GRAPHS

OUTLINE:

1) Various types of graphs
2) Some special graphs
3) Operations on graphs
4) Representing graphs
5) Properties of graphs

1. VARIOUS TYPES OF GRAPHS

Graphs

- There are various types of graphs, which are used to represent different situations.
- Common features of all types of graphs:

1) A set V of vertices (points) representing objects

2) A set E of edges (segments) connecting pairs of vertices and representing relations between objects

- Notation: $G = (V,E)$

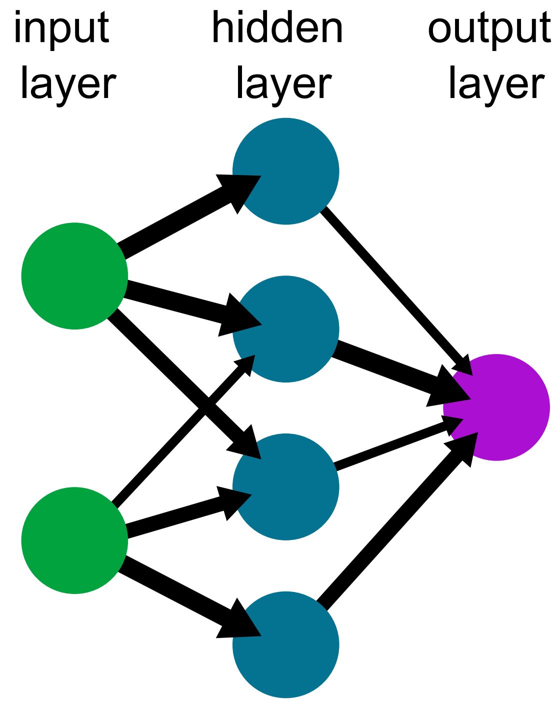
Directed graphs

- A directed graph consists of a set V of vertices (aka nodes or points) and a set $E \subseteq V x V$ of edges. If $(a,b) \in E$, then a is the initial vertex and b is the terminal vertex of the edge (a,b). An edge of the form (a,a) is a loop. Edges are drawn as arrows from their initial to their terminal vertex.
- Between 2 vertices there may only be 1 edge.

Example

- (natural or artificial) neural networks
- Vertices: “neurons”
- Edges: “synapses”
- Well represented by a directed graph because the flux of information has a direction

A simple neural network

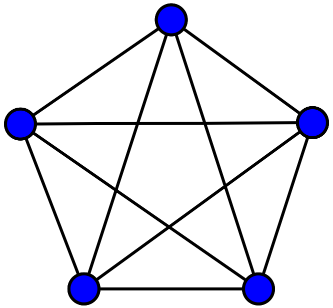


Undirected graphs

- Edges are non-directional: an edge connects 2 vertices (called the endpoints) without distinguishing between an initial vertex and a terminal vertex.
- Edges connecting a vertex with itself are called loops.
- Between 2 vertices there may only be 1 edge.
- Useful to represent “connections”

Example

- How many handshakes happen at a meeting with 5 people?
- Count the edges!



By David Benbennick wrote this file. - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=511709

- Directed and undirected graphs are also called simple graphs to mark the distinction with......

Directed multigraphs

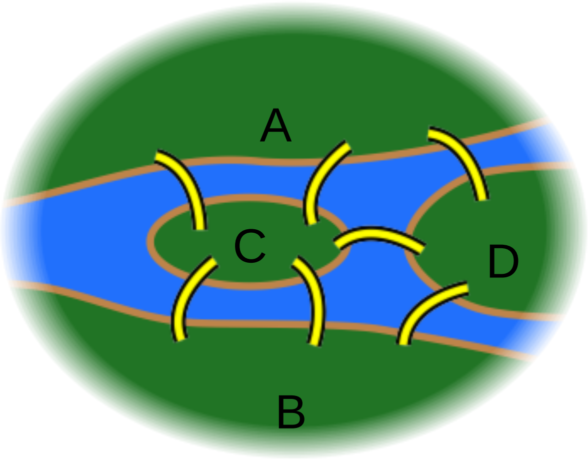
- Edges are directional (as in directed graphs): each edge has an initial vertex and a terminal vertex.
- Edges for which the initial and terminal vertex are the same are called loops.
- The same pair of vertices may be connected by multiple edges.
- Useful to represent a “flux” in which the number and identity of “connection channels” matter

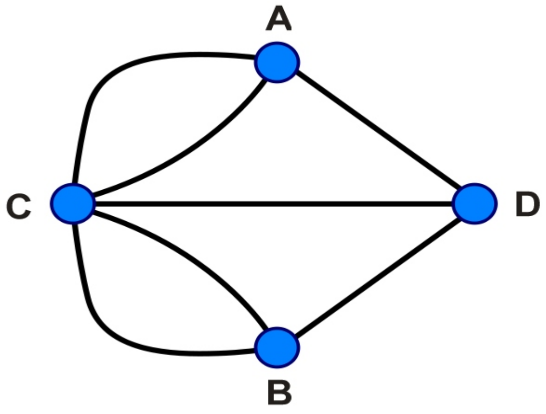
Undirected multigraphs

- Edges are non-directional: an edge connects 2 vertices (called the endpoints) without distinguishing between an initial vertex and a terminal vertex.
- Edges connecting a vertex with itself are called loops.
- The same pair of vertices may be connected by multiple edges.
- Useful to represent "connections" in which the number and identity of "connection channels" matter

Example

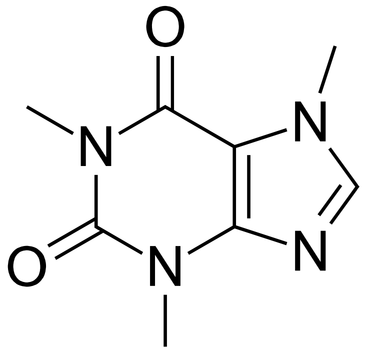
- Königsberg bridges (the problem which founded topology):
- The city of Königsberg in Prussia was set on both sides of a River, and included two islands which were connected to each other, or to the two mainland portions of the city, by seven bridges. The problem was to devise a walk through the city that would cross each of those bridges once and only once.





Example

- Graph representation of molecules



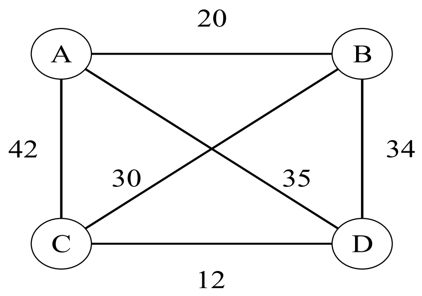
Molecular structure of caffeine.
By Vaccinationist - Own work, based on PubChem,
Public Domain, https://commons.wikimedia.org/w/index.php?curid=54417143

Weighted (multi)graphs

- A number (the weight) is assigned to each edge. The weight might represent lengths, or any other quantity associated with the edge.
- Weights can be applied to directed or undirected graphs or multigraphs

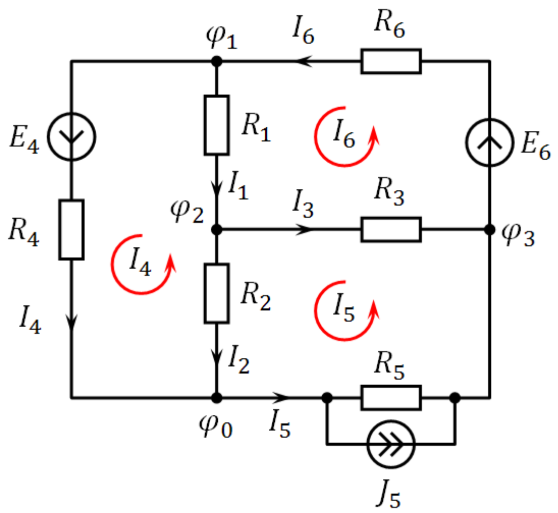
Example

- Travelling salesman problem: Given a list of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?
- weighted undirected graph: vertices = cities, edges = routes, labels = distances



By Sdo - self-made using xfig, CC BY-SA 2.5, https://commons.wikimedia.org/w/index.php?curid=715485

- Electric circuits: weighted directed multigraph: vertices = intersections, edges = actual wires, labels = electric currents



By AdmiralHood - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=17509343

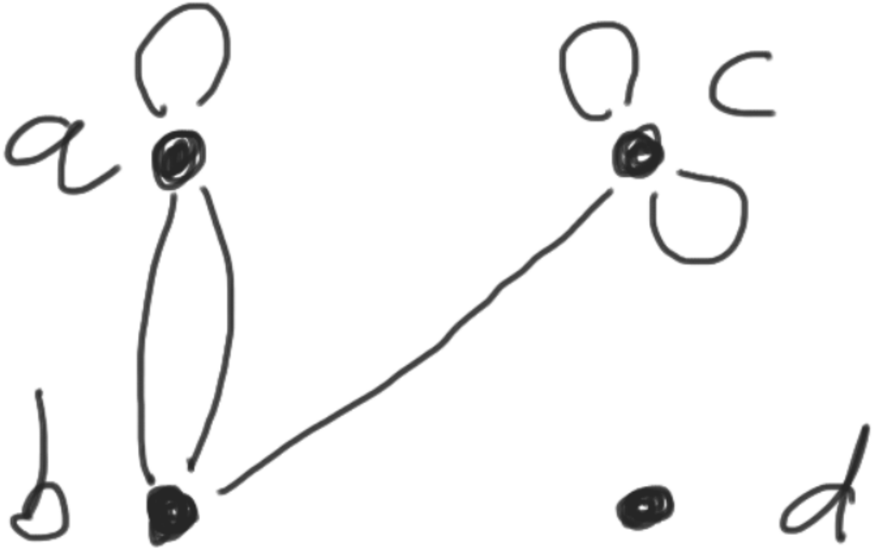
Degrees in undirected graphs

- Two vertices in an undirected (multi)graph G = (V,E) are called adjacent (or neighbours of each other) if there is an edge connecting them.
- The neighbourhood of a vertex v is the set N(v) of all vertices adjacent to v. If A is a subset of V, the neighbourhood of A is the set N(A) of all all vertices adjacent to at least one v∈A, that is:

$$N(A) = \bigcup_{v \in A} N(v)$$

- The degree of a vertex v in an undirected (multi)graph (denoted deg(v)) is the number of edges having v as endpoint (counting loops twice)

Example



- deg(a) = 4; N(a) = {a,b}
- deg(b) = 3; N(b) = {a,c}

loop counts as 2.

- deg(c) = 5; N(c) = {b,c}
- deg(d) = 0; N(d) = Ø
- (a and c are neighbours of themselves thanks to the loops)

Theorem

- Handshaking Theorem: for any undirected (multi)graph G = (V,E),

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

- Proof: each edge has 2 endpoints, so the sum of all the vertex degrees is twice the number of edges.
- Corollary: the sum of the degrees of the vertices of an undirected (multi)graph is even

Example consequences

- If at a meeting of 19 people everybody shakes hands with everybody else (only once), how many handshakes happen?
- SOL: introduce a graph with vertices V = {people at the meeting}, edges E = {handshakes}.
- There are 19 vertices; each is connected to any other vertex via 1 edge. So each vertex has degree 18.
- By the handshaking theorem, $2|E| = 19 \cdot 18$
- Therefore $|E| = 19 \cdot 9 = 171$.

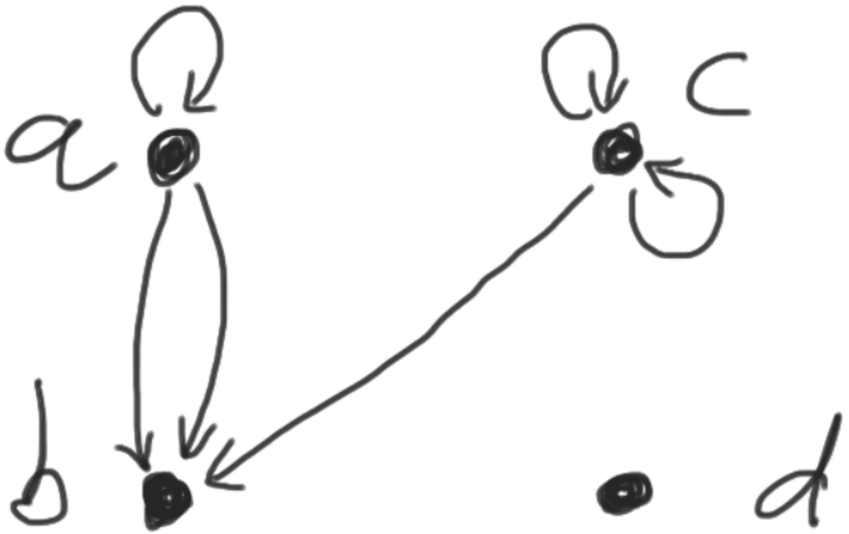
Example consequences

- Can there be a (multi)graph with 7 vertices, all of degree 3?
  - No: if such a graph existed, the sum of the degrees of the vertices would be $3 \cdot 7 = 21$ which is odd.
- Can there be a (multi)graph in which the number of vertices of odd degree is odd?
  - No: if such a graph existed, the sum of the degrees of its vertices would be odd [why?].
- So a (multi)graph can only have an even number of vertices of odd degree.

Degrees in directed graphs

- in a directed (multi)graph,
  - the in-degree of a vertex v, denoted $\deg^-(v)$, is the number of edges having v as terminal vertex.
  - The out-degree of a vertex v, denoted $\deg^+(v)$, is the number of edges having v as initial vertex.
  - Note that a loop at a vertex contributes 1 to both the in-degree and the out-degree of the vertex.

Example



- $deg^{-}(a) = 1$; $deg^{+}(a) = 3$
- $deg^{-}(b) = 3$; $deg^{+}(b) = 0$
- $deg^{-}(c) = 2$; $deg^{+}(b) = 3$
- $deg^{-}(d) = 0$; $deg^{+}(d) = 0$

Theorem

- For any directed (multi)graph G = (V,E),

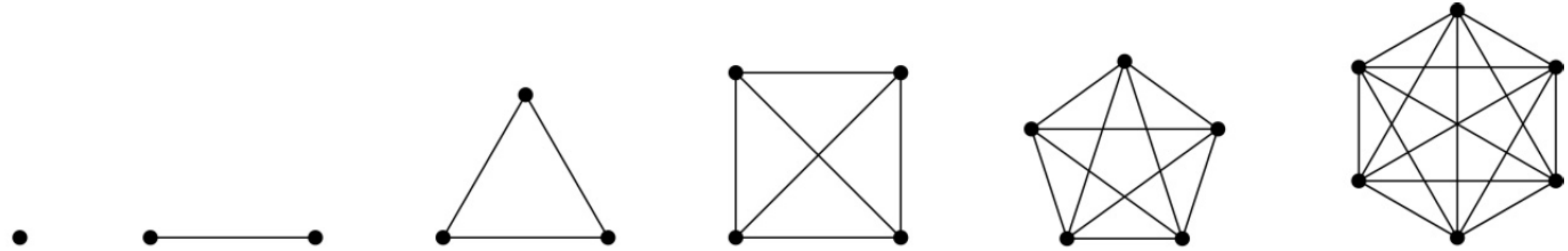
$$|E| = \sum_{v \in V} deg^{-}(v) = \sum_{v \in V} deg^{+}(v)$$

- Proof: Each edge has precisely 1 initial vertex and precisely 1 terminal vertex, so the number of edges matches the number of terminal vertices (1^{st} sum) and the number of initial vertices (2^{nd} sum).

2. SOME SPECIAL GRAPHS

Complete graphs

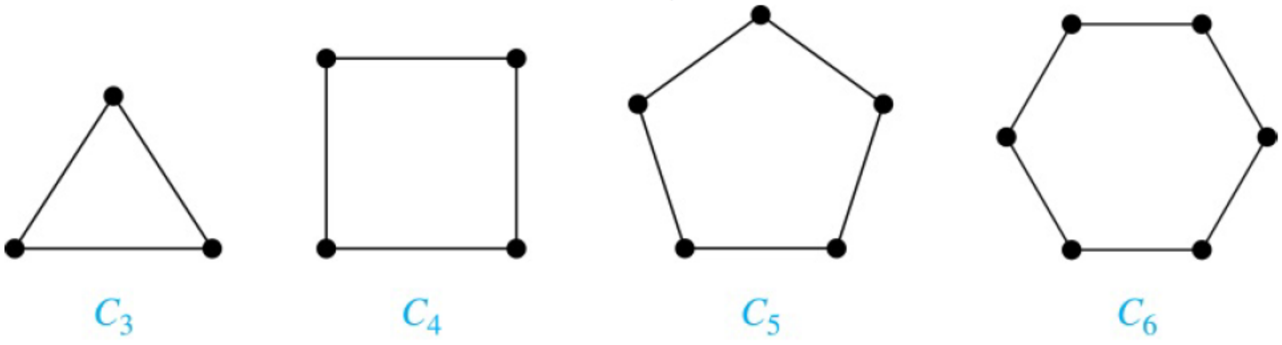
- A complete graph on n vertices, denoted by K_n, is an undirected (simple) graph in which there is exactly one edge between each pair of distinct vertices.



K_1 K_2 K_3 K_4 K_5 K_6

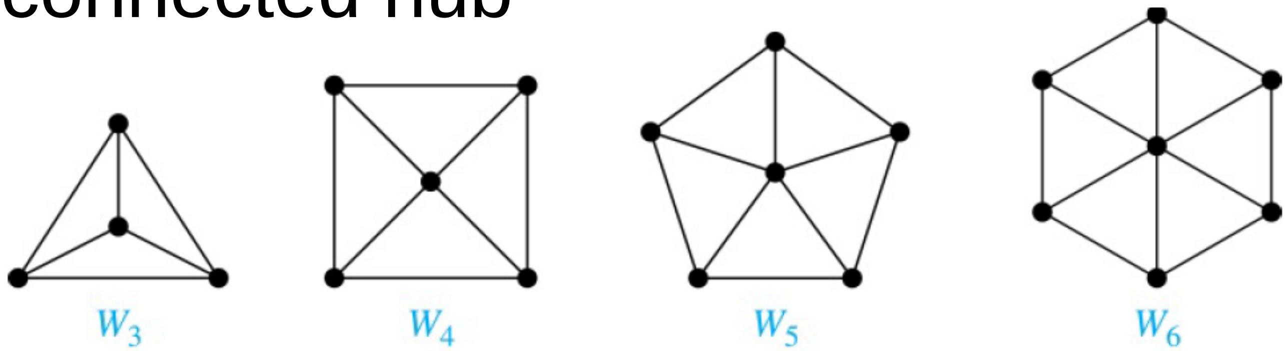
Cycles and wheels

- A cycle C_n ($n \geq 3$) is an undirected (simple) graph made of n vertices, in which each vertex is connected to the previous and the following one, and the last vertex is connected with the 1st.
- Used to model local area networks



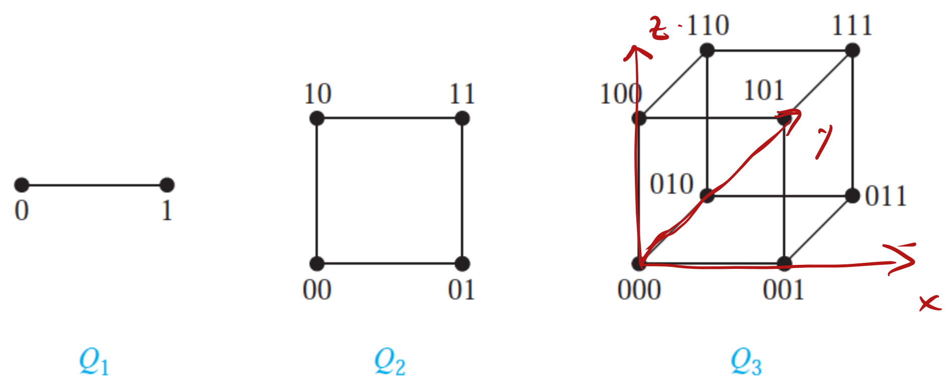
Cycles and wheels

- A wheel W_n ($n \geq 3$) is an undirected (simple) graph built adding one additional vertex to the cycle C_n and connecting the new vertex to all the other vertices.
- Used to model local area networks with a central highly-connected hub



Cubes

- An n-cube, or n-dimensional hypercube, is a graph Q_n whose vertices can be associated to the 2^n bit strings of length n. Two vertices are adjacent if and only if the bit strings that they represent differ in exactly one bit.
- Used to arrange several microprocessors in parallel computing



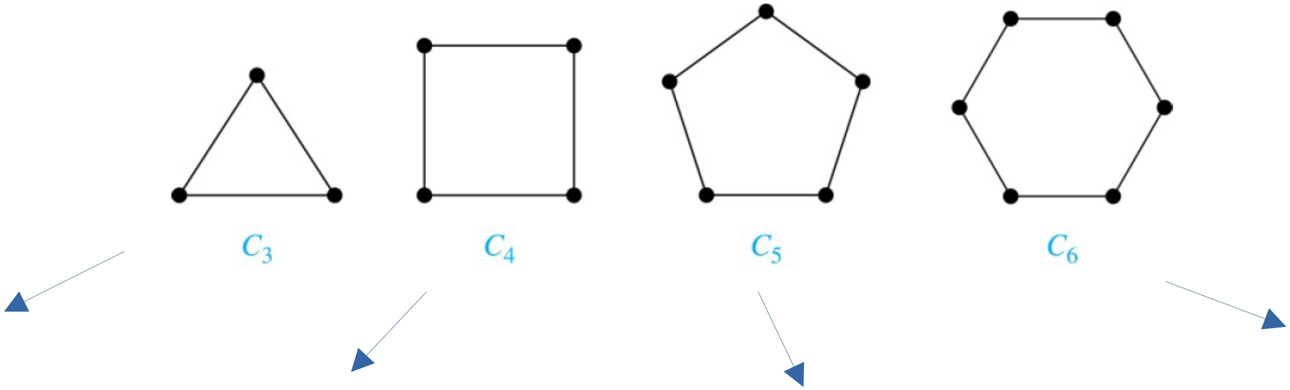
Bipartite graphs

二分体

- An undirected simple graph G = (V,E) is bipartite if V can be partitioned into 2 (disjoint) subsets A and B such that every edge connects a vertex in A and a vertex in B. That is, there are no edges between either two vertices in A or two vertices in B.
- Graphically, an undirected simple graph is bipartite if we can colour its vertices with 2 colours in such a way that there are no edges between vertices of the same colour.

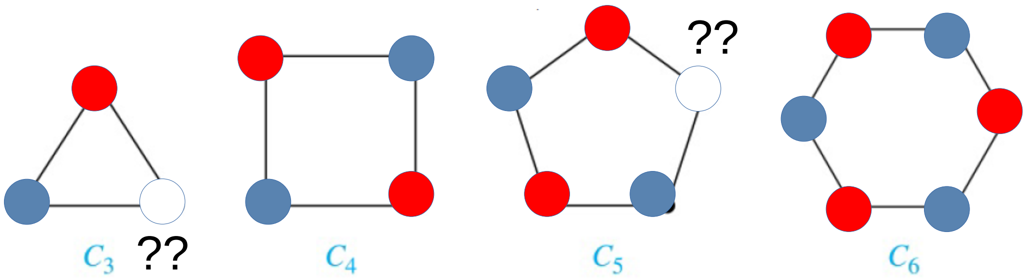
Bipartite graphs

- EX: which of the following is bipartite?



Bipartite graphs

- EX: which of the following is bipartite?



NO!

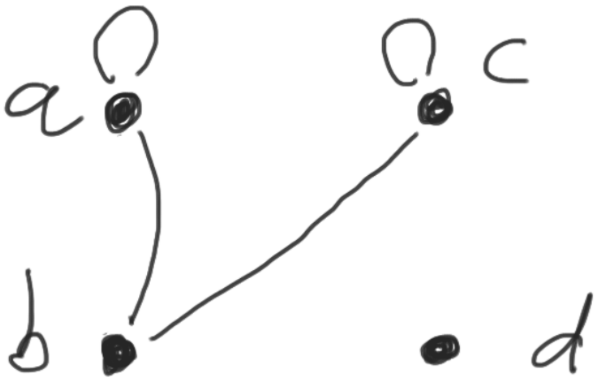
YES!

NO!

YES!

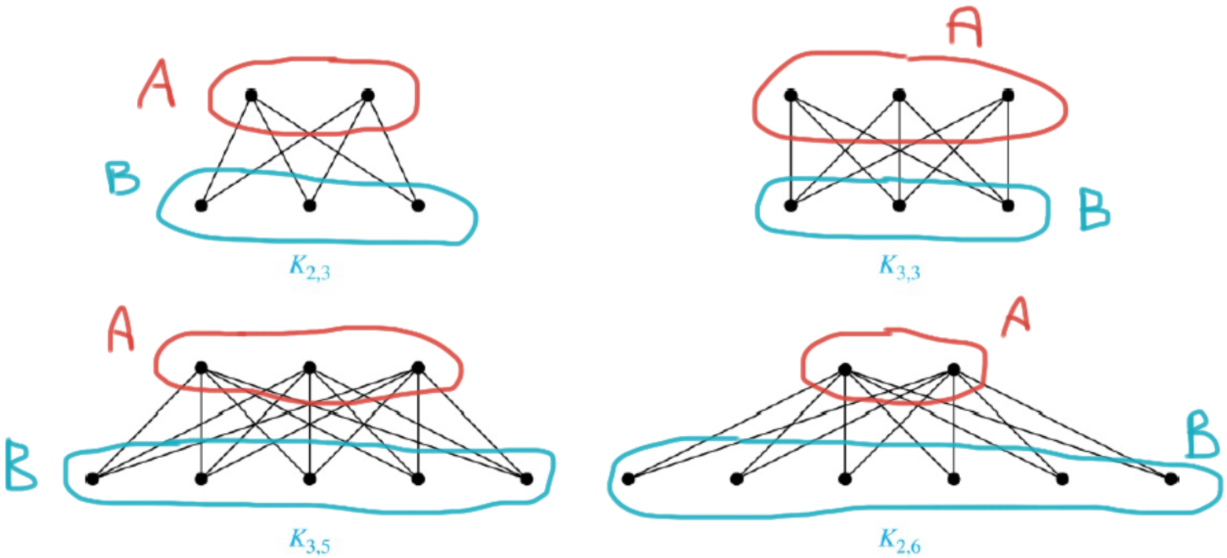
Bipartite graphs

- A graph with loops cannot be bipartite: no matter what colour you assign to a vertex with a loop, the loop would connect 2 vertices with the same colour (the same vertex!).



Bipartite graphs

- A complete bipartite graph $K_{m,n}$ is a bipartite graph that has its vertex set partitioned into two subsets A of size m and B of size n, such that each vertex in A is connected to all and only the vertices in B. That is, there are no edges between vertices in A or between vertices in B, and there is 1 edge between every vertex in A and every vertex in B.



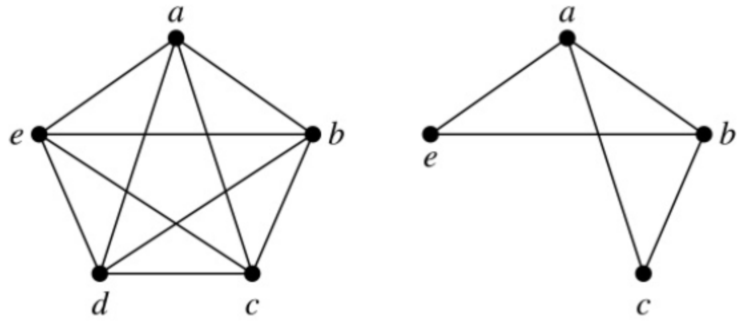
3. OPERATIONS ON GRAPHS

Subgraphs

- A sub(multi)graph of a (multi)graph $G = (V,E)$ is a (multi)graph $H = (W,F)$ with $W \subseteq V$ and $F \subseteq E$. A sub(multi)graph H of G is a proper sub(multi)graph of G if $H \neq G$.
- Let $G = (V;E)$ be a (simple) graph and $W \subseteq V$. The subgraph induced by W is the graph $(W; F)$, where the edge set F contains those edges in E both endpoints of which are in W.

Subgraphs

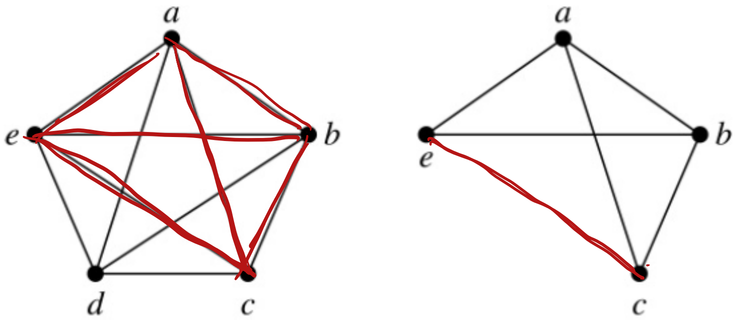
- EX: The picture shows a K_5 and a subgraph on the subset of vertices W = {a,b,c,e}



- Is the subgraph induced by W?

Subgraphs

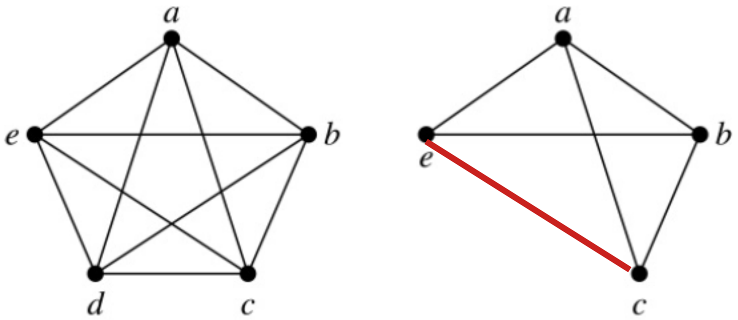
- EX: The picture shows a K_5 and a subgraph on the subset of vertices W = {a,b,c,e}



- Is the subgraph induced by W?
- No, because e and c are both in W, but the subgraph misses the edge between them

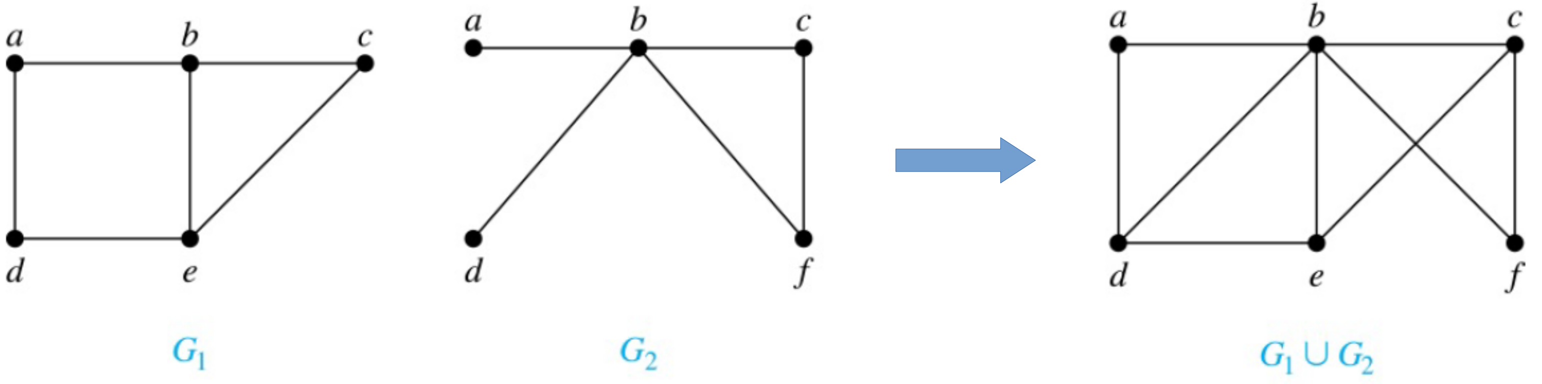
Subgraphs

- The subgraph induced by W is the following (on the right)



Unions of graphs

- The union of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph, denoted $G_1 \cup G_2$, with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$.
- EX:



4. REPRESENTING GRAPHS

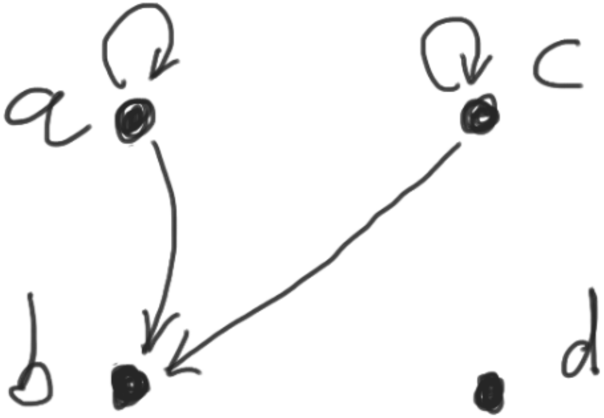
Adjacency lists

- An adjacency list is a table specifying the vertices that are adjacent to each vertex of the graph.
- EX (undirected graph):

vertex	Adjacent vertices
a	a,b
b	a,c
c	b,c
d	/

Adjacency lists

- For directed graphs, we distinguish between initial and terminal vertices.
- EX (directed graph):



Initial vertex	terminal vertices
a	a,b
b	a,c
c	b,c
d	/

Adjacency matrices

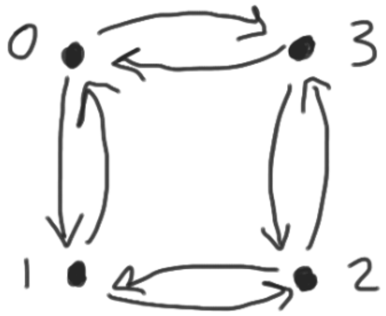
- Let G = (V,E) be a directed graph. Assume |V| = n and choose an ordering $v_1, v_2, \ldots v_n$ of the vertices in V.
- The adjacency matrix of G is the n x n matrix $A_G = [a_{ij}]$ with

$$a_{ij} = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise} \end{cases}$$

- That is, $a_{ij} = 1$ iff there is an edge with initial vertex v_i and terminal vertex v_j
- The adjacency matrix depends on the chosen ordering of the vertices

Adjacency matrices

- EX: let G =



- With the ordering 0,1,2,3

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

- With the ordering 0,2,1,3

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

Adjacency matrices

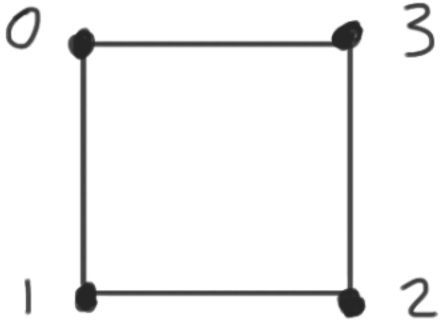
- Let G = (V,E) be an undirected graph. Assume |V| = n and choose an ordering $v_1, v_2, \ldots v_n$ of the vertices in V.
- The adjacency matrix of G is the n x n matrix $A_G = [a_{ij}]$ with

$$a_{ij} = \begin{cases} 1 & \text{if } E \text{ contains an edge between } v_i \text{ and } v_j \\ 0 & \text{otherwise} \end{cases}$$

- Note that the adjacency matrix of an undirected graph is symmetric by construction
- The adjacency matrix depends on the chosen ordering of the vertices

Adjacency matrices

- EX: let G =



- With the ordering 0,1,2,3

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

- With the ordering 0,2,1,3

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

Adjacency matrices

- The previous 2 examples also show that we can think of undirected graphs as directed graphs in which the edge set is a symmetric binary relation, that is, directed graphs with the property that, whenever (a,b) is an edge, (b,a) is an edge as well

Adjacency matrices

- Let $G = (V,E)$ be a directed multigraph. Assume $|V| = n$ and choose an ordering $v_1,v_2,\ldots v_n$ of the vertices in V.
- The adjacency matrix of G is the $n \times n$ matrix $A_G = [a_{ij}]$ with

$$a_{ij} = \text{number of edges with initial vertex } v_i \text{ and terminal vertex } v_j$$

- The adjacency matrix depends on the chosen ordering of the vertices

Adjacency matrices

- Let $G = (V,E)$ be a undirected multigraph. Assume $|V| = n$ and choose an ordering $v_1,v_2,\ldots v_n$ of the vertices in V.
- The adjacency matrix of G is the n x n matrix $A_G = [a_{ij}]$ with

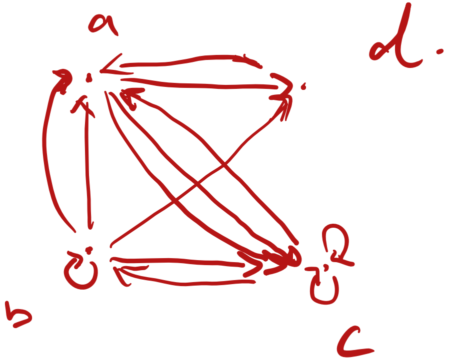
$$a_{ij} = \text{number of edges between vertices } v_i \text{ and } v_j$$

- Note that the adjacency matrix of an undirected multigraph is symmetric by construction
- The adjacency matrix depends on the chosen ordering of the vertices

Adjacency matrices

- EX: construct a graph with adjacency matrix

$$A_G = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 2 & 1 & 2 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$



Adjacency matrices

- EX: construct a graph with adjacency matrix

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

- Preliminary remarks:
  - the matrix is not symmetric, so the graph is directed
  - the matrix contains entries = 2, so it is a multigraph

Adjacency matrices

- EX: construct a graph with adjacency matrix

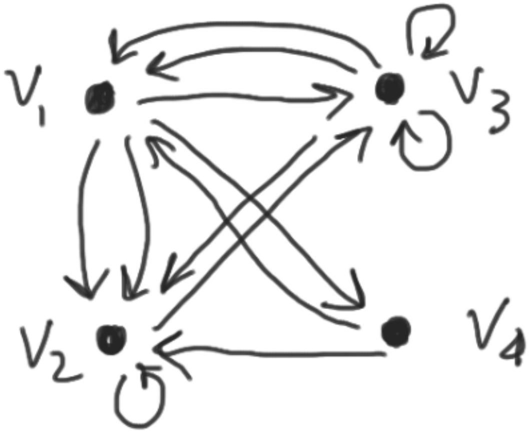
$$A_G = \begin{bmatrix} 0 & 2 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 2 & 1 & 2 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

Adjacency matrices

- EX: construct a graph with adjacency matrix

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

A different order →



Incidence matrices

- Let G = (V,E) be a directed (multi)graph. Assume |V| = n and choose an ordering $v_1, v_2, \ldots v_n$ of the vertices in V. Assume |E| = k and choose an ordering $e_1, e_2, \ldots e_k$ of the edges in E
- The incidence matrix of G is the n x k matrix $M_G = [m_{ij}]$ with

$$m_{ij} = \begin{cases} -1 & \text{if } v_i \text{ is the initial vertex of } e_j \\ 1 & \text{if } v_i \text{ is the terminal vertex of } e_j \\ 0 & \text{if } e_j \text{ is a loop on } v_i \text{, or if } v_i \text{ is unrelated to } e_j \end{cases}$$

- The adjacency matrix depends on the chosen orderings of the vertices and of the edges

Incidence matrices

- Let $G = (V,E)$ be an undirected (multi)graph. Assume $|V| = n$ and choose an ordering $v_1, v_2, \ldots v_n$ of the vertices in V. Assume $|E| = k$ and choose an ordering $e_1, e_2, \ldots e_k$ of the edges in E
- The incidence matrix of G is the n x k matrix $M_G = [m_{ij}]$ with

$$m_{ij} = \begin{cases} 2 & \text{if } e_j \text{ is a loop at } v_i \\ 1 & \text{if } e_j \text{ is not a loop and } v_i \text{ is one of its endpoints} \\ 0 & \text{otherwise} \end{cases}$$

- The adjacency matrix depends on the chosen orderings of the vertices and of the edges

5. PROPERTIES OF GRAPHS

双射：单射且满射

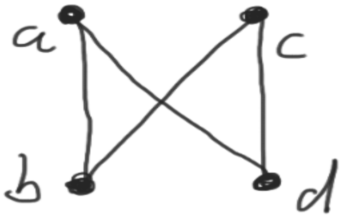
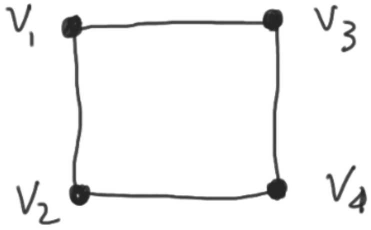
$A \to A_1$ $A \to A_1$
$B \to B_1$ $B \to B_1$
$C \to D_1$ C_1 $C \to C_1$
$D \to C_1$

同构

Graph isomorphisms

- Two simple graphs $G_1 = (V_1,E_1)$ and $G_2 = (V_2,E_2)$ are isomorphic if there is a bijective (双射) function $f : V_1 \to V_2$ with the property that, for any two vertices $a,b \in V_1$, there is an edge from a to b in G_1 if and only if there is an edge from f(a) to f(b) in G_2. Such a function f is called an isomorphism of graphs.
- Sometimes it is easy to show two graphs are not isomorphic finding a property, preserved by isomorphism (a so-called graph invariant), that only one of the two graphs has. Examples of graph invariants: number of vertices, number of edges, degree sequence (list of the degrees of the vertices in non-increasing order).

Graph isomorphisms

- EX: The 2 graphs  and  are isomorphic

- An isomorphism is given by

  $f : \{a,b,c,d\} \to \{v_1,v_2,v_3,v_4\}$, $f(a)=v_1$, $f(b)=v_2$, $f(c)=v_4$, $f(d)=v_3$.

  - a is adjacent to b and d, and $f(a)=v_1$ is adjacent to $f(b)=v_2$ and $f(d)=v_3$.
  - b is adjacent to a and c, and $f(b)=v_2$ is adjacent to $f(a)=v_1$ and $f(c)=v_4$.
  - Etc.

Graph isomorphisms

- EX: The following 2 graphs are not isomorphic

- In fact, the in-and out-degrees don't match:
  - The sequence of out-degrees for G is 0,0,2,2 (corresponding to the vertex ordering d,b,a,c)
  - The sequence of out-degrees for G is 0,1,1,2 (corresponding to the vertex ordering r,q,s,p)

Graph isomorphisms

- Checking whether two graphs are isomorphic by brute force is long and painful.
- The best known algorithms for determining whether two graphs are isomorphic have exponential worst-case time complexity (in the number of vertices)...
- ... but linear average-case time complexity.
- Graph isomorphism is a problem of special interest because it is NP, but it is not known to be either NP-complete or not.

Application of graph isomorphisms

- When a new chemical compound is synthesized, it is checked against a database of molecular graphs to determine whether the graph representing the new compound is isomorphic to the graph of an already known compound.
- Graph isomorphisms are used in electric circuit analysis to check
  - whether a particular layout of a circuit corresponds to the design's original schematics.
  - whether a chip from one producer includes the intellectual property of another producer.

Paths

- Informally, a path in a graph is a sequence of edges that begins at a vertex, travels from vertex to vertex along edges of the graph, and ends up at another vertex.
- BEWARE: other sources (e.g. Wikipedia) use a different terminology, so check which definitions are adopted in the book/article/project you have at hands

Paths

- A path of length n in an undirected (multi)graph is a sequence of n edges $(e_1,e_2,...,e_n)$ such that, for all $i=1,2,...,n$, e_i and e_{i+1} have a common endpoint. In other words, there exists a sequence $(v_0, v_1, . . . , v_n)$ of vertices such that, for $i = 1, . . . , n$, e_i has the endpoints v_{i-1} and v_i. In this case we speak of a path from v_0 to v_n. If the graph is simple, we can unambiguously identify a path with the sequence of vertices $(v_0, v_1, . . . , v_n)$ it passes through.
- A circuit is a path of length ≥ 0 from a vertex to itself.
- A path or circuit is simple if it does not have repeated edges.

Paths

- A path of length n in a directed (multi)graph is a sequence of n edges $(e_1,e_2,\ldots,e_n)$ such that, for all $i=1,2,\ldots,n$, the terminal vertex of e_i coincides with the initial vertex of e_{i+1}. In other words, there exists a sequence $(v_0, v_1, \ldots, v_n)$ of vertices such that, for $i = 1, \ldots, n$, e_i has initial vertex v_{i-1} and terminal vertex v_i. In this case we speak of a path from v_0 to v_n. If the graph is simple, we can unambiguously identify a path with the sequence of vertices $(v_0, v_1, \ldots, v_n)$ it passes through.
- A circuit is a path of length ≥ 0 from a vertex to itself.
- A path or circuit is simple if it does not have repeated edges.

Circuit: 

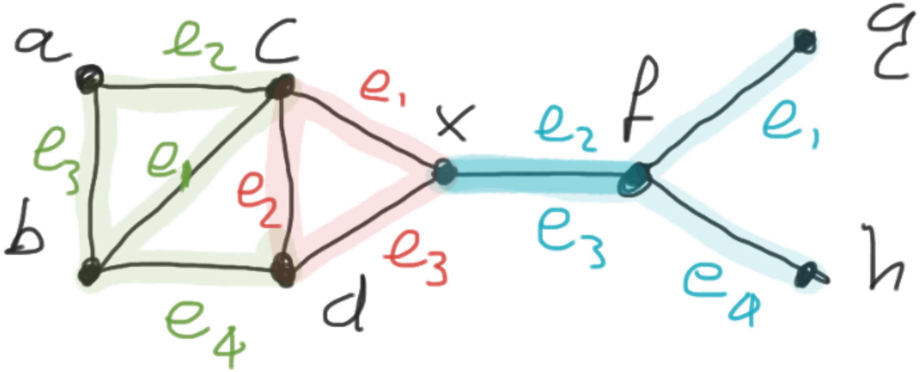
Simple circuit:  else: 

Paths

- BEWARE: other sources (e.g. Wikipedia) use a different terminology, so check which definitions are adopted in the book/article/project you have at hands

Examples

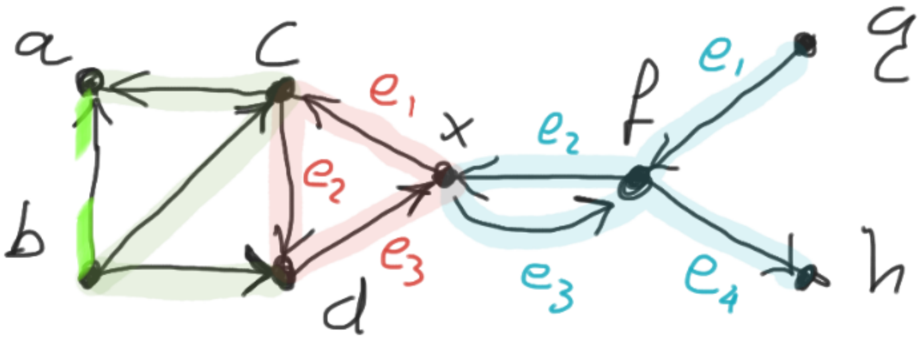
- In the following undirected simple graph



- (g,f,x,f,h) is a path of length 4 which is not a circuit (g≠h) nor simple (it goes through the edge xf twice)
- (b,c,a,b,d) is a simple path of length 4
- (x,c,d,x) is a simple circuit of length 3

Examples

- In the following directed simple graph

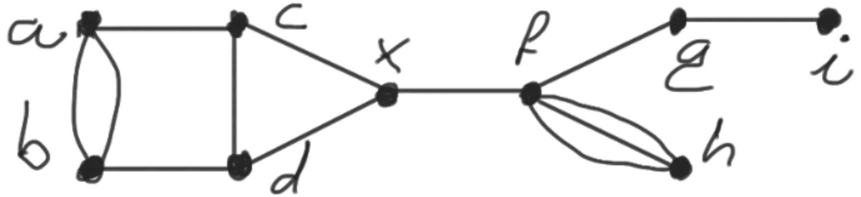


- (g,f,x,f,h) is a simple path of length 4 which is not a circuit (g≠h) (note that (f,x) and (x,f) count as distinct edges)
- (b,c,a,b,d) is not a path ((a,b) is not an edge)
- (x,c,d,x) is a simple circuit of length 3

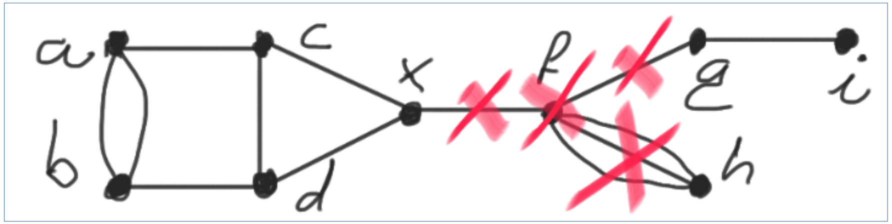
Connectivity (undirected)

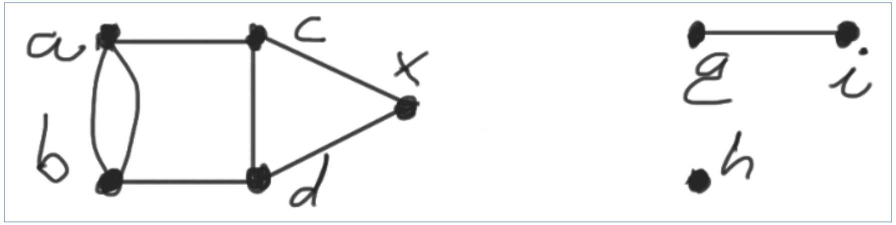
- An undirected (multi)graph is connected if between every pair of vertices there is (at least) one path. Otherwise the graph is disconnected.
- The connected components of a (multi)graph are its maximal (w.r.t. inclusion) connected subgraphs (if a graph is connected, its only connected component is itself).
- We say that we disconnect a (multi)graph when we remove vertices or edges to produce a subgraph which is disconnected.

Examples

- This is a connected multigraph: 

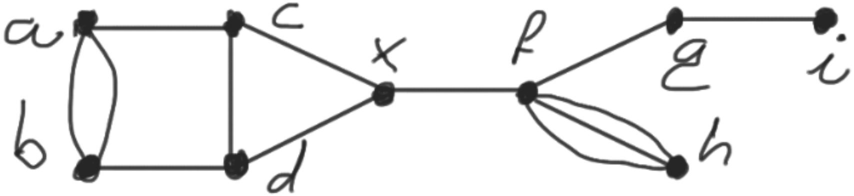
- If we remove the vertex f (and therefore all the edges with endpoint f), we disconnect the multigraph:



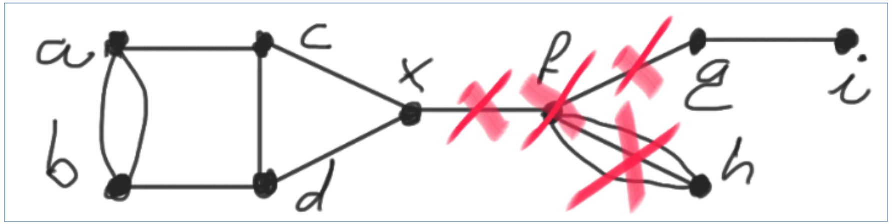


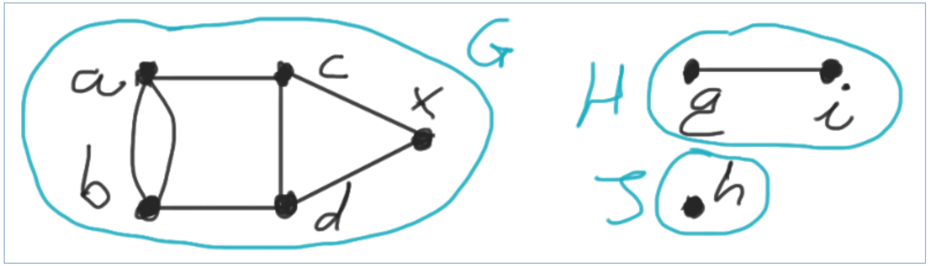
Examples

- This is a connected multigraph:



- If we remove the vertex f (and therefore all the edges with endpoint f), we disconnect the multigraph:





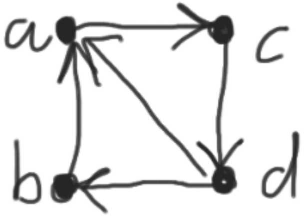
- The new graph has 3 connected components: the subgraphs G, H, J

Connectivity (directed)

- A directed (multi)graph is strongly connected if between every pair of vertices there is (at least) one path. The graph is weakly connected if the underlying undirected graph, which is the undirected graph obtained by ignoring the directions of the edges of the directed graph, is connected. Otherwise the graph is disconnected.
- The strongly connected components of a directed (multi)graph are its maximal (w.r.t. inclusion) strongly connected subgraphs. Its weakly connected components are the connected components of its underlying undirected graph.

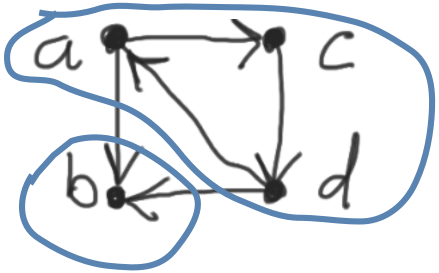
Examples

- The directed graph



is strongly connected: starting from any vertex we can reach any other vertex going around the square (although this may not give the shortest path).
- Its only strongly connected component is itself.

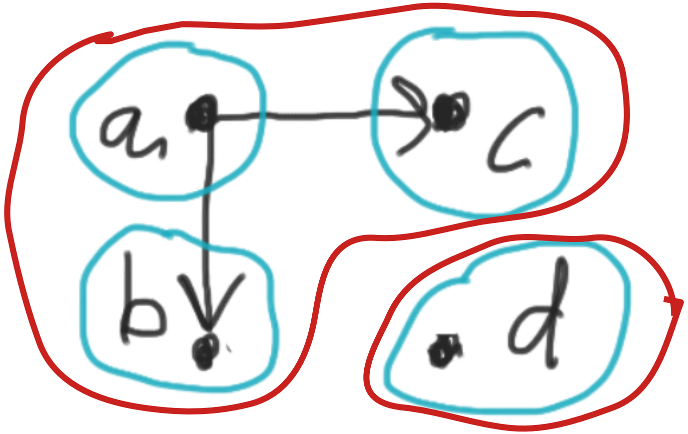
- The directed graph



is weakly but not strongly connected: e.g. there is no path from b to any vertex.
- It has 2 strongly connected components, circled in blue.
- It has 1 weakly connected component (the graph itself).

Examples

- The directed graph



is neither weakly not strongly connected. It has 4 strongly connected components (the isolated vertices with no edges, circled in blue). It has 2 weakly connected components: the subgraph induced by {a,b,c} and the isolated vertex d (circled in red).

Counting paths between vertices

- Awesome theorem: Let G be a directed or undirected (multi)graph on n vertices with adjacency matrix A (with respect to the vertex ordering $v_1,\ldots,v_n$).

  For any integer $k>0$, the number of distinct paths of length k from v_i to v_j is equal to the (i,j) entry of A^k.
- The proof is by induction on k.

Counting paths between vertices

- EX: the directed multigraph

Has adjacency matrix
(w.r.t. the ordering a,b,c)

$$A = \begin{array}{c} \\ a \\ b \\ c \end{array} \begin{array}{c} \begin{array}{ccc} a & b & c \end{array} \\ \begin{bmatrix} 1 & 0 & 2 \\ 1 & 0 & 1 \\ 1 & 2 & 0 \end{bmatrix} \end{array}$$

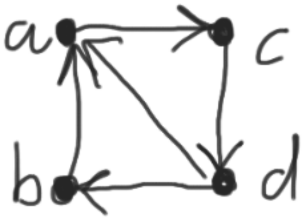
- Since $A^3 = \begin{bmatrix} 9 & 4 & 10 \\ 6 & 4 & 6 \\ 7 & 8 & 6 \end{bmatrix}$, there are 4 circuits of length 3 starting (and ending) at b.

Euler paths and circuits

- How did Leonard Euler solve the Königsberg bridges problem?
- Let G be a directed or undirected (multi)graph.
- An Euler path in G is a simple path containing every edge of G.
- An Euler circuit in a graph G is a simple circuit containing every edge of G.

Examples

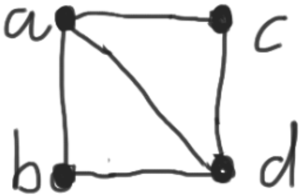
- The directed graph



has exactly 1 Euler path (d,a,c,d,b,a) and no Euler circuits

Euler circuit : 1. start & end at same vertex (circuit).
2. simple
3. contain every edge.

- The undirected graph



has 12 Euler paths (you can check that they ought to start at either a or d and end at the other) and no Euler circuit

Conditions for Euler paths and circuits

277

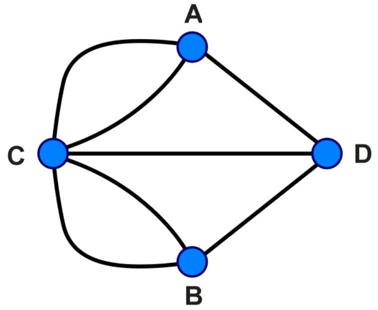
- An undirected graph has an Euler cycle if and only if every vertex has even degree, and all of its vertices with nonzero degree belong to a single connected component.
- In fact, each time the circuit passes through a vertex, it contributes +2 to the vertex's degree (this is true for the start vertex as well, since it is also the end vertex).
- The same reasoning shows that the start vertex and the end vertex of an Euler path have odd degree, while every other vertex has even degree. That is, If a graph has an Euler path (but not an Euler circuit), then all of its vertices with nonzero degree belong to a single connected component, and exactly two of its vertices have an odd degree.

Conditions for Euler paths and circuits

- A directed graph has an Euler cycle iff all of its vertices with nonzero degree belong to a single strongly connected component, and every vertex has equal in-degree and out-degree.
- A directed graph has an Euler path (but not an Euler cycle) iff all of its vertices with nonzero degree belong to a single strongly connected component, and exactly 1 vertex has (out-degree)−(in-degree) = 1, and exactly 1 vertex has (in-degree) − (out-degree) = 1

Königsberg, solved

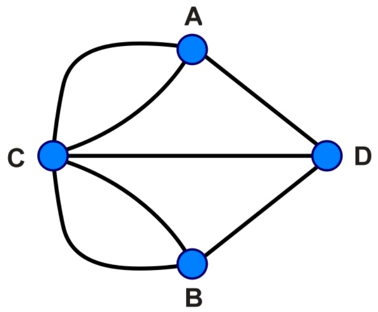
- Is it possible to walk over all 7 bridges exactly once and return at the starting point?



- That is, is there an Euler circuit in the Königsberg multigraph?

Königsberg, solved

- Is it possible to walk over all 7 bridges exactly once and return at the starting point?



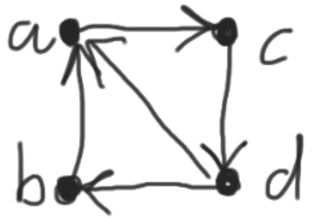
- That is, is there an Euler circuit in the Königsberg multigraph?
- NO! all 4 vertices have odd degrees, so there is not even an Euler *path*.

Hamilton paths and circuits

- (Euler) : (edges) = (Hamilton) : (vertices)
- Let G be a directed or undirected (multi)graph.
- A Hamilton path in G is a simple path passing through every vertex of G exactly once.
- A Hamilton circuit in G is a simple circuit passing through the start vertex exactly twice (at the beginning and at the end) and through every other vertex of G exactly once.

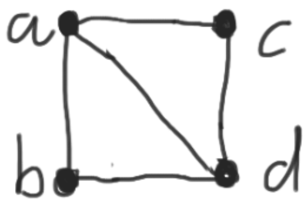
Examples

- The directed graph



- has 4 Hamilton circuits: (a,c,d,b,a), (c,d,b,a,c), (d,b,a,c,d), (b,a,c,d,b) and 4 additional Hamilton paths (a,c,d,b), (c,d,b,a), (d,b,a,c), (b,a,c,d)

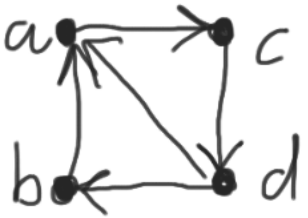
- The undirected graph



- has 8 Hamilton circuits (2 starting at each vertex, and running along the perimeter of the square in either direction) and 8 additional Hamilton paths obtained removing the last edge from a circuit.

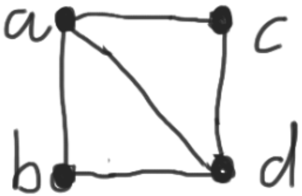
Examples

- The directed graph



- has 4 Hamilton circuits: (a,c,d,b,a), (c,d,b,a,c), (d,b,a,c,d), (b,a,c,d,b) and 4 additional Hamilton paths (a,c,d,b), (c,d,b,a), (d,b,a,c), (b,a,c,d)

- The undirected graph



- has 8 Hamilton circuits (2 starting at each vertex, and running along the perimeter of the square in either direction) and 8 additional Hamilton paths obtained removing the last edge from a circuit.

Conditions for Hamilton paths and circuits

- Unlike for Euler paths and circuits, no simple necessary and sufficient conditions are known for the existence of Hamilton paths and circuits. However, various sufficient conditions have been proved.
- Theorem [G. A. Dirac]: If G is an undirected simple graph with $n \geq 3$ vertices such that the degree of every vertex in G is $\geq n/2$, then G has a Hamilton circuit.
- Theorem [Ghouila-Houri]: If G is a strongly connected directed simple graph with n vertices such that each vertex has both out-degree and in-degree at least $n/2$, then G has a Hamilton circuit.

Travelling salesman problem

- Recall the problem: Given a list of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?
- The problem is represented as a weighted undirected graph:
  - vertices = cities,
  - edges = routes,
  - weights = distances
- This problem reduces to finding a Hamilton circuit such that the total sum of the weights of its edges is as small as possible.