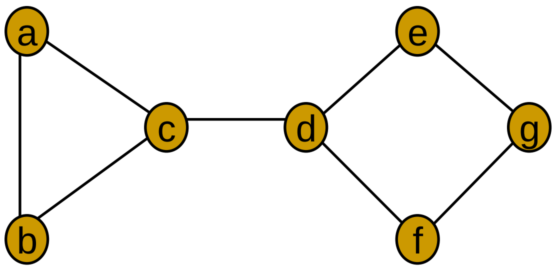


Graphs – Complete Graphs

A graph G = (V, E) is called complete (vollständig) when every two distinct nodes is connected by an edge

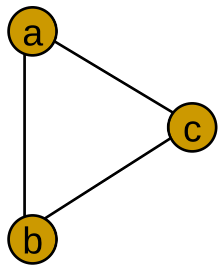
Note: G is complete when every two distinct nodes are adjacent.

Example:

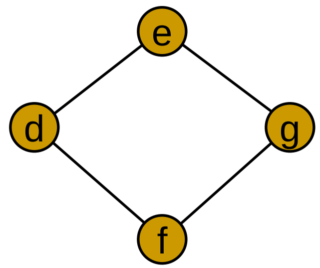


Not complete graph!

ex: {b,f} is missing



Complete graph!



Not complete graph!

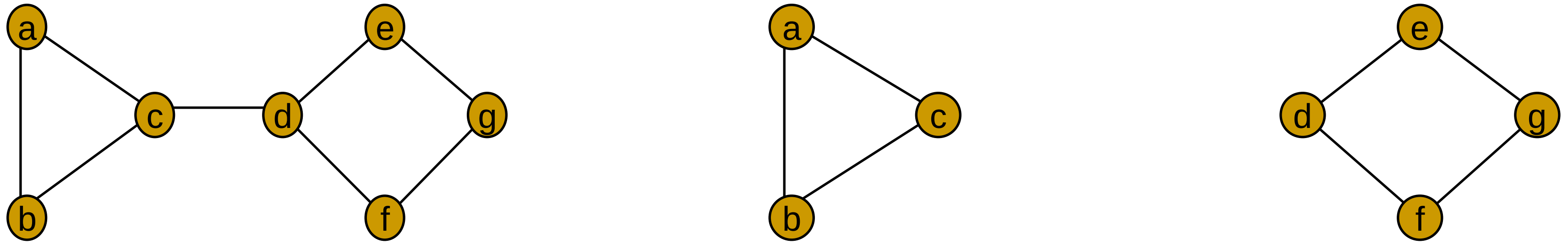
ex: {d.g} is missing

Graphs – Complete Graphs

A graph G = (V, E) is called complete (vollständig) when every two distinct nodes is connected by an edge

Note: G is complete when every two distinct nodes are adjacent.

Example:



Not complete graph!

ex: {b,f} is missing

Complete graph!

Not complete graph!

ex: {d.g} is missing

In a complete graph with n nodes, the degree of every node is n-1.

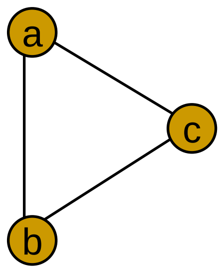
Note: A graph refers to an undirected graph. When a graph is directed, then we explicitly say directed graph.

Graphs – Complete Graphs

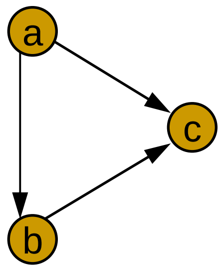
A graph G = (V, E) is called complete (vollständig) when every two distinct nodes is connected by an edge

Note: G is complete when every two distinct nodes are adjacent.

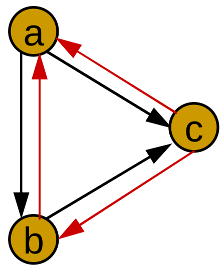
Example:



Complete graph!



Not a complete directed graph!



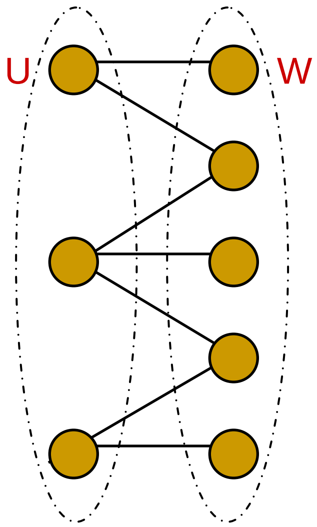
Complete directed graph!

Graphs – Bipartite Graphs

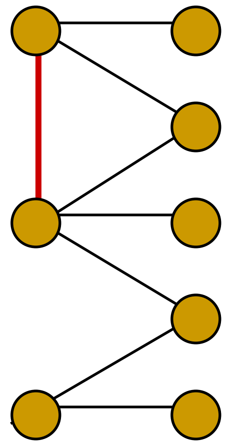
A graph G = (V, E) is called bipartite (bipartit) if:

- its nodes can be divided into two disjoint sets *U* and *W* ($V=U \cup W$, $U \cap W=\emptyset$);
- its edges only connect a node from U with a node from W.

Example:



Bipartite graph



Not a bipartite graph

Graphs – Paths and Cycles

Consider a graph G = (V, E).

- A path/way (Pfad/Weg) in a graph is a sequence of nodes k nodes

  $(u_1, u_2, \ldots, u_k)$ $u_1,\ldots,u_k \in V$

  such that each node and the next node are connected by an edge.

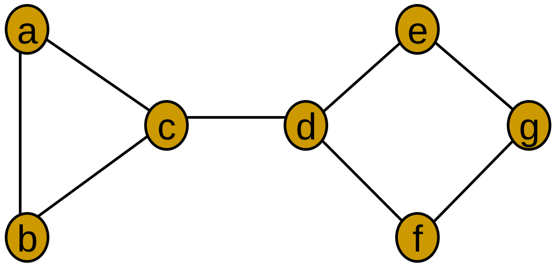
- The Length (Länge) of the path $(u_1, u_2, \ldots, u_k)$ is k-1.

.

Example:

- (a, b, c, d, f) is a path of length 4.

- (a, b, c, d, g) is NOT a path.



Graphs – Paths and Cycles

Consider a graph $G = (V, E)$.

- A path/way (Pfad/Weg) in a graph is a sequence of nodes k nodes

  $(u_1, u_2, \ldots, u_k)$ $u_1,\ldots,u_k \in V$

  such that each node and the next node are connected by an edge.

- The Length (Länge) of the path $(u_1, u_2, \ldots, u_k)$ is k-1.

- The path $(u_1, u_2, \ldots, u_k)$ is a cycle (Zyklus, Kreis) if:
  - $u_1 = u_k$ and the **length** of the path **is ≥ 3** (that is $k \geq 4$)

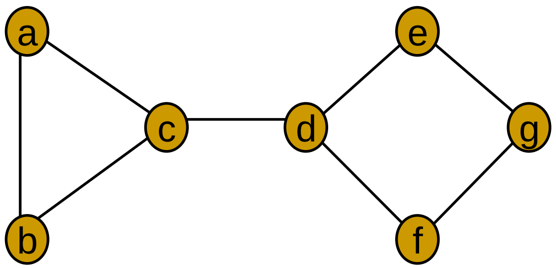
Example:

- (a, b, c, d, f) is a path of length 4.

- (a, b, c, d, g) is NOT a path.

- (a, b, c) is a path of length 2, and is not a cycle!

- **(a, b, c, a) is a path of length 3, and is a cycle!**



Graphs – Paths and Cycles

Consider a graph $G = (V, E)$.

- A path/way (Pfad/Weg) in a graph is a sequence of nodes k nodes

  $(u_1, u_2, \ldots, u_k)$ $u_1,\ldots,u_k \in V$

  such that each node and the next node are connected by an edge.

- The Length (Länge) of the path $(u_1, u_2, \ldots, u_k)$ is k-1.

- The path $(u_1, u_2, \ldots, u_k)$ is a cycle (Zyklus, Kreis) if:
  - $u_1 = u_k$ and the **length** of the path **is ≥ 3** (that is $k \geq 4$)

- If the graph G has one or more cycles, then it is called a cyclic (zyklisch) graph.

- A graph with no cycles is called an acyclic (azyklish) graph. *Ex: Trees are acyclic graphs.*

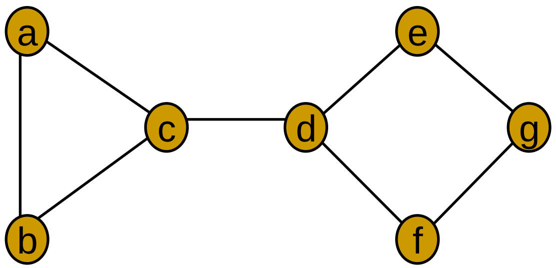
Example:

- (a, b, c, d, f) is a path of length 4.

- (a, b, c, d, g) is NOT a path.

- (a, b, c) is a path of length 2, and is not a cycle!

- (a, b, c, a) is a path of length 3, and is a cycle!



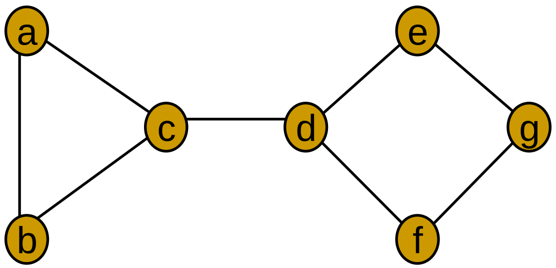
An acyclic binary relation can be modelled with an acyclic graph.

Graphs – Loops

Consider a graph G = (V, E).

- An edge connecting a node u with the node u itself is called a loop (Schlaufe).

Example:

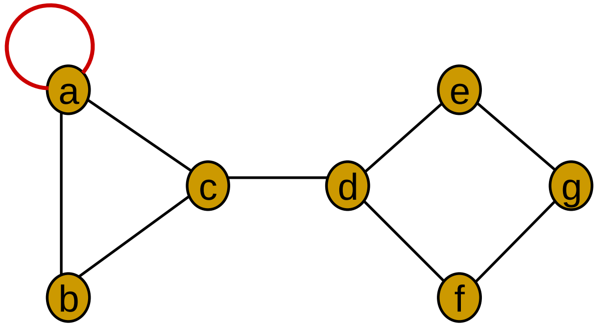


Loop-free graph

(that is, a graph with no loop)

(a) is a path of length 0

(a,a) is not a path



Graph with a loop

(a) is a path of length 0

(a,a) is a path of length 1

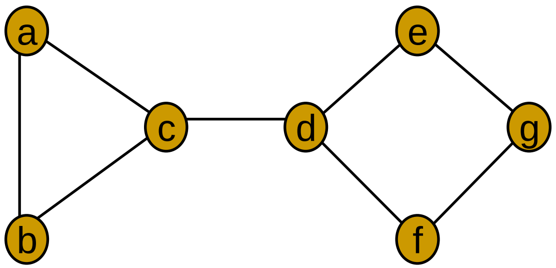
{a,a} is a loop

Graphs – Loops

Consider a graph G = (V, E).

- An edge connecting a node u with the node u itself is called a loop (Schlaufe).

Example:

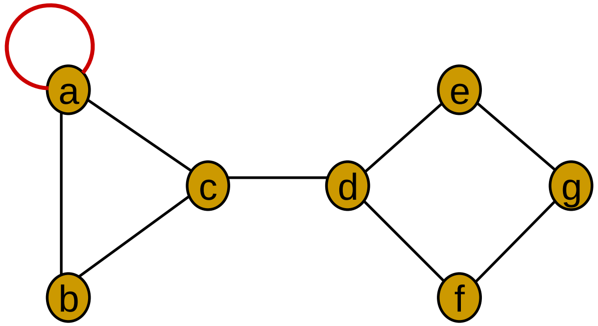


Loop-free graph

(that is, a graph with no loop)

(a) is a path of length 0

(a,a) is not a path



Graph with a loop

(a) is a path of length 0

(a,a) is a path of length 1

{a,a} is a loop

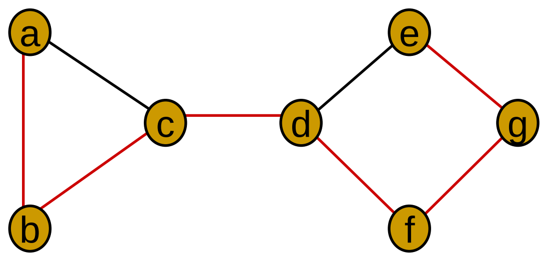
- A reflexive binary relation can be modelled with a graph with loops on each node.
- An irreflexive binary relation can be modelled with a loop-free graph.

- For a complete and loop-free graph G=(V,E): $\forall$ u,v: u,v$\in$V: u $\neq$v $\Rightarrow$ {u,v} $\in$ E.

Graphs – Hamiltonian and Eulerian Paths and Cycles

Consider a graph G = (V, E).

- A path is called a hamiltonian path (Hamilton-Pfad) if:
  - it contains all nodes of the graph;
  - each node is contained only once.
- A cycle is a hamiltonian cycle (Hamilton-Kreis) if:
  - it contains all nodes of the graph;
  - each node is contained only once, except the start and end node u_1 which is contained exactly twice.



Example:

(a, b, c, d, f, g, e) is a hamiltonian path

Graph has no hamiltonian cycles

Graphs – Hamiltonian and Eulerian Paths and Cycles

Consider a graph $G = (V, E)$.

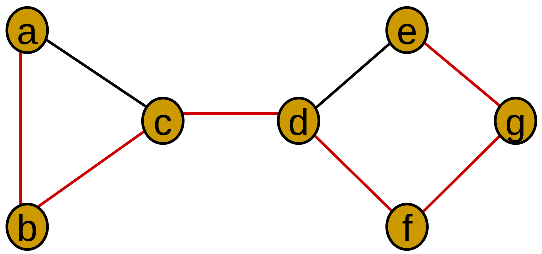
Old Swiss 10 Franc banknote honoring
Leonard Euler (1707-1783)

- A path is called a hamiltonian path (Hamilton-Pfad) if:
  - it contains all nodes of the graph;
  - each node is contained only once.
- A cycle is a hamiltonian cycle (Hamilton-Kreis) if:
  - it contains all nodes of the graph;
  - each node is contained only once, except the start and end node u_1 which is contained exactly twice.
- A path is called an eulerian path (Euler-Pfad) if:
  - it contains all edges of the graph;
  - each edge is contained only once.
- An eulerian path that is a cycle is called an eulerian cycle (Euler-Kreis).

Example:

(a, b, c, d, f, g, e) is a hamiltonian path, not an eulerian path!

Graph has no hamiltonian cycles

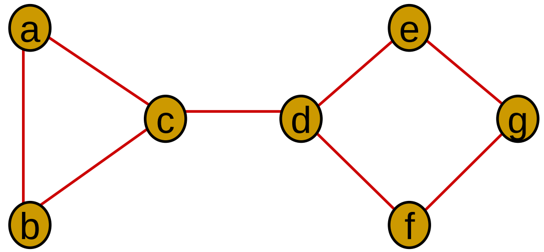


Graphs – Hamiltonian and Eulerian Paths and Cycles

Consider a graph G = (V, E).

Old Swiss 10 Franc banknote honoring
Leonard Euler (1707-1783)

- A path is called a hamiltonian path (Hamilton-Pfad) if:
  - it contains all nodes of the graph;
  - each node is contained only once.
- A cycle is a hamiltonian cycle (Hamilton-Kreis) if:
  - it contains all nodes of the graph;
  - each node is contained only once, except the start and end node u_1 which is contained exactly twice.
- A path is called an eulerian path (Euler-Pfad) if:
  - it contains all edges of the graph;
  - each edge is contained only once.
- An eulerian path that is a cycle is called an eulerian cycle (Euler-Kreis).



Example:

(a, b, c, d, f, g, e) is a hamiltonian path, not an eulerian path!

Graph has no hamiltonian cycles, nor eulerian cycles.

(c, a, b, c, d, e, g, f, d) is an eulerian path, but not a hamiltonian path.

Graphs – Spanning Trees and Components

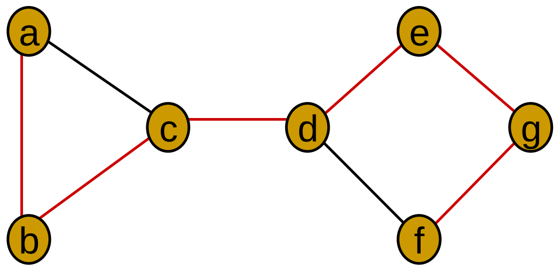
Consider a graph G = (V, E).

The subset $T \subseteq E$ is a spanning tree (spannender Baum) of G if:

- *every node* in V belongs to an edge of T;
- between *every two distinct nodes* of G there is a *path* in T;
- edges of T form *no cycles*.

Example:

T={ {a,b}, {b,c}, {c,d}, {d,e}, {e,g}, {g,f}} is a spanning tree.



Graphs – Spanning Trees and Components

Consider a graph G = (V, E).

The subset T ⊆ E is a spanning tree (spannender Baum) of G if:

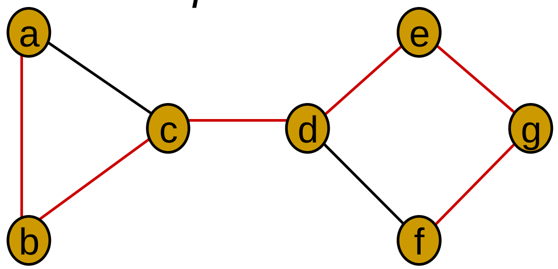
- *every node* in V belongs to an edge of T;
- between *every two distinct nodes* of G there is a *path* in T;
- edges of T form *no cycles*.

The subset T ⊆ E is a component (Komponent) of G if:

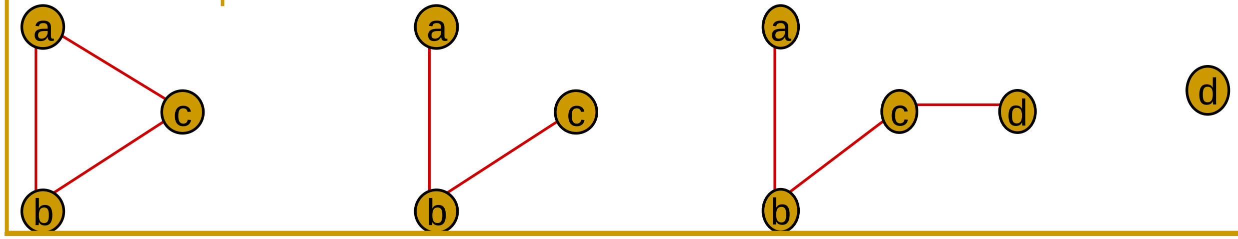
- between *every two distinct nodes* belonging to some edges of T there is a *path* in T.

Example:

T={ {a,b}, {b,c}, {c,d}, {d,e}, {e,g}, {g,f}} is a spanning tree.



Some components:



Graphs – Critical and Isolated Nodes

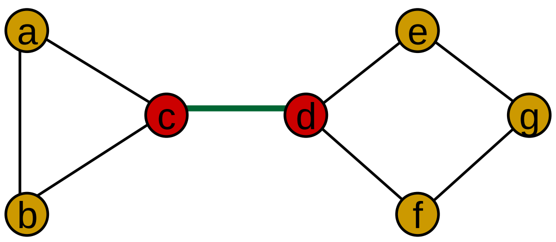
Consider a graph G = (V, E).

- A node u∈V in the graph G is critical (kritisch) if
  by deleting u from G the graph G is divided into not connected components.
- An edge {u,v}∈E in the graph G is critical (kritisch) if
  by deleting {u,v} from G the graph G is divided into not connected components.
- Critical nodes and edges of the graph G form the articulation points (Artikulationspunkte) of G.

Example:

- Critical nodes: c, d;

- Critical edge: {c,d};



Graphs – Critical and Isolated Nodes

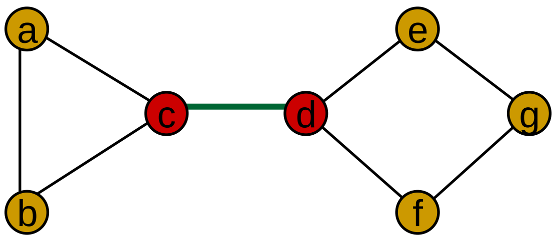
Consider a graph G = (V, E).

- A node u∈V in the graph G is critical (kritisch) if
  by deleting u from G the graph G is divided into not connected components.
- An edge {u,v}∈E in the graph G is critical (kritisch) if
  by deleting {u,v} from G the graph G is divided into not connected components.
- Critical nodes and edges of the graph G form the articulation points (Artikulationspunkte) of G.
- A node u∈V in the graph G is isolated (isoliert) if it is the only node of a component of G.

Example:

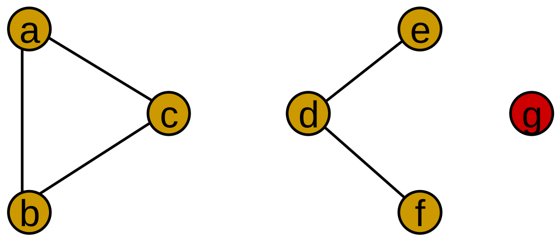
- Critical nodes: c, d;

- Critical edge: {c,d};



Example:

- Isolated node: g



Graphs – Biconnected Components

Consider a (undirected) graph G = (V, E).

- A component $T \subseteq E$ of G is a biconnected component (zweifach zusammenhängend), if
  - by deleting an arbitrary node from T,
  - the remaining nodes and edges in T still form a component of G.

Example:

- Some Biconnected components:

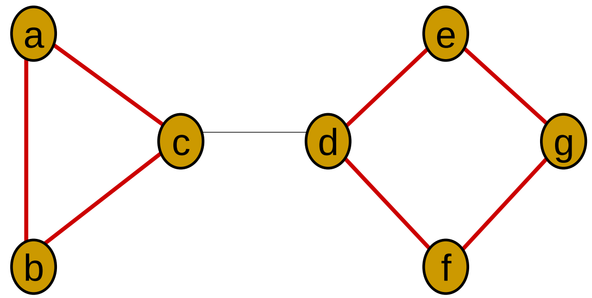
T_1 ={ {a,b}, {a,c}, {b,c} }

T_2 = { {d,e}, {d,f}, {e,g}, {f,g} }

- Not biconnected component:

T_3 = { {a,b}, {b,c} }

T_4 = { {d,e}, {d,f} }

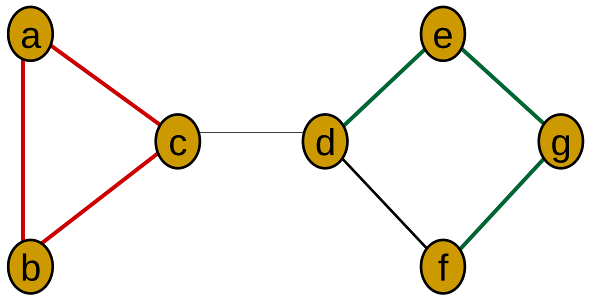


Graphs – Subgraphs and Clique

Consider a graph $G = (V, E)$.

- The graph $G_1=(V_1,E_1)$ is a subgraph (Subgraph) of G, if

  $V_1 \subseteq V$ and $E_1=\{\{u,v\}\in E \mid u, v \in V_1 \} \subseteq E$.



Example:

- $G_1 = \{V_1,E_1\}$ is a subgraph, where:

  $V_1 = \{a,b,c\}$ $T_1 = \{ \{a,b\}, \{a,c\}, \{b,c\}\}$

- $G_2 = \{V_2,E_2\}$ is NOT a subgraph, where:

  $V_2 =\{d,e,f,g\}$ $T_2 = \{ \{d,e\}, \{e,g\}, \{g,f\}\}$

Graphs – Subgraphs and Clique

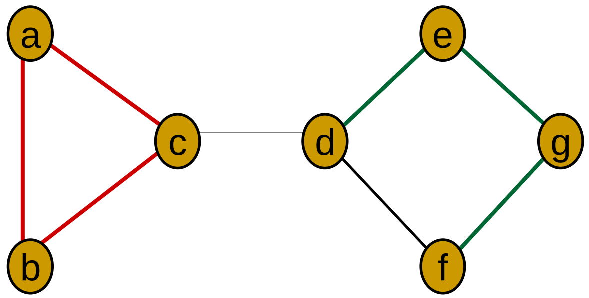
Consider a graph G = (V, E).

- The graph $G_1=(V_1,E_1)$ is a subgraph (Subgraph) of G, if

  $V_1 \subseteq V$ and $E_1=\{\{u,v\}\in E \mid u, v \in V_1 \} \subseteq E$.

- A k-clique (k-Clique, Clique der Grösse k) of G is a *subgraph* of G which is

  complete and contains *k nodes*.



Example:

- $G_1 = \{V_1,E_1\}$ is a subgraph, where:

  $V_1 = \{a,b,c\}$ $T_1 = \{ \{a,b\}, \{a,c\}, \{b,c\}\}$

  G_1 is a 3-Clique! Since it is complete, one can also write that {a,b,c} forms a 3-clique!

  No other 3-cliques, nor 4-cliques! Note: {d,e,g,f} is not a 4-clique! (Although these nodes with their edges form a subgragh!)

- $G_2 = \{V_2,E_2\}$ is NOT a subgraph, where:

  $V_2 =\{d,e,f,g\}$ $T_2 = \{ \{d,e\}, \{e,g\}, \{g,f\}\}$

Directed Graphs – Connected Components

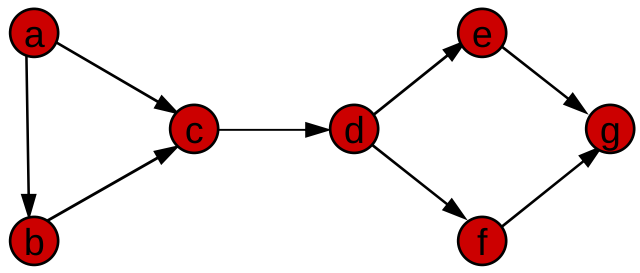
Consider a digraph graph G = (V, E).

- A node v is weakly reachable (schwach erreichbar) from a node u, if there is an *undirected path* from u to v.
- A component T⊆ E is weakly connected (schwach zusammenhängend) if every node in T is weakly reachable from any other node in T.

Example:

- Node a is weakly reachable from node d;

- {{a,b}, {b,c}, {a,c}, {c,d}} is weakly connected;



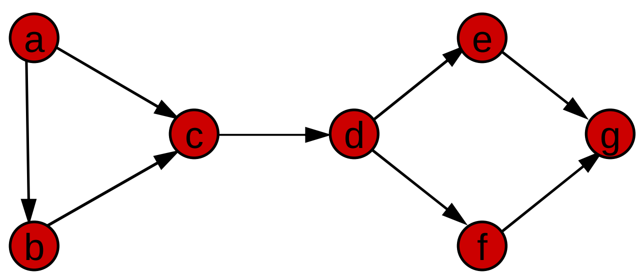
Directed Graphs – Connected Components

Consider a digraph graph G = (V, E).

- A node v is weakly reachable (schwach erreichbar) from a node u, if there is an *undirected path* from u to v.
- A component T⊆ E is weakly connected (schwach zusammenhängend) if every node in T is weakly reachable from any other node in T.
- A node v is strongly reachable (stark erreichbar) from a node u, if there is an *(directed) path* from u to v.
- A component T⊆ E is strongly connected (stark zusammenhängend) if every node in T is strongly reachable from any other node in T.

Example:

- Node a is weakly reachable from node d;
- {{a,b}, {b,c}, {a,c}, {c,d}} is weakly connected;
- Node a is NOT strongly reachable from node d;
- {{a,b}, {b,c}, {a,c}, {c,d}} is weakly connected;



- The only strongly connected components are given by Ø,

  that is only one node and no edge in a strongly connected component.

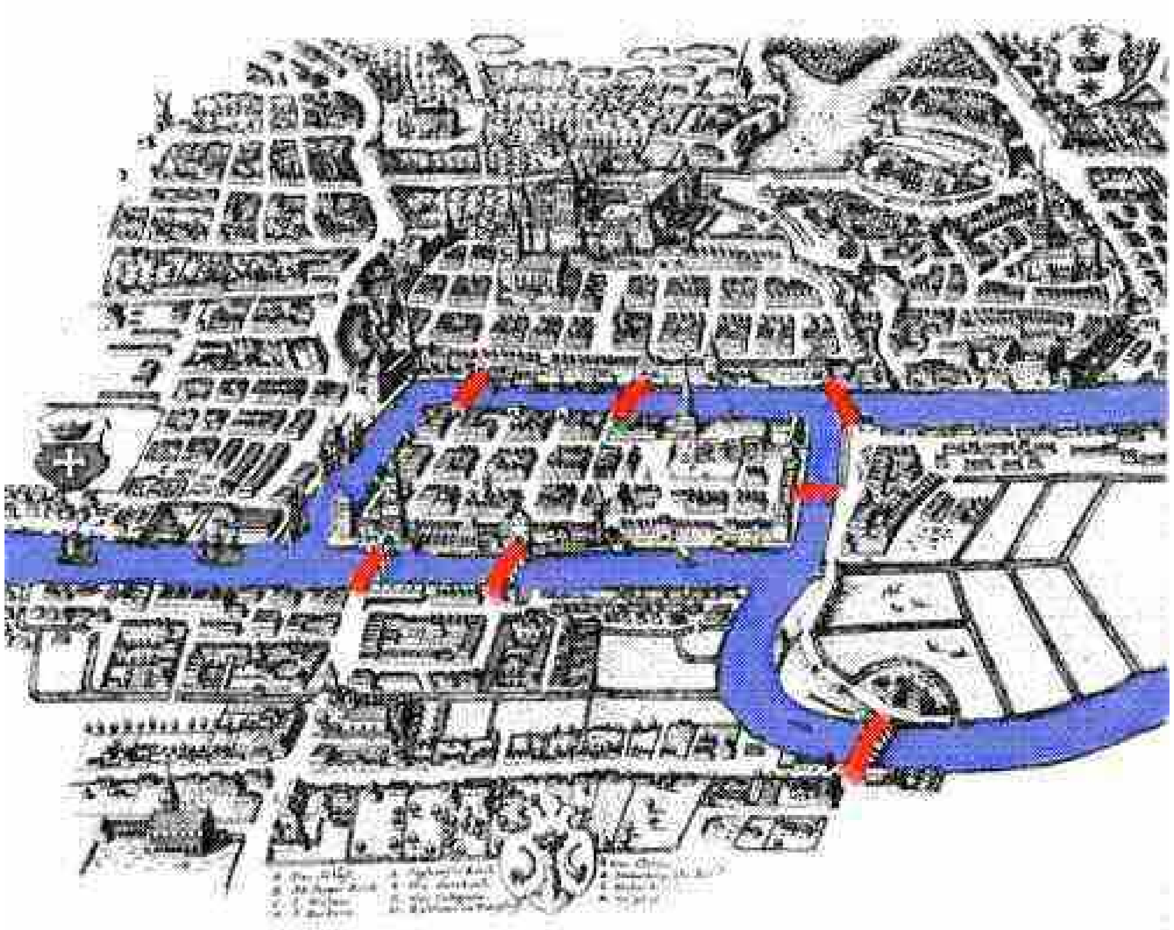
Example: Seven Bridges of Königsberg

Leonhard Euler, 1736

<u>Problem:</u>

Two large islands connected to each other and the mainland by seven bridges.

Decide whether it is possible to follow a path that crosses each bridge exactly once and returns to the starting point.



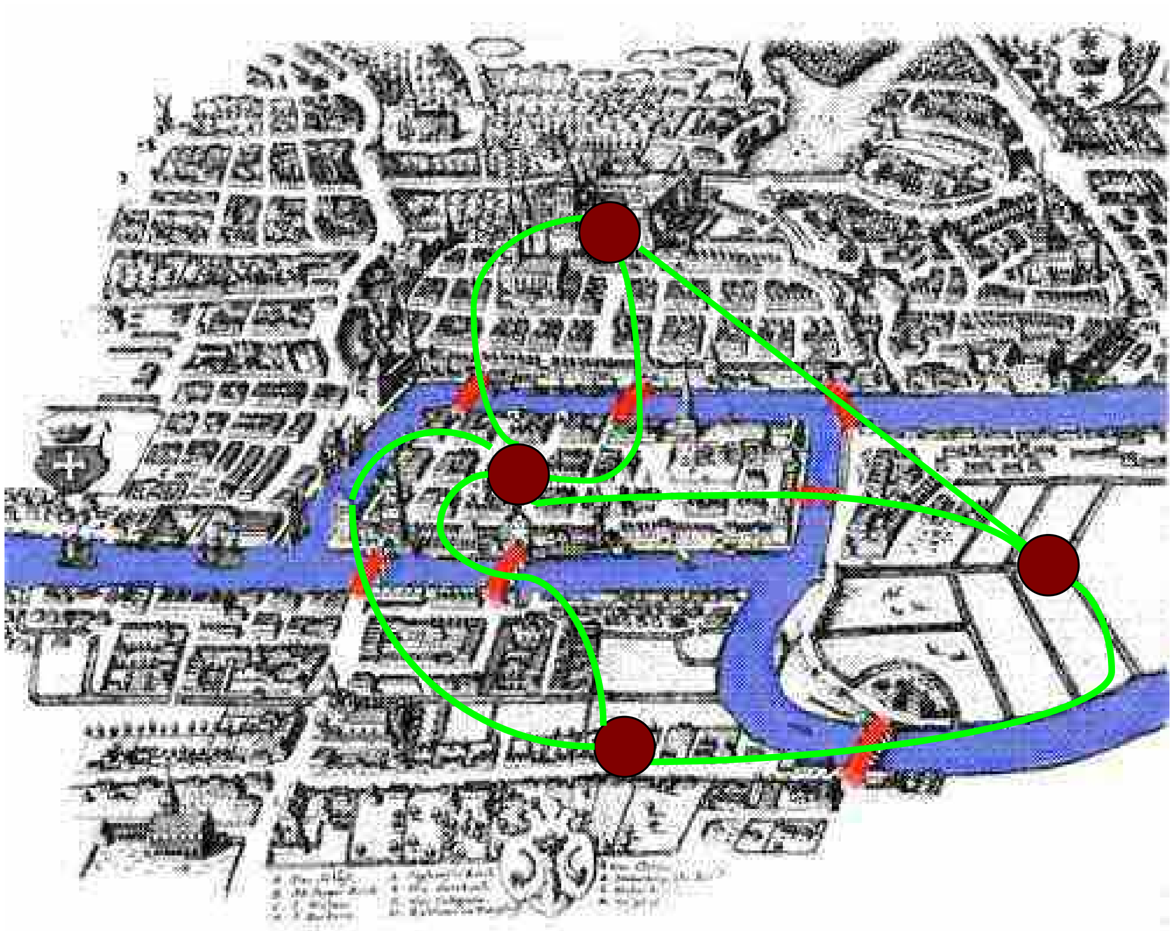
Example: Seven Bridges of Königsberg

Leonhard Euler, 1736

<u>Problem:</u>

Two large islands connected to each other and the mainland by seven bridges.

Decide whether it is possible to follow a path that crosses each bridge exactly once and returns to the starting point.



Example: Seven Bridges of Königsberg

Leonhard Euler, 1736

<u>Problem:</u>

Two large islands connected to each other and the mainland by seven bridges.

Decide whether it is possible to follow a path that crosses each bridge exactly once and returns to the starting point.

⬇

Is there an **Eulerian Cycle**?

Euler proved: no eulerian cycle.