Introduction to graph theory

Giacomo Como, DISMA, Politecnico di Torino

Fabio Fagnani, DISMA, Politecnico di Torino

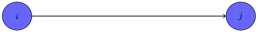
The skeleton of a network: the graph

Elements of graph theory

- Set of *nodes* $\mathcal{V}$ which represent the units participating in the network (e.g. computers, sensors, web pages, companies, individuals, biological entities).

Elements of graph theory

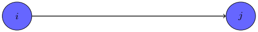
- Set of *nodes* $\mathcal{V}$ which represent the units participating in the network (e.g. computers, sensors, web pages, companies, individuals, biological entities).
- Set of *links* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The presence of a link (i, j) may have different interpretation depending on the applicative set-up:
  - node i influences node j;
  - node i 'sees' node j, i has access to the state of j;
  - flow can take place from i to j.



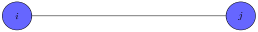
Elements of graph theory

- Set of *nodes* $\mathcal{V}$ which represent the units participating in the network (e.g. computers, sensors, web pages, companies, individuals, biological entities).

- Set of *links* $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The presence of a link (i, j) may have different interpretation depending on the applicative set-up:
  - node i influences node j;
  - node i 'sees' node j, i has access to the state of j;
  - flow can take place from i to j.

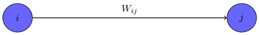


- $(i, j), (j, i) \in \mathcal{E}$



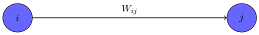
Elements of graph theory

- *Weights* W_{ij} associated to each link: the strength of a connection, the conductance or capacity of the link.



Elements of graph theory

- *Weights* W_{ij} associated to each link: the strength of a connection, the conductance or capacity of the link.

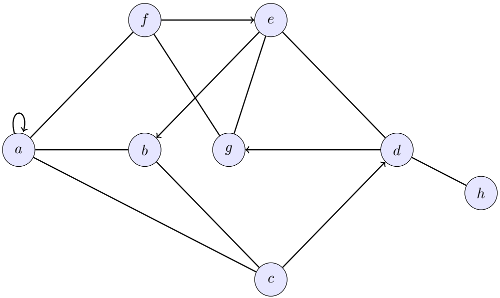


NOTATION: $G = (\mathcal{V}, \mathcal{E}, W)$

If $W_{ij} = 1$ for every $(i, j) \in \mathcal{E}$: $G = (\mathcal{V}, \mathcal{E})$ *unweighted graph*

Elements of graph theory

How graphs are represented:



The birth of graph theory

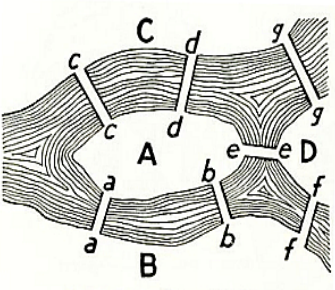
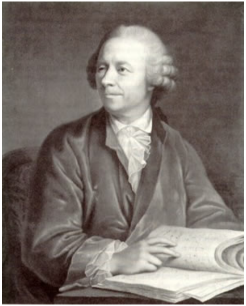


FIGURE 98. *Geographic Map: The Königsberg Bridges.*

Leonhard Euler (1707 - 1783)



The birth of graph theory

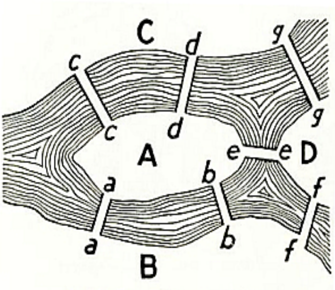
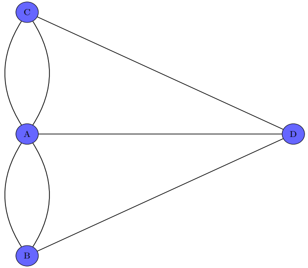


FIGURE 98. *Geographic Map: The Königsberg Bridges.*

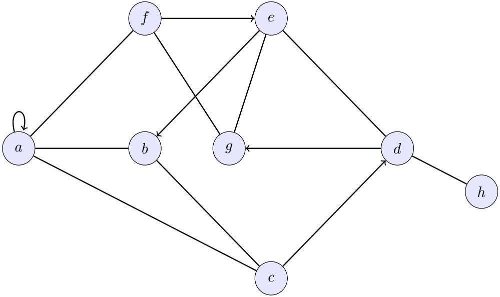


Neighborhood, sink, source

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$$

- The *out-neighborhood*: $\mathcal{N}_i = \{j \in \mathcal{V} \,|\, (i,j) \in \mathcal{E}\}$;
- The *in-neighborhood*: $\mathcal{N}_i^- = \{j \in \mathcal{V} \,|\, (j,i) \in \mathcal{E}\}$;
- If $(i,i) \in \mathcal{E})$ *(self-loop)*, then, $i \in \mathcal{N}_i$ and $i \in \mathcal{N}_i^-$
- $\mathcal{N}_i = \emptyset, \{i\} \Rightarrow i$ is a *sink*;
- $\mathcal{N}_i^- = \emptyset, \{i\} \Rightarrow i$ is a *source*

Example



$\mathcal{N}_d = \{e, g, h\}$, $\mathcal{N}_d^- = \{c, e, h\}$.
No sinks, no sources.

Degrees

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$

- The *out-degree*: $w_i = \sum_{j \in \mathcal{V}} W_{ij}$;
- The *in-degree*: $w_i^- = \sum_{j \in \mathcal{V}} W_{ji}$.
- $\sum_i w_i = \sum w_i^- = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} W_{ij}$

- $\mathcal{G}$ *balanced* if $w_i = w_i^-$ for all $i \in \mathcal{V}$
- $\mathcal{G}$ *regular* if balanced and $w_i = w_j$ for every $i, j \in \mathcal{V}$

If $\mathcal{G}$ unweighted:

- $w_i = |\mathcal{N}_i|$, $w_i^- = |\mathcal{N}_i^-|$.
- $\sum_i w_i = \sum w_i^- = |\mathcal{E}|$

Connectivity issues

Walks, paths, circuits, cycles

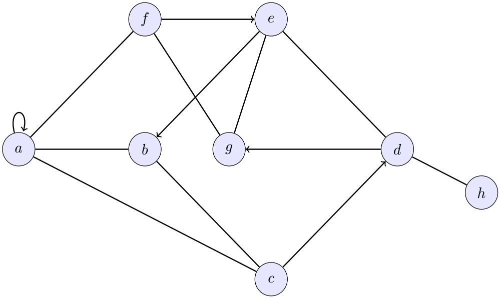
- *walk* from node i to node j: $\gamma = (i = i_0, i_1, \ldots, j = i_l)$ s. t. $(i_{h-1}, i_h) \in \mathcal{E}$ for all $h = 1, \ldots, l$. l *length* of the walk.
- *concatenation* of walks: $\gamma^1 = (i, i_1, \ldots, j)$ and $\gamma^2 = (j, j_1, \ldots, k) \to$, $\gamma^1\gamma^2 = (i, i_1, \ldots, j, , j_1, \ldots, k)$.
- j is *reachable* from i is there exists a walk from i to j.
- *path*: a walk $i_0, i_1, \ldots, i_l$ such that $i_h \neq i_k$ for all $0 \leq h < k \leq l$, except for possibly $i_0 = i_l$.
- *trail*: a walk $i_0, i_1, \ldots, i_l$ with all distinct edges.
- *circuit*: a closed trail, namely $i_0, i_1, \ldots, i_l$ with $i_0 = i_l$.
- *cycle*: a closed path of length $l \geq 3$.

Walks, paths, circuits, cycles

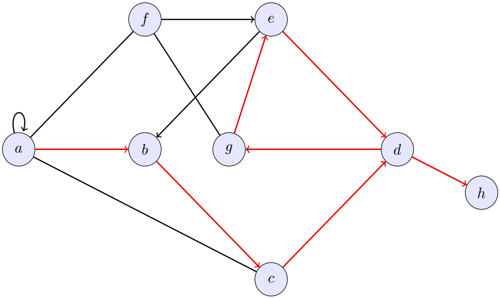
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$

- *acyclic* if no cycles.
- *directed acyclic graph DAG* if no closed walks
- *distance* on $\mathcal{V}$: $i, j \in \mathcal{V}$, $d_{\mathcal{G}}(i,j)$ is the length of the shortest path from i to j in $\mathcal{G}$ (with the convention that $d_{\mathcal{G}}(i,j) = +\infty$ if no such path exists).
- *geodesic path* is a path from i to j of minimal length
- *diameter* of $\mathcal{G}$ is $\mathrm{diam}(\mathcal{G}) := \max_{i,j} d_{\mathcal{G}}(i,j)$.
- *strongly connected* if for all i and j, i is reachable from j.

Example

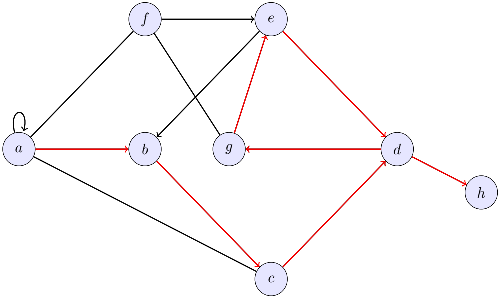


Example



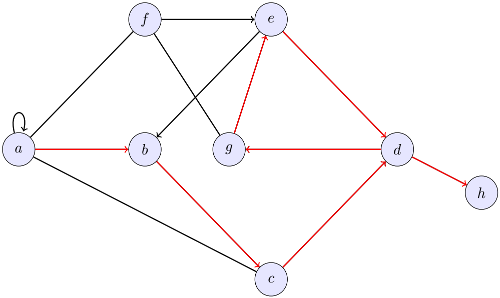
$$\gamma = (a, b, c, d, g, e, d, h)$$

Example



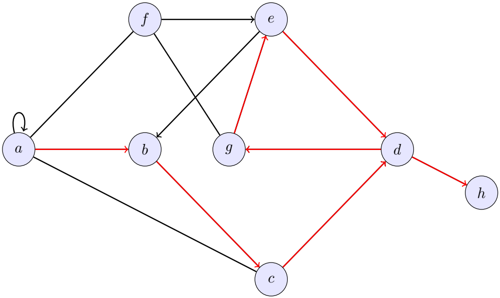
$\gamma = (a, b, c, d, g, e, d, h)$ trail, not path

Example



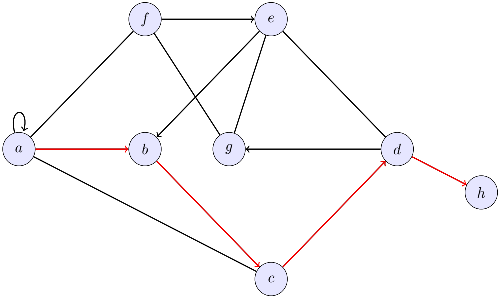
$$\gamma = (a, b, c, d, g, e, d, g, e, d, h)$$

Example



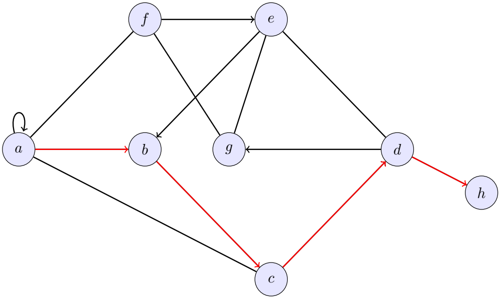
$\gamma = (a, b, c, d, g, e, d, g, e, d, h)$ walk, not trail

Example



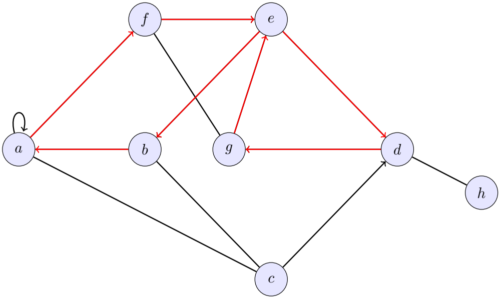
$$\gamma = (a, b, c, d, h)$$

Example



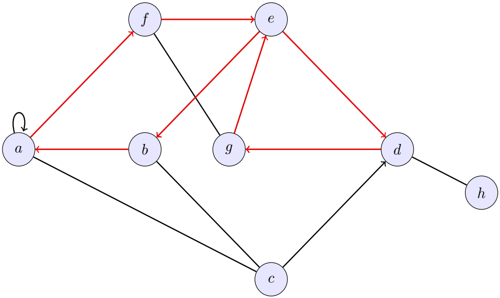
$\gamma = (a, b, c, d, h)$ path

Example



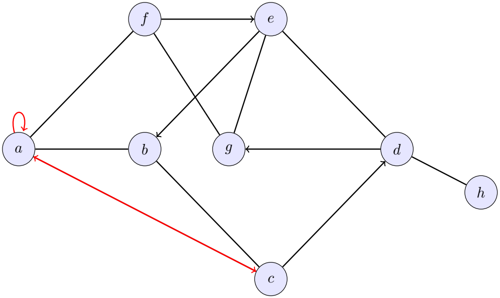
$$\gamma = (d, g, e, b, a, f, e, d)$$

Example



$\gamma = (d, g, e, b, a, f, e, d)$ circuit, not cycle

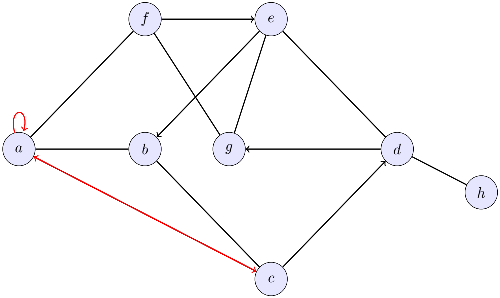
Example



$$\gamma = (a, c, a)$$

$$\gamma' = (a, a)$$

Example

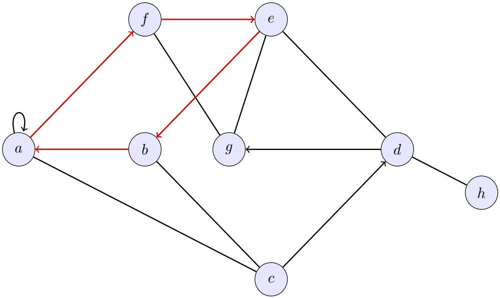


$\gamma = (a, c, a)$ circuit, not cycle

$\gamma' = (a, a)$ circuit, not cycle

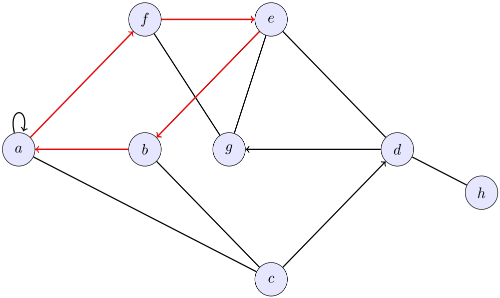
They are called directed cycles: closed paths of any length

Example



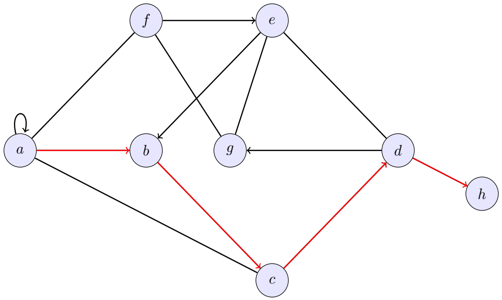
$$\gamma = (e, b, a, f, e)$$

Example



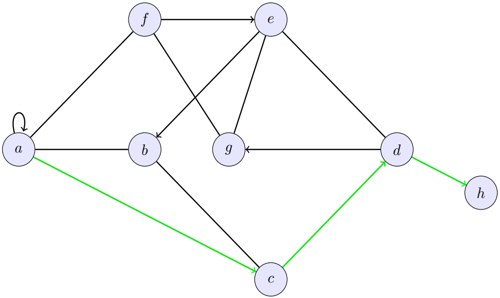
$\gamma = (e, b, a, f, e)$ cycle

Example



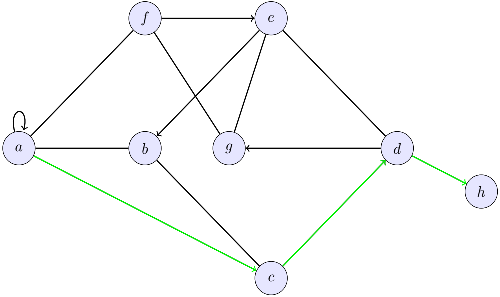
non geodesic path: $\gamma = (a, b, c, d, h)$

Example



geodesic path: $\gamma = (a, c, d, h)$

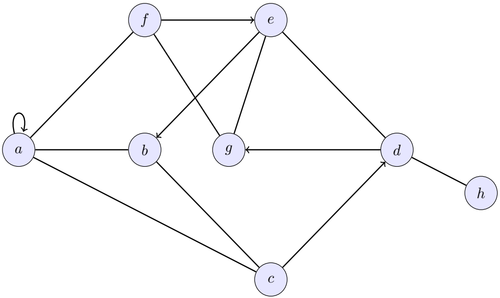
Example



geodesic path: $\gamma = (a, c, d, h)$

$d_{\mathcal{G}}(a, h) = 3$

Example



Is strongly connected?

Subgraphs

$\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, \tilde{W})$, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$

$$\tilde{\mathcal{G}} \subseteq \mathcal{G} \text{ if } \tilde{\mathcal{V}} \subseteq V, \ \tilde{\mathcal{E}} \subseteq \mathcal{E}, \ \tilde{W}_{ij} \leq W_{ij}$$

- *induced subgraph* if $\tilde{\mathcal{E}} = \mathcal{E} \cap (\tilde{\mathcal{V}} \times \tilde{\mathcal{V}}), \ \tilde{W} = W_{|\tilde{V} \times \tilde{V}}$;
- *spanning subgraph* if $\tilde{\mathcal{V}} = \mathcal{V}, \ \tilde{W}_{ij} = \begin{cases} W_{ij} & \text{if } (i,j) \in \tilde{\mathcal{E}} \\ 0 & \text{otherwise} \end{cases}$.

Subgraphs

$\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, \tilde{W})$, $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$

$$\tilde{\mathcal{G}} \subseteq \mathcal{G} \text{ if } \tilde{\mathcal{V}} \subseteq V,\ \tilde{\mathcal{E}} \subseteq \mathcal{E},\ \tilde{W}_{ij} \leq W_{ij}$$

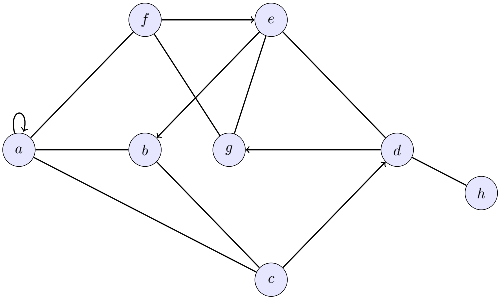
- *induced subgraph* if $\tilde{\mathcal{E}} = \mathcal{E} \cap (\tilde{\mathcal{V}} \times \tilde{\mathcal{V}}),\ \tilde{W} = W_{|\tilde{V} \times \tilde{V}}$;
- *spanning subgraph* if $\tilde{\mathcal{V}} = \mathcal{V},\ \tilde{W}_{ij} = \begin{cases} W_{ij} & \text{if } (i,j) \in \tilde{\mathcal{E}} \\ 0 & \text{otherwise} \end{cases}$.

$\mathcal{P}$ *Monotone* property:

$\tilde{\mathcal{G}}$ spanning subgraph of $\mathcal{G}$, $\mathcal{P}$ true for $\tilde{\mathcal{G}} \Rightarrow \mathcal{P}$ true for $\mathcal{G}$.

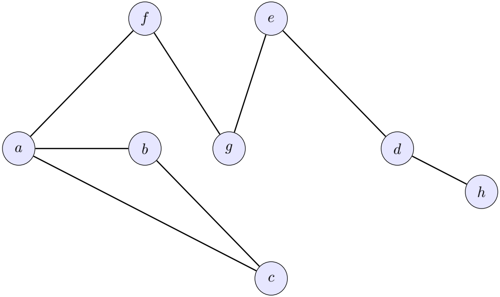
Example: strong connectivity.

Example



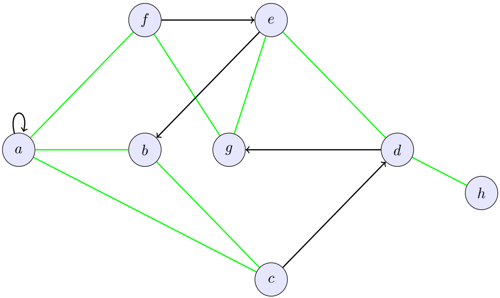
Is strongly connected?

Example



Strongly connected spanning subgraph.

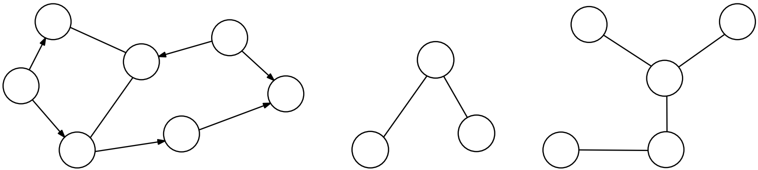
Example



Strongly connected.

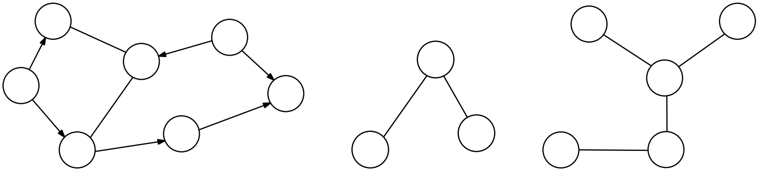
Acyclic versus DAG

Acyclic graphs not DAGs:

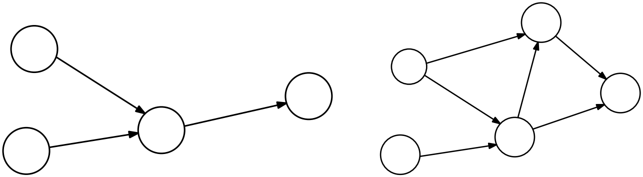


Acyclic versus DAG

Acyclic graphs not DAGs:



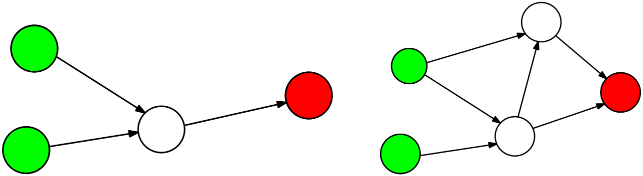
DAGs:



Properties of DAGs

Theorem

- *$\mathcal{G}$ DAG $\Leftrightarrow$ $\mathcal{G}$ acyclic and no 'undirected' edges or self-loops.*
- *In a DAG there are always sources and sinks*



Undirected graphs

Definitions and properties

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ is

- *undirected* if $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$ and W symmetric;
- *simple* if undirected, unweighted $W_{ij} \in \{0, 1\}$, no self-loops.

Definitions and properties

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ is

- *undirected* if $(i, j) \in \mathcal{E} \Leftrightarrow (j, i) \in \mathcal{E}$ and W symmetric;
- *simple* if undirected, unweighted $W_{ij} \in \{0, 1\}$, no self-loops.

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ undirected

- $\mathcal{N}_i = \mathcal{N}_i^-$, $w_i = w_i^-$ (balanced);
- i reachable from $j \Leftrightarrow j$ reachable from i;
- i sink $\Leftrightarrow$ i source $\Leftrightarrow$ i isolated node.

Definitions and properties

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ is

- *undirected* if $(i,j) \in \mathcal{E} \Leftrightarrow (j,i) \in \mathcal{E}$ and W symmetric;
- *simple* if undirected, unweighted $W_{ij} \in \{0,1\}$, no self-loops.

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ undirected

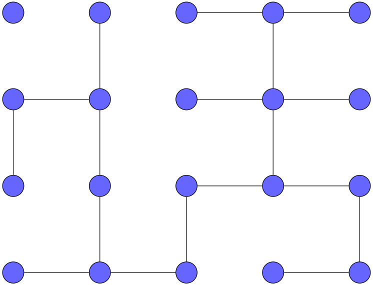
- $\mathcal{N}_i = \mathcal{N}_i^-$, $w_i = w_i^-$ (balanced);
- i reachable from $j \Leftrightarrow j$ reachable from i;
- i sink $\Leftrightarrow$ i source $\Leftrightarrow$ i isolated node.

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ simple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

- $\sum w_i = |\mathcal{E}|$ is even (handshaking lemma);
- i *leaf* if $|\mathcal{N}_i| = 1$

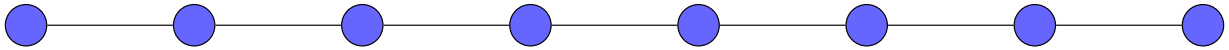
Trees

Trees are acyclic (strongly) connected simple graphs.

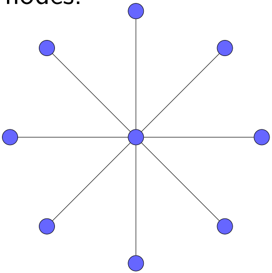


Trees: examples

L_n *Line* with n nodes.



S_n *Star* with $n+1$ nodes.



Combinatorial properties of simple graphs

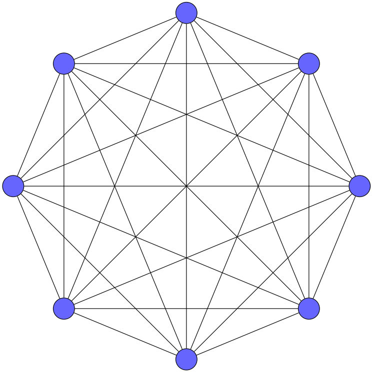
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ simple graph. $n = |\mathcal{V}|$ nodes, $m = |\mathcal{E}|/2$ 'undirected' edges.

Theorem

1. *$\mathcal{G}$ connected $\Rightarrow m \geq n - 1$;*
2. *Assume $\mathcal{G}$ is connected. Then, $\mathcal{G}$ is a tree if and only if $m = n - 1$;*
3. *In a tree there are at least two leaves;*
4. *A tree with exactly two leaves is a line;*

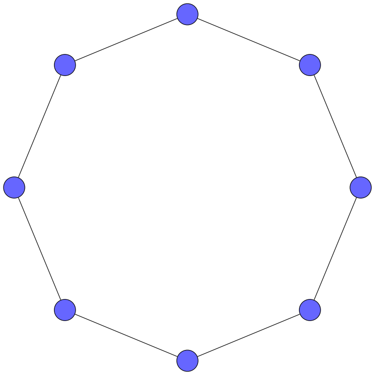
Other examples of simple graphs

K_n *Complete graph*: $m = n(n-1)/2$



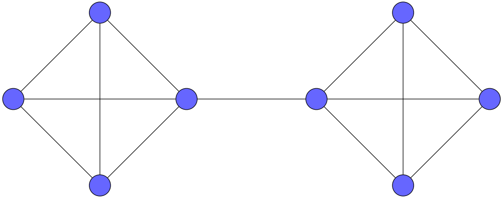
Other examples

R_n *Ring*: $m = n$



Other examples

Barbell: two K_n's connected by an edge



Product of graphs

Let $\mathcal{G}^i = (\mathcal{V}^i, \mathcal{E}^i)$ for $i = 1, 2$ be two simple graphs.

Product graph: $\mathcal{G}^1 \times \mathcal{G}^2 = (\mathcal{V}^1 \times \mathcal{V}^2, \mathcal{E}^1 \otimes \mathcal{E}^2)$

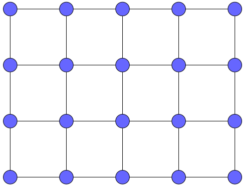
$$((v^1, v^2), (w^1, w^2)) \in \mathcal{E}^1 \otimes \mathcal{E}^2 \;\Leftrightarrow\; \begin{cases} v^1 = w^1, & (v^2, w^2) \in \mathcal{E}^2 \\ v^2 = w^2, & (v^1, w^1) \in \mathcal{E}^1 \end{cases}$$

Theorem

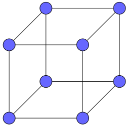
- *If $\mathcal{G}^1$ and $\mathcal{G}^2$ are connected, $\mathcal{G}^1 \times \mathcal{G}^2$ is connected;*
- $\mathrm{diam}(\mathcal{G}^1 \times \mathcal{G}^2) = \mathrm{diam}(\mathcal{G}^1) + \mathrm{diam}(\mathcal{G}^2)$.

Product of graphs: examples

$L_h \times L_k$ *Grid* with $n = h \cdot k$ nodes and $m =$?.

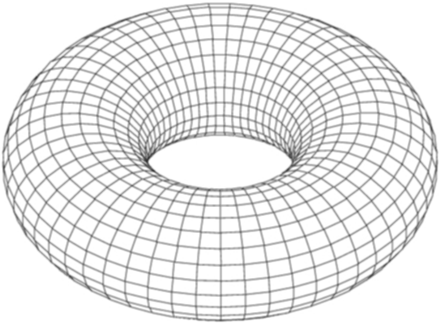


$L_2^k = L_2 \times L_2 \times \cdots L_2$ *Hypercube* with $n = 2^k$ and $m =$?.



Product of graphs: examples

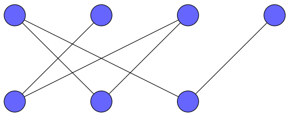
$R_h \times R_k$ *Toroidal grid* with $n = h \cdot k$ nodes and $m = ?$



Bipartite graphs

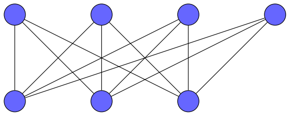
Bipartite graphs:

$$\mathcal{G} = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E}),\ \mathcal{E} \subseteq (\mathcal{V}_1 \times \mathcal{V}_2) \cup (\mathcal{V}_2 \times \mathcal{V}_1)$$



Bipartite graphs: examples

$K_{p,q}$ *Complete bipartite*: $n = p + q$, $m = pq$



Bipartite graphs: properties

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ simple graph. $n = |\mathcal{V}|$ nodes, $m = |\mathcal{E}|/2$ 'undirected' edges.

Theorem

A simple graph is bipartite if and only if it does not have any cycle of odd length.

Examples of bipartite graphs

- trees;
- grids $L_n \times L_m$, hypercubes;
- product of bipartite is bipartite;
- R_n for even n

Advanced connectivity concepts

Periodic and aperiodic graphs

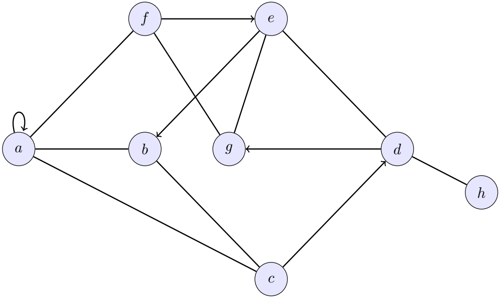
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ connected.

$i \in \mathcal{V} \quad \mathrm{per}(i) := g.c.d\{l \ \exists\text{closed walk of length } l \text{ in } i\}$

Periodic and aperiodic graphs

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ connected.

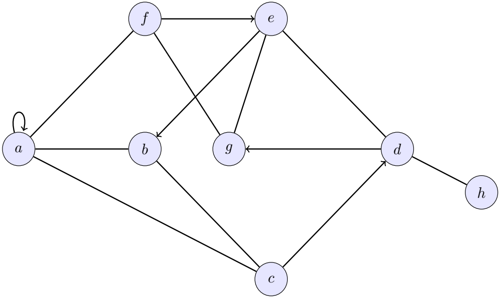
$i \in \mathcal{V} \quad \mathrm{per}(i) := g.c.d\{l \ \exists \text{closed walk of length } l \text{ in } i\}$



Periodic and aperiodic graphs

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ connected.

$i \in \mathcal{V} \quad \mathrm{per}(i) := g.c.d\{l \ \exists \text{closed walk of length } l \text{ in } i\}$

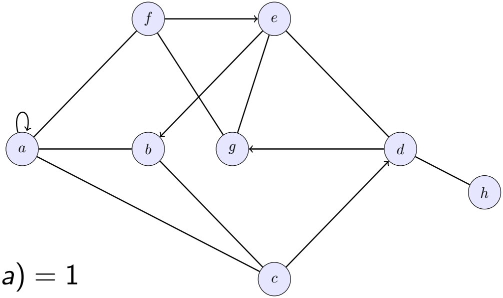


$\mathrm{per}(a) = ?$

Periodic and aperiodic graphs

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ connected.

$i \in \mathcal{V} \quad \mathrm{per}(i) := g.c.d\{l \ \exists \text{closed walk of length } l \text{ in } i\}$

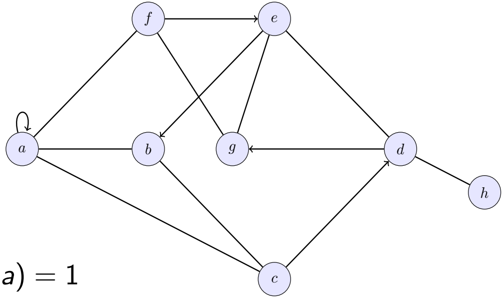


$(a, a) \in \mathcal{E} \Rightarrow \mathrm{per}(a) = 1$

Periodic and aperiodic graphs

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ connected.

$i \in \mathcal{V} \quad \mathrm{per}(i) := g.c.d\{l \; \exists \text{closed walk of length } l \text{ in } i\}$



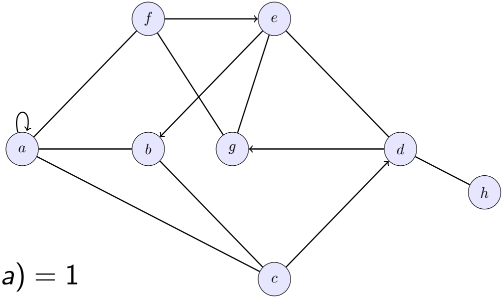
$(a, a) \in \mathcal{E} \Rightarrow \mathrm{per}(a) = 1$

$\mathrm{per}(e) = ?$

Periodic and aperiodic graphs

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ connected.

$i \in \mathcal{V} \quad \mathrm{per}(i) := g.c.d\{l \; \exists \text{closed walk of length } l \text{ in } i\}$



$(a, a) \in \mathcal{E} \Rightarrow \mathrm{per}(a) = 1$

$(e, d, g, e), (e, d, e)$ closed walks $\Rightarrow \mathrm{per}(e) = 1$

Periodic and aperiodic graphs

Theorem

Assume that $\mathcal{G}$ is connected. Then, $\mathrm{per}(i) = \mathrm{per}(j)$ *for every* $i, j \in \mathcal{V}$.

Periodic and aperiodic graphs

Theorem

Assume that $\mathcal{G}$ is connected. Then, $\mathrm{per}(i) = \mathrm{per}(j)$ *for every* $i, j \in \mathcal{V}$.

$\mathrm{per}_{\mathcal{G}}$ *period* of $\mathcal{G}$. $\mathcal{G}$ is called *aperiodic* if $\mathrm{per}_{\mathcal{G}} = 1$

Periodic and aperiodic graphs

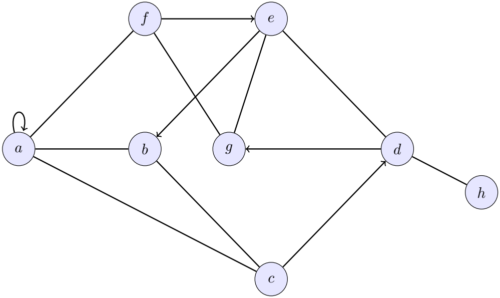
Theorem

Assume that $\mathcal{G}$ is connected. Then, $\mathrm{per}(i) = \mathrm{per}(j)$ for every $i, j \in \mathcal{V}$.

$\mathrm{per}_{\mathcal{G}}$ *period* of $\mathcal{G}$. $\mathcal{G}$ is called *aperiodic* if $\mathrm{per}_{\mathcal{G}} = 1$

- $\exists$ self loop $\Rightarrow$ aperiodic
- $\mathcal{G}$ undirected $\Rightarrow$ $\mathrm{per}_{\mathcal{G}} = 1, 2$

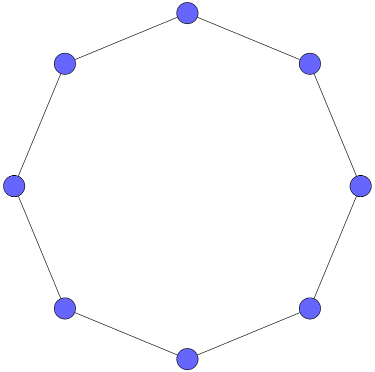
Periodic and aperiodic graphs



Aperiodic

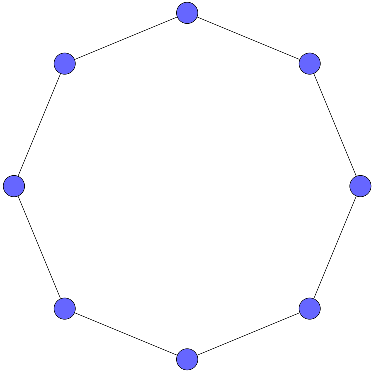
Periodic and aperiodic graphs

R_n Ring



Periodic and aperiodic graphs

R_n Ring



$$\mathrm{per}_{R_n} = g.c.d.\{2, n\} = \begin{cases} 1 & \text{if } n \text{ is odd} \\ 2 & \text{if } n \text{ is even} \end{cases}$$

Connected components

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$

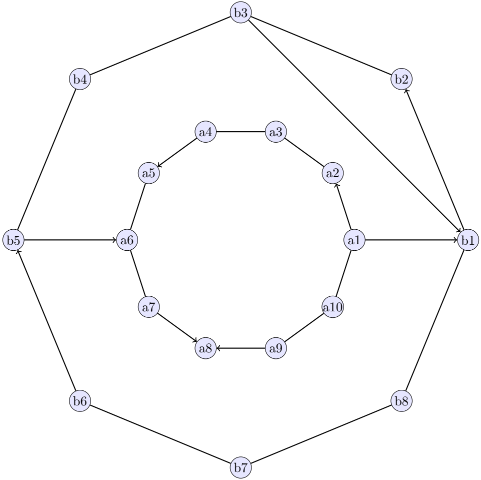
Equivalence relation: $v \sim w$ if v is reachable from w and w is reachable from v.

Equivalence classes are called *connected components* $C \in \mathcal{C}$

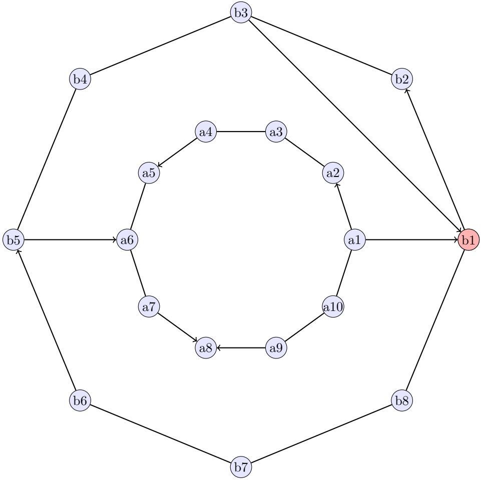
Condensation graph $\mathcal{H} = (\mathcal{C}, \mathcal{F})$: $(C_1, C_2) \in \mathcal{F}$ if there is an edge from some node in C_1 to some node in C_2.

- $\mathcal{H}$ is a DAG;
- $\mathcal{G}$ undirected $\Rightarrow \mathcal{H}$ is composed of isolated nodes.

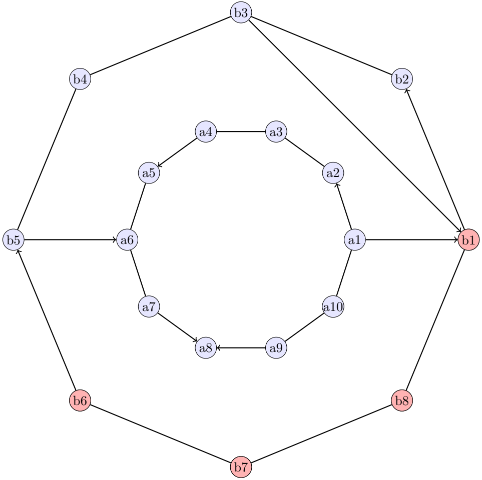
Example



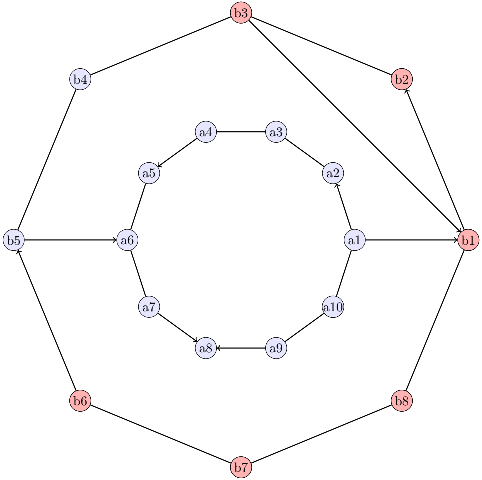
Example



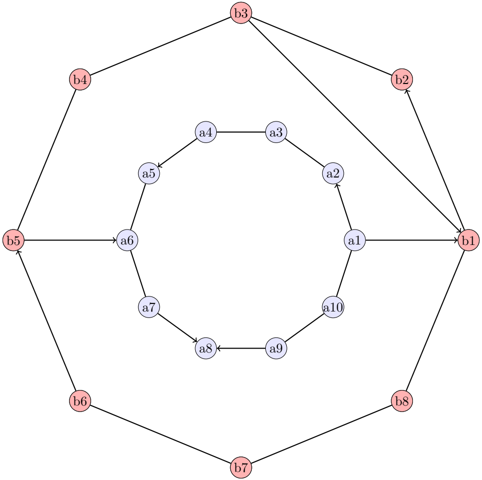
Example



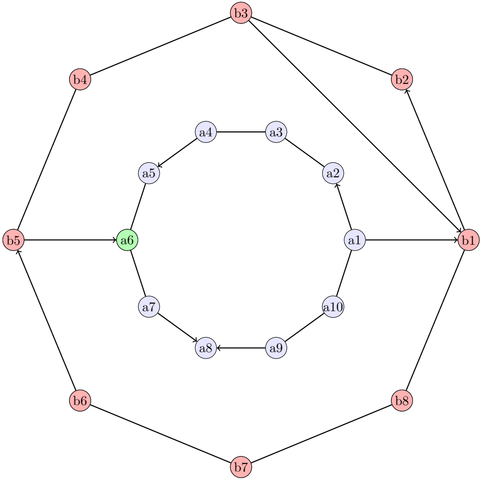
Example



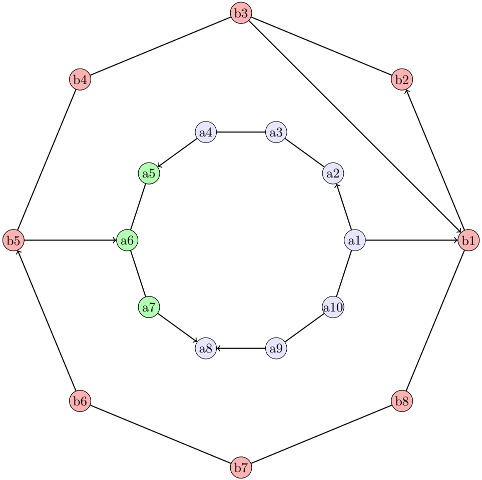
Example



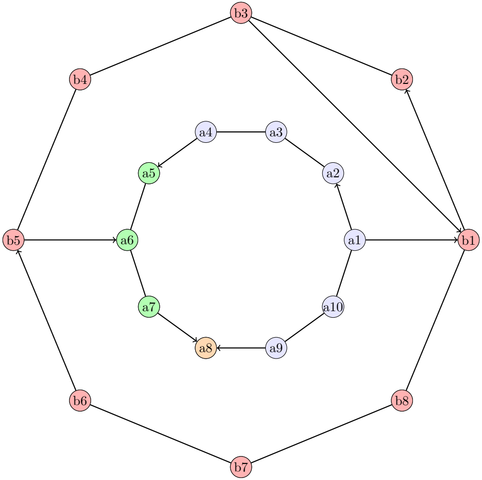
Example



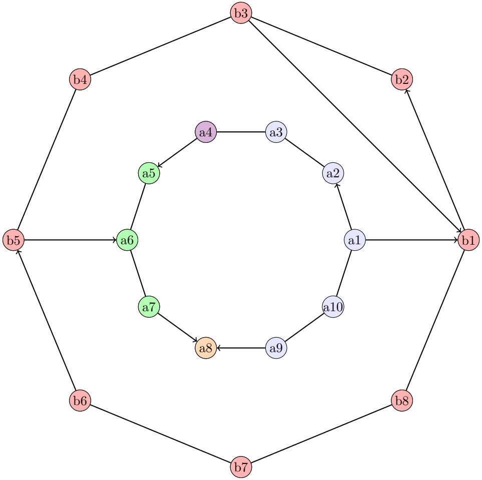
Example



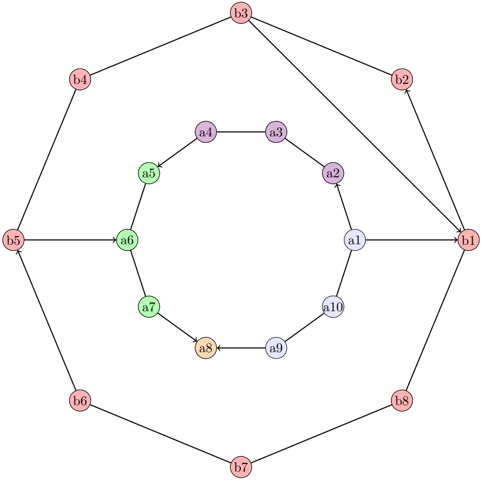
Example



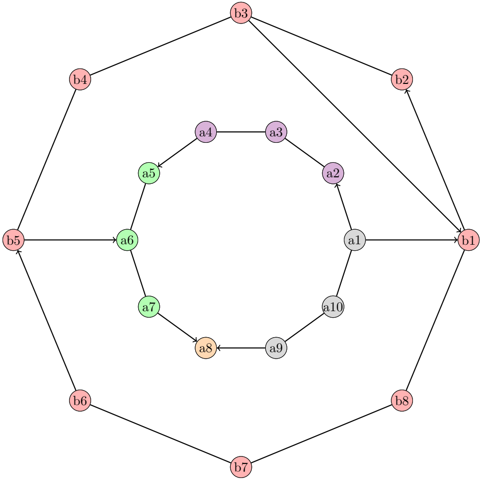
Example



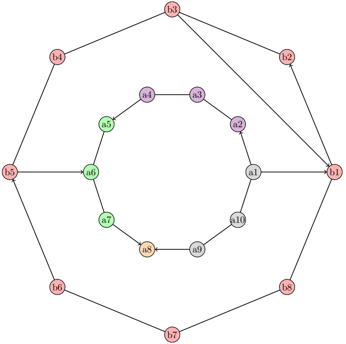
Example

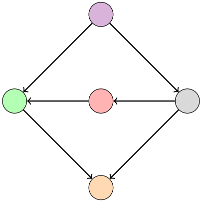


Example



Example





Other connectivity issues

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W), \mathcal{U} \subseteq \mathcal{V}$

- $\mathcal{U}$ is *trapping* if $i \in \mathcal{U},\ (i, j) \in \mathcal{E} \ \Rightarrow j \in \mathcal{U}$;
- $\mathcal{U}$ is *globally reachable* if for every $i \in \mathcal{V}$ there is a walk from i to some $j \in \mathcal{U}$.

Other connectivity issues

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, $\mathcal{U} \subseteq \mathcal{V}$

- $\mathcal{U}$ is *trapping* if $i \in \mathcal{U},\ (i,j) \in \mathcal{E}\ \Rightarrow j \in \mathcal{U}$;
- $\mathcal{U}$ is *globally reachable* if for every $i \in \mathcal{V}$ there is a walk from i to some $j \in \mathcal{U}$.

If $\mathcal{G}$ is connected,

- each subset $\mathcal{U} \neq \emptyset$ is globally reachable;
- the only subset $\mathcal{U} \neq \emptyset$ that is trapping is $\mathcal{U} = \mathcal{V}$.

Other connectivity issues

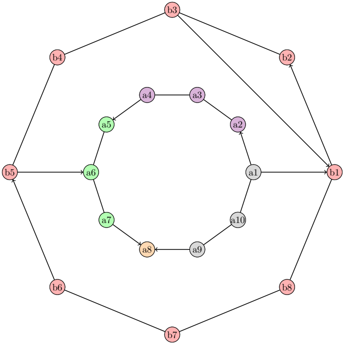
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, $\mathcal{U} \subseteq \mathcal{V}$

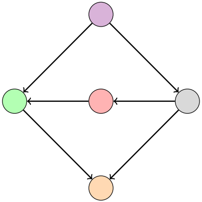
- $\mathcal{U}$ is *trapping* if $i \in \mathcal{U},\ (i,j) \in \mathcal{E}\ \Rightarrow j \in \mathcal{U}$;
- $\mathcal{U}$ is *globally reachable* if for every $i \in \mathcal{V}$ there is a walk from i to some $j \in \mathcal{U}$.

If $\mathcal{G}$ is any graph,

- sink components in the condensation graph are trapping;
- from every node, there is a walk to a node belonging to a sink component;
- if there is just one sink component C in the condensation graph, then C is trapping and globally reachable.

Example





$\{a8\}$ is trapping and globally reachable

Some applications in AI

Patrolling with drones

- $\mathcal{V}$ set of locations to be patrolled at certain times.
- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ simple graph describing constraints: $(i, j) \in \mathcal{E}$ means that i and j can not be patrolled by the same drone (e.g. temporal constraints).

Patrolling with drones

- $\mathcal{V}$ set of locations to be patrolled at certain times.
- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ simple graph describing constraints: $(i, j) \in \mathcal{E}$ means that i and j can not be patrolled by the same drone (e.g. temporal constraints).

Problems:

- Determine the minimum number k of drones needed to patrol all locations.
- Find a function $\psi : \mathcal{V} \to \Omega$ where $\Omega = \{1, 2, \dots, k\}$ so that

$$(i, j) \in \mathcal{E} \ \Rightarrow \ \psi(i) \neq \psi(j)$$

Interpretation: $\psi(i) \in \Omega$ indicates the drone that will patrol location i.

The coloring problem

The problem just explained is an instance of the so-called coloring problem.

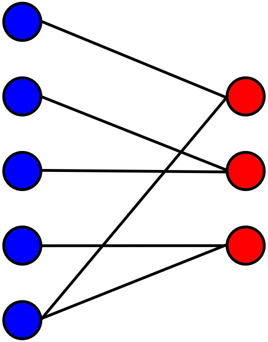
The function $\psi : \mathcal{V} \to \Omega$ is called a *coloring* with k colors: the requirement is that neighbor nodes must have different colors.

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$ is said to be *k-colorable* if there exists a coloring with k colors.

The minimum number of colors k for which $\mathcal{G}$ is k-colorable, is called the *chromatic number* of $\mathcal{G}$.

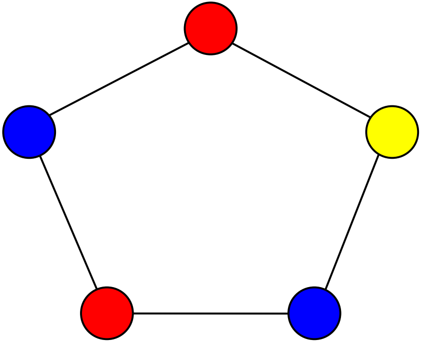
The coloring problem: examples

Bipartite graphs (trees, even length cycles) have chromatic number 2.



The coloring problem: examples

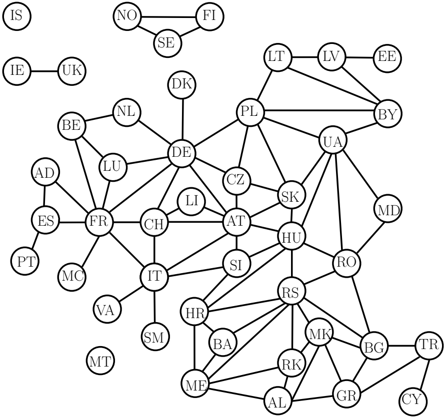
An odd length cycle has chromatic number 3.



The coloring problem: examples

Planar graphs have chromatic number at most 4 (the famous 4-colors problem).

The coloring problem: examples



The coloring problem: problems

Compute the chromatic number of the following graphs:

- the complete graph K_n
- the barbell graph
- the $L_h \times L_k$ grid
- the hypercube

A different patrolling problem

- $\mathcal{V}$ set of locations to be patrolled.
- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ simple graph describing constraints: $(i, j) \in \mathcal{E}$ means that i and j can not be patrolled by the same drone.
- There is only one drone available.

A different patrolling problem

- $\mathcal{V}$ set of locations to be patrolled.
- $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ simple graph describing constraints: $(i, j) \in \mathcal{E}$ means that i and j can not be patrolled by the same drone.
- There is only one drone available.

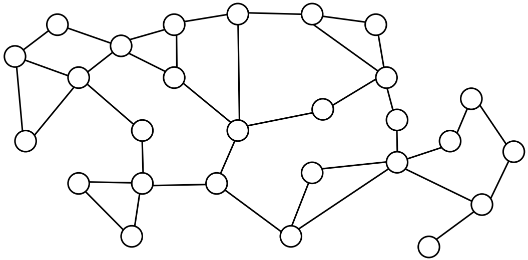
Problems:

- Determine the maximum number of locations k that can be patrolled by a single drone.
- Find all subsets $\mathcal{W} \subseteq \mathcal{V}$ consisting of k locations that can be patrolled by a single machine.

Independent sets

This leads to another classical graph theoretic problem. $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ undirected graph.

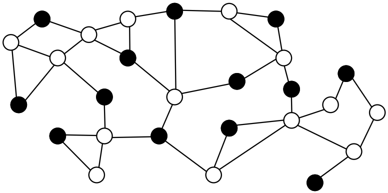
- $\mathcal{W} \subseteq \mathcal{V}$ is called an *independent* set if, given any $i, j \in \mathcal{W}$, it holds that $(i, j) \notin \mathcal{E}$.
- We are looking for independent sets of $\mathcal{V}$ of *maximum cardinality*.
- It is easy to find *maximal independent sets* (they can not be enlarged)

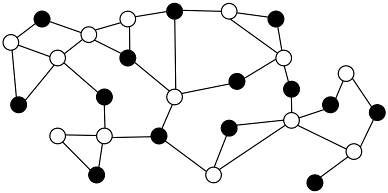


Independent sets

This leads to a classical graph theoretic problem. $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ simple graph.

- $\mathcal{W} \subseteq \mathcal{V}$ is called an *independent* set if, given any $i, j \in \mathcal{W}$, it holds that $(i, j) \notin \mathcal{E}$.
- We are looking for independent sets of $\mathcal{V}$ of *maximum cardinality*.
- It is easy to find *maximal independent sets* (they can not be enlarged)

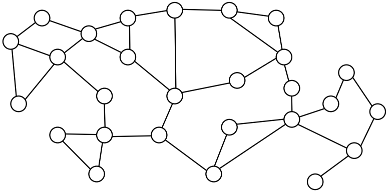




Independent edges, the matching problem

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ undirected graph.

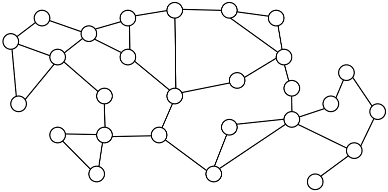
$\mathcal{F} \subseteq \mathcal{E}$ is called an *independent edge set* or a *matching* if any pair of edges in $\mathcal{F}$ does not have nodes in common

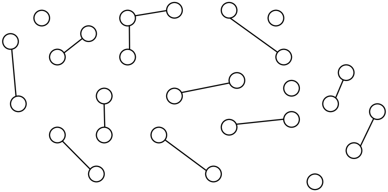


Independent edges, the matching problem

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ undirected graph.

$\mathcal{F} \subseteq \mathcal{E}$ is called an *independent edge set* or a *matching* if any pair of edges in $\mathcal{F}$ does not have nodes in common





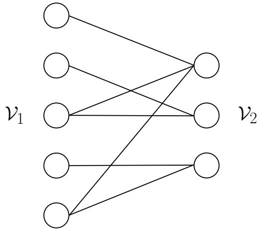
Maximal matching

Matching in bipartite graphs

$\mathcal{G} = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E})$, bipartite graph w.r. to the partition $\mathcal{V}_1 \cup \mathcal{V}_2$

$\mathcal{F} \subseteq \mathcal{E}$ matching $\Rightarrow |\mathcal{F}| \leq \min\{|\mathcal{V}_1|, |\mathcal{V}_2|\}$

Matching *complete* on $\mathcal{V}_1$ if $|\mathcal{F}| = |\mathcal{V}_1|$ (and thus $|\mathcal{V}_2| \geq |\mathcal{V}_1|$)

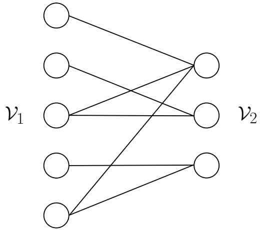


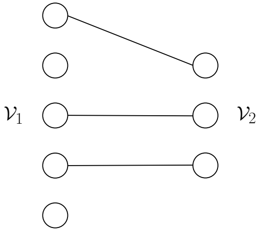
Matching in bipartite graphs

$\mathcal{G} = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E})$, bipartite graph w.r. to the partition $\mathcal{V}_1 \cup \mathcal{V}_2$

$\mathcal{F} \subseteq \mathcal{E}$ matching $\Rightarrow |\mathcal{F}| \leq \min\{|\mathcal{V}_1|, |\mathcal{V}_2|\}$

Matching *complete* on $\mathcal{V}_1$ if $|\mathcal{F}| = |\mathcal{V}_1|$ (and thus $|\mathcal{V}_2| \geq |\mathcal{V}_1|$)

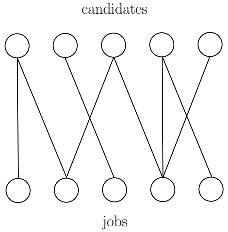




Matching complete on $\mathcal{V}_2$

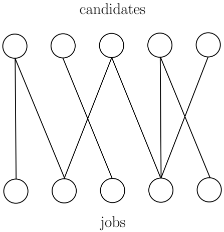
Application

A group of candidates and a set of jobs, an edge indicates that the candidate is qualified for that job.



Application

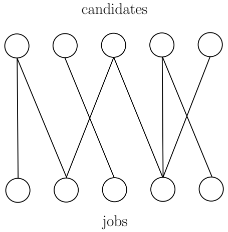
A group of candidates and a set of jobs, an edge indicates that the candidate is qualified for that job.



Is there a way to assign each candidate to a single job they are qualified such that every job has only one person assigned to it?

Application

A group of candidates and a set of jobs, an edge indicates that the candidate is qualified for that job.

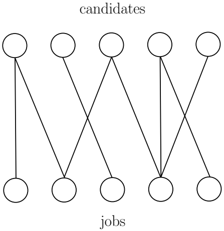


Is there a way to assign each candidate to a single job they are qualified such that every job has only one person assigned to it?

Equivalently, is there a matching complete on candidates?

Application

A group of candidates and a set of jobs, an edge indicates that the candidate is qualified for that job.



Complete matching on candidates and on jobs: *perfect matching*

Matching in bipartite graphs

Theorem (Hall's marriage 1934)

$\mathcal{G} = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E})$, bipartite graph w.r. to the partition $\mathcal{V}_1 \cup \mathcal{V}_2$. The following conditions are equivalent:

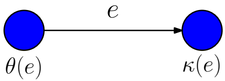
1. *There exists a complete matching on $\mathcal{V}_1$;*
2. *For every subset $\mathcal{W} \subseteq \mathcal{V}_1$, we have that $|N_{\mathcal{W}}| \geq |\mathcal{W}|$, where*

$$N_{\mathcal{W}} = \{v \in \mathcal{V}_2 \,|\, (v, w) \in \mathcal{E} \text{ for some } w \in \mathcal{W}\}$$

We will prove it later as a corollary to the max-flow min-cut theorem. Direct proofs exist.

Multigraphs

- Node set $\mathcal{V}$
- Link set $\mathcal{E}$
- $\theta : \mathcal{E} \to \mathcal{V}$, $\kappa : \mathcal{E} \to \mathcal{V}$: $\theta(e)$ *tail* of e, $\kappa(e)$ *head* of e

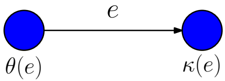


- $h \in \mathbb{R}_+^{\mathcal{E}}$ weight on edges

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, h)$ *multigraph*

Multigraphs

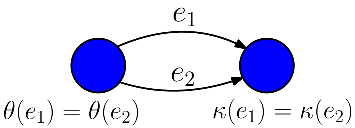
- Node set $\mathcal{V}$
- Link set $\mathcal{E}$
- $\theta : \mathcal{E} \to \mathcal{V}$, $\kappa : \mathcal{E} \to \mathcal{V}$: $\theta(e)$ *tail* of e, $\kappa(e)$ *head* of e



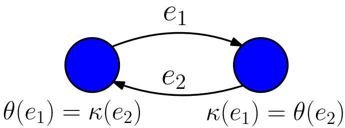
- $h \in \mathbb{R}_+^{\mathcal{E}}$ weight on edges

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, h)$ *multigraph*

Parallel edges:



Opposite edges:



Multigraphs

$\mathcal{G} = (\mathcal{V}, \mathcal{E}, h)$ *multigraph*

Most of graph theoretic concepts naturally generalize to the new setting.

- A walk in $\mathcal{G}$ is a sequence of edges $\gamma = (e_1, e_2, \ldots, e_r)$ such that $\kappa(e_k) = \theta(e_{k+1})$ for $k = 1, \ldots r - 1$.
- $w_k = \sum_{e\,:\,\theta(e)=k} h_e$ degree of node k
- $\mathcal{G}$ is undirected if $\mathcal{E}$ consists of pairs of opposite edges $\{e, \bar{e}\}$ and $h_e = h_{\bar{e}}$.
- An undirected graph $\mathcal{G}$ is called *Eulerian* if there exists a closed walk using all edges just once (identifying opposite pairs).

Eulerian graphs

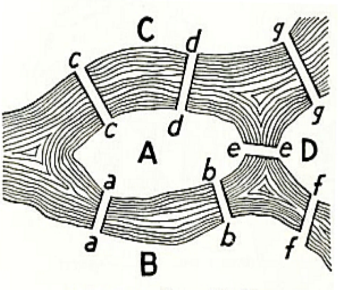
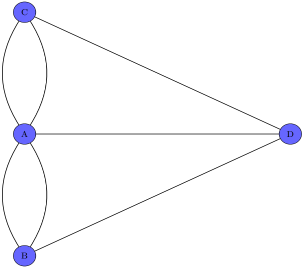


FIGURE 98. *Geographic Map: The Königsberg Bridges.*



Is it Eulerian?

Eulerian graphs

Theorem

An unweighted undirected multigraph $\mathcal{G}$ is Eulerian if and only if it is connected and every node has even degree.