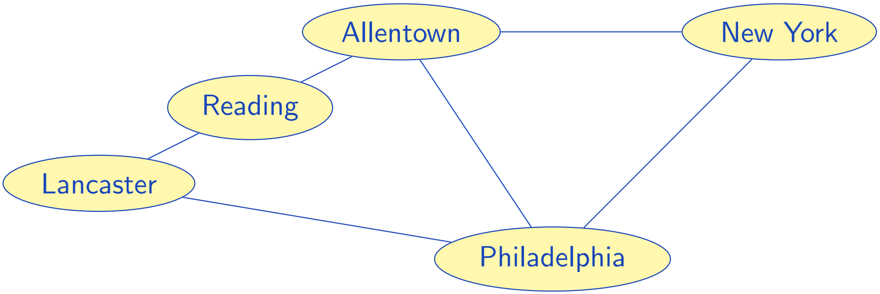
ITMO UNIVERSITY

How to Win Coding Competitions: Secrets of Champions

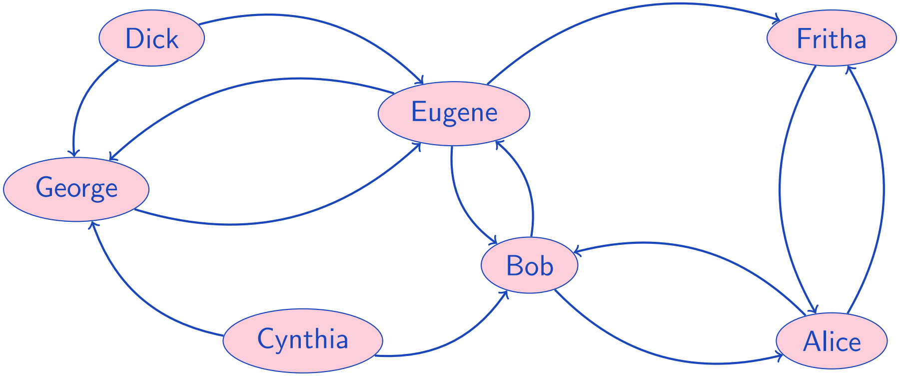
Week 5: Algorithms on Graphs 1
Lecture 1: Introduction to graphs

Maxim Buzdalov
Saint Petersburg 2016

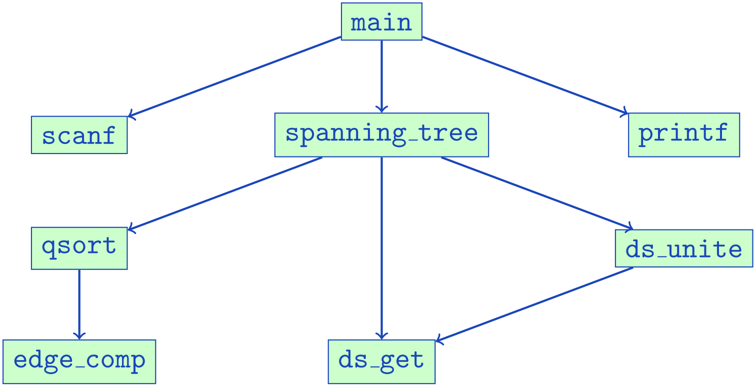
Roads and cities



Social networks



Even computer programs

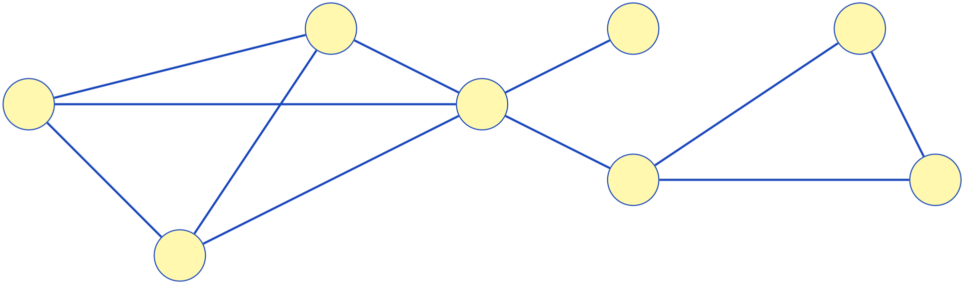


Undirected graph: an ordered pair $G = \langle V, E \rangle$

- V – set of graph's vertices
- E – set of graph's edges, a multiset of **unordered** pairs of vertices
- $(u, v) \in E$ means that an edge connecting vertices u and v exists in the graph

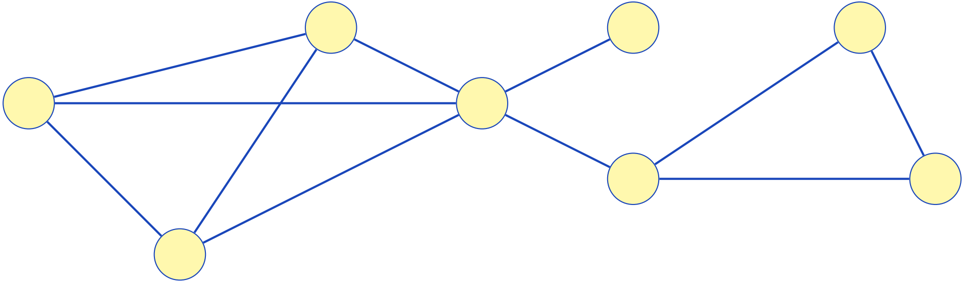
Undirected graph: an ordered pair $G = \langle V, E \rangle$

- V – set of graph's vertices
- E – set of graph's edges, a multiset of **unordered** pairs of vertices
- $(u, v) \in E$ means that an edge connecting vertices u and v exists in the graph



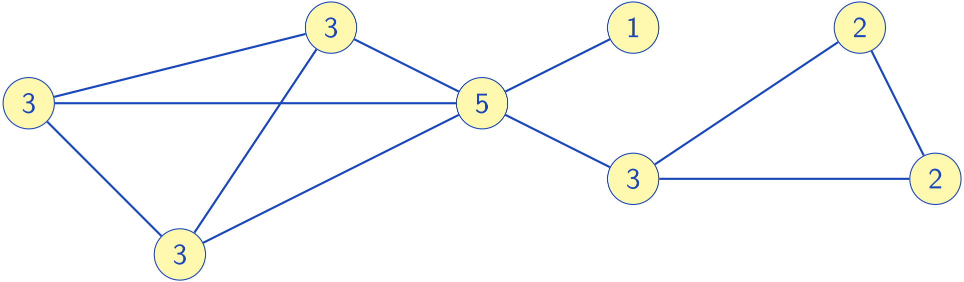
Undirected graph: an ordered pair $G = \langle V, E \rangle$

- V – set of graph's vertices
- E – set of graph's edges, a multiset of **unordered** pairs of vertices
- $(u, v) \in E$ means that an edge connecting vertices u and v exists in the graph
- Degree of a vertex v – $deg(v)$ – the number of edges in E which contain v



Undirected graph: an ordered pair $G = \langle V, E \rangle$

- V – set of graph's vertices
- E – set of graph's edges, a multiset of **unordered** pairs of vertices
- $(u, v) \in E$ means that an edge connecting vertices u and v exists in the graph
- Degree of a vertex v – $deg(v)$ – the number of edges in E which contain v

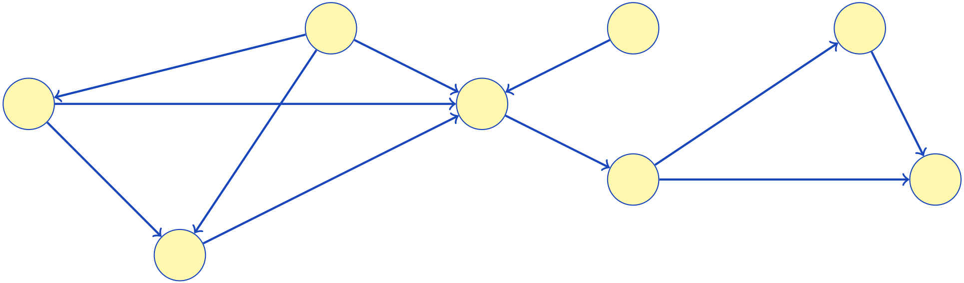


Directed graph: an ordered pair $G = \langle V, E \rangle$

- V – set of graph's vertices
- E – set of graph's edges, a multiset of **ordered** pairs of vertices

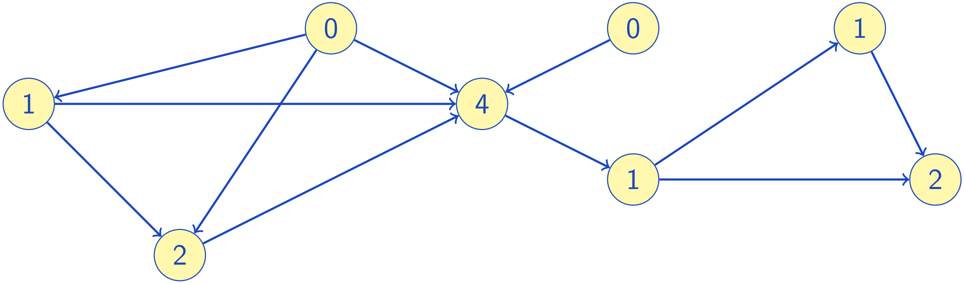
Directed graph: an ordered pair $G = \langle V, E \rangle$

- V – set of graph's vertices
- E – set of graph's edges, a multiset of **ordered** pairs of vertices



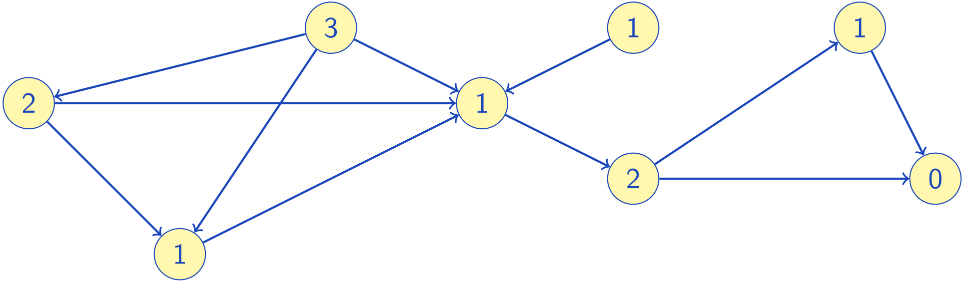
Directed graph: an ordered pair $G = \langle V, E \rangle$

- V – set of graph's vertices
- E – set of graph's edges, a multiset of **ordered** pairs of vertices
- Incoming degree of a vertex v – $deg^{-}(v)$ – the number of edges $(x, v) \in E$



Directed graph: an ordered pair $G = \langle V, E \rangle$

- V – set of graph's vertices
- E – set of graph's edges, a multiset of **ordered** pairs of vertices
- Incoming degree of a vertex v – $deg^-(v)$ – the number of edges $(x, v) \in E$
- Outgoing degree of a vertex v – $deg^+(v)$ – the number of edges $(v, x) \in E$

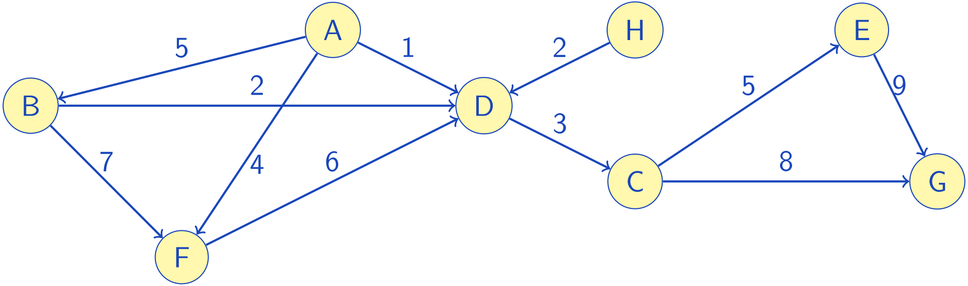


Weighted graph

- Can be either directed or undirected graph
- Has a function $W : E \to X$, where X is the set of weights
- X is typically integers or reals, can also be symbols or strings

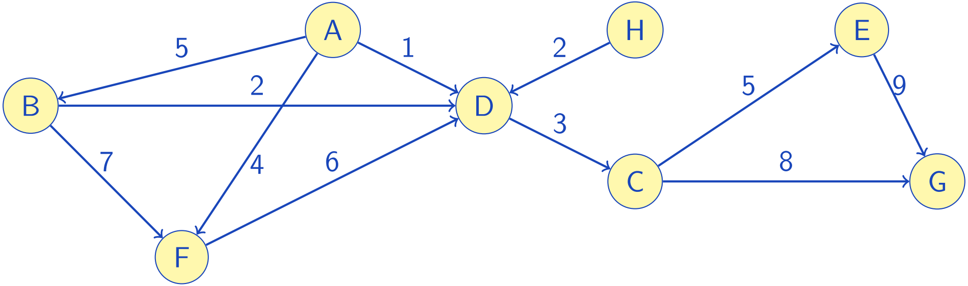
Weighted graph

- Can be either directed or undirected graph
- Has a function $W : E \to X$, where X is the set of weights
- X is typically integers or reals, can also be symbols or strings



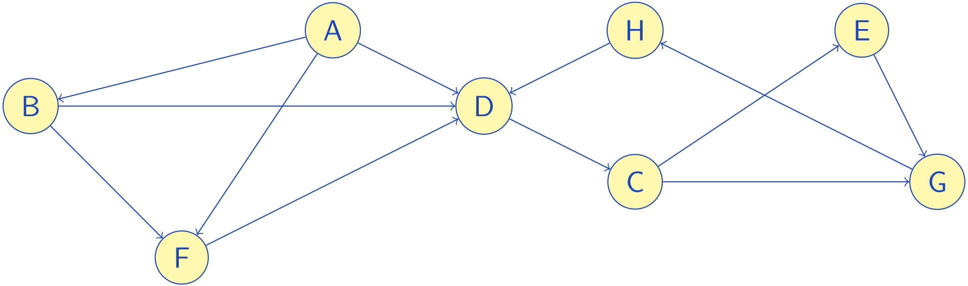
Weighted graph

- Can be either directed or undirected graph
- Has a function $W : E \to X$, where X is the set of weights
- X is typically integers or reals, can also be symbols or strings
- In some cases, an unweighted graph is the same as a weighted graph with $X = \{1\}$



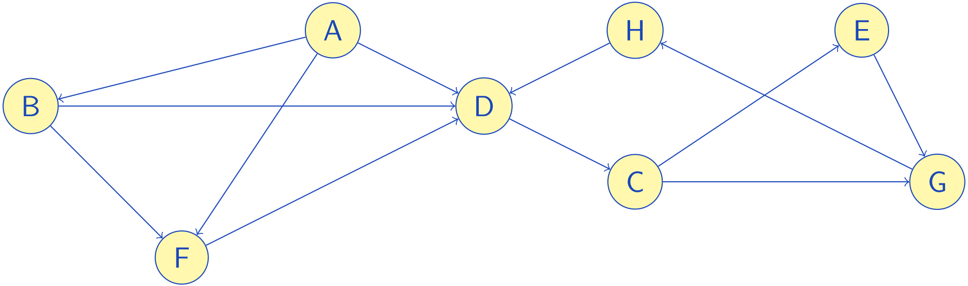
Path in the graph $G = \langle V, E \rangle$

- A list of vertices $v_1, v_2, \ldots, v_k$, such that $(v_i, v_{i+1}) \in E$ for all $1 \leq i < k$



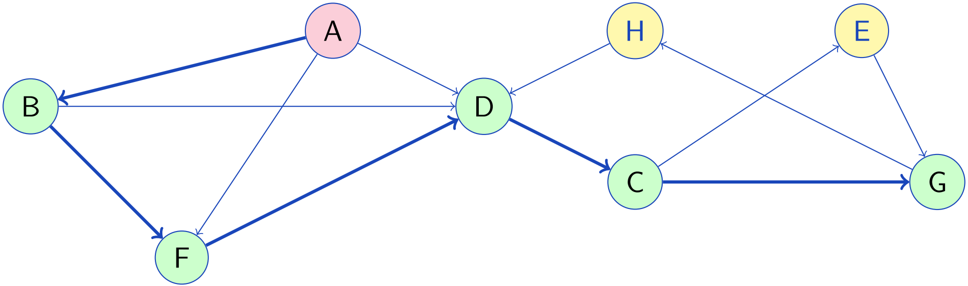
Path in the graph $G = \langle V, E \rangle$

- A list of vertices $v_1, v_2, \ldots, v_k$, such that $(v_i, v_{i+1}) \in E$ for all $1 \leq i < k$
- Example: $[A, B, F, D, C, G]$ is a path



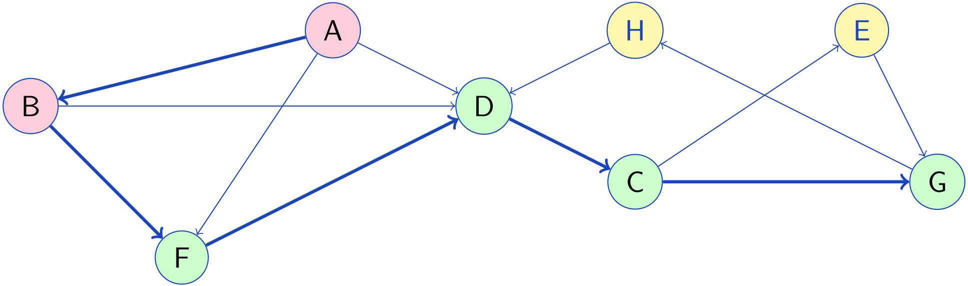
Path in the graph $G = \langle V, E \rangle$

- A list of vertices $v_1, v_2, \ldots, v_k$, such that $(v_i, v_{i+1}) \in E$ for all $1 \leq i < k$
- Example: $[A, B, F, D, C, G]$ is a path



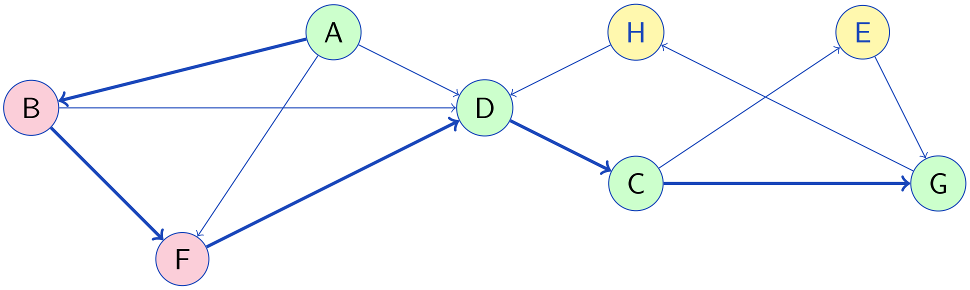
Path in the graph $G = \langle V, E \rangle$

- A list of vertices $v_1, v_2, \ldots, v_k$, such that $(v_i, v_{i+1}) \in E$ for all $1 \le i < k$
- Example: $[A, B, F, D, C, G]$ is a path



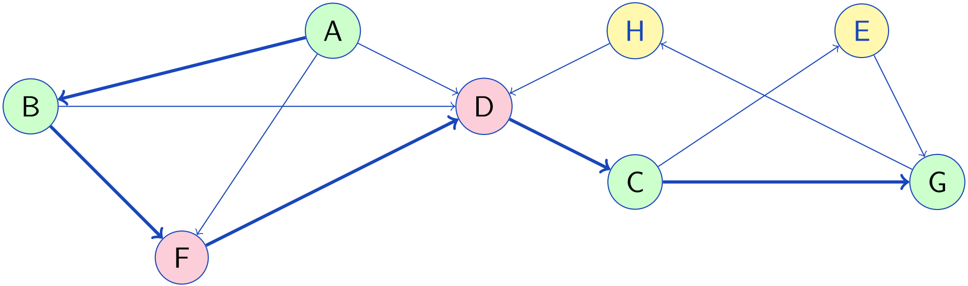
Path in the graph $G = \langle V, E \rangle$

- A list of vertices $v_1, v_2, \ldots, v_k$, such that $(v_i, v_{i+1}) \in E$ for all $1 \leq i < k$
- Example: $[A, B, F, D, C, G]$ is a path



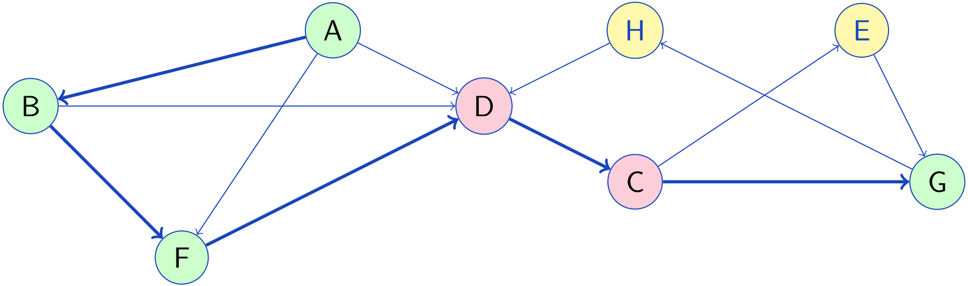
Path in the graph $G = \langle V, E \rangle$

- A list of vertices $v_1, v_2, \ldots, v_k$, such that $(v_i, v_{i+1}) \in E$ for all $1 \leq i < k$
- Example: $[A, B, F, D, C, G]$ is a path



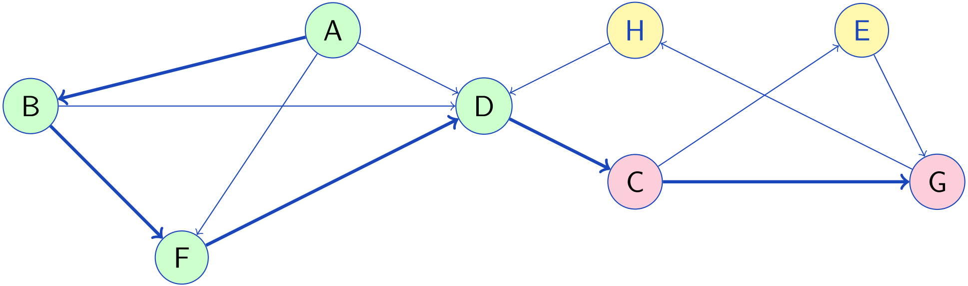
Path in the graph $G = \langle V, E \rangle$

- A list of vertices $v_1, v_2, \ldots, v_k$, such that $(v_i, v_{i+1}) \in E$ for all $1 \leq i < k$
- Example: $[A, B, F, D, C, G]$ is a path



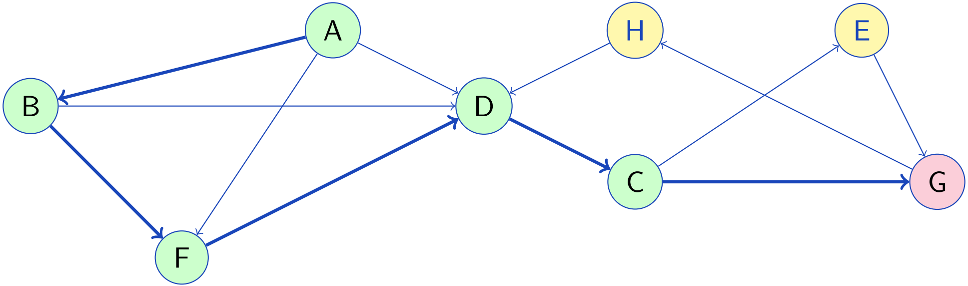
Path in the graph $G = \langle V, E \rangle$

- A list of vertices $v_1, v_2, \ldots, v_k$, such that $(v_i, v_{i+1}) \in E$ for all $1 \leq i < k$
- Example: $[A, B, F, D, C, G]$ is a path



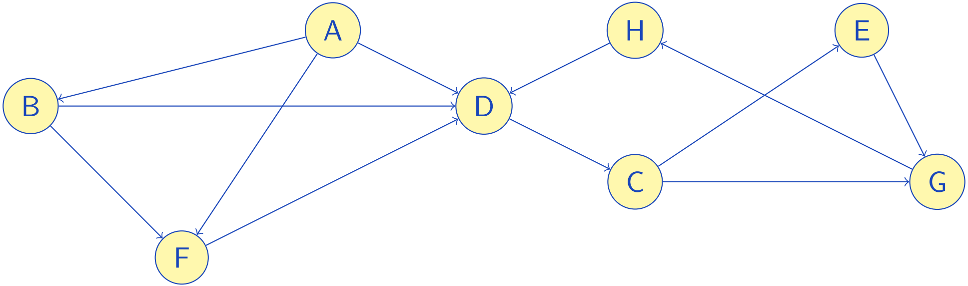
Path in the graph $G = \langle V, E \rangle$

- A list of vertices $v_1, v_2, \ldots, v_k$, such that $(v_i, v_{i+1}) \in E$ for all $1 \leq i < k$
- Example: $[A, B, F, D, C, G]$ is a path



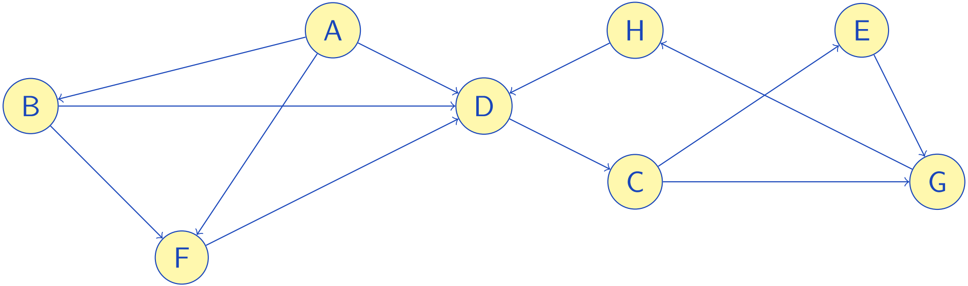
Path in the graph $G = \langle V, E \rangle$

- A list of vertices $v_1, v_2, \ldots, v_k$, such that $(v_i, v_{i+1}) \in E$ for all $1 \leq i < k$
- Example: $[A, B, F, D, C, G]$ is a path
- Simple path: no vertex appears twice in the path



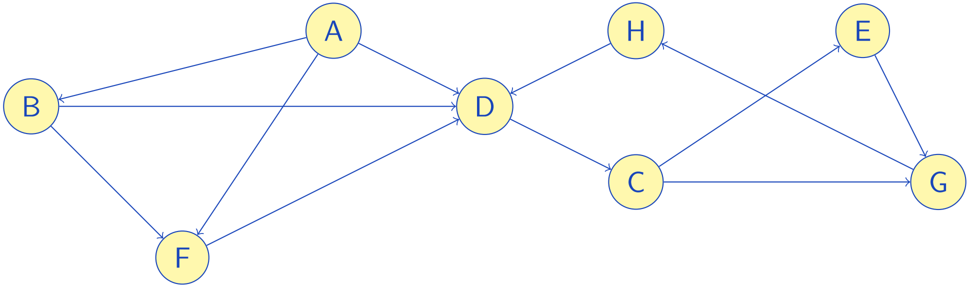
Cycle in the graph $G = \langle V, E \rangle$

- A path $v_1, v_2, \ldots, v_k$, such that $v_k = v_1$



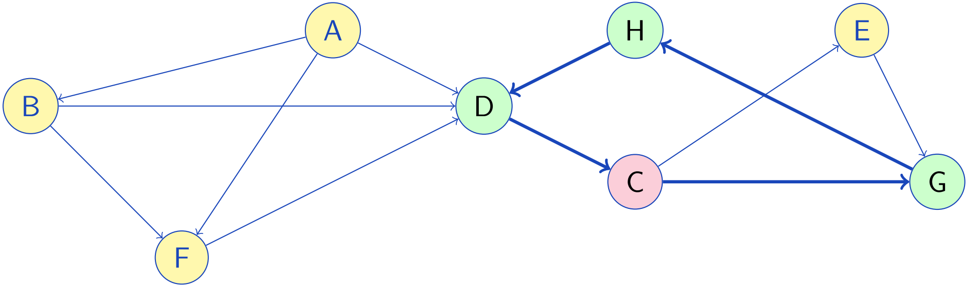
Cycle in the graph $G = \langle V, E \rangle$

- A path $v_1, v_2, \dots, v_k$, such that $v_k = v_1$
- Example: $[C, G, H, D]$ is a cycle



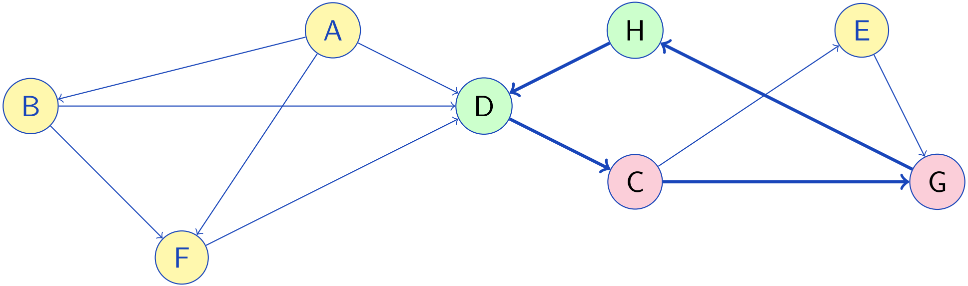
Cycle in the graph $G = \langle V, E \rangle$

- A path $v_1, v_2, \ldots, v_k$, such that $v_k = v_1$
- Example: $[C, G, H, D]$ is a cycle



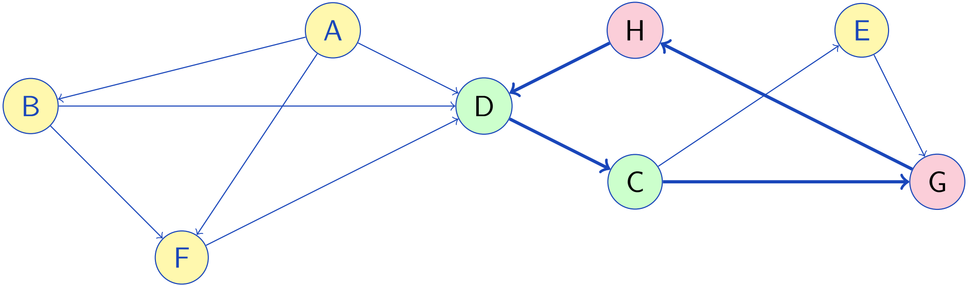
Cycle in the graph $G = \langle V, E \rangle$

- A path $v_1, v_2, \ldots, v_k$, such that $v_k = v_1$
- Example: $[C, G, H, D]$ is a cycle



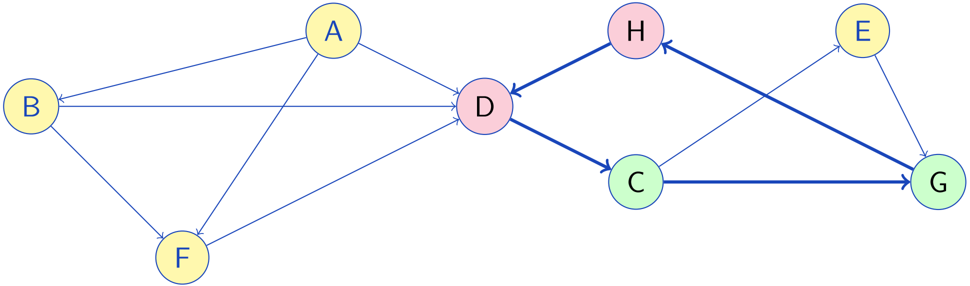
Cycle in the graph $G = \langle V, E \rangle$

- A path $v_1, v_2, \ldots, v_k$, such that $v_k = v_1$
- Example: $[C, G, H, D]$ is a cycle



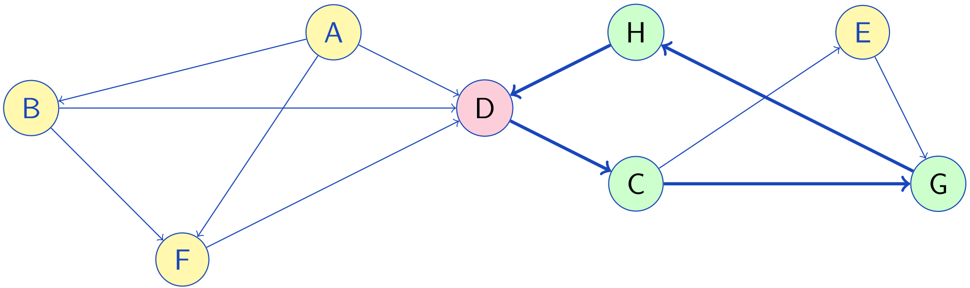
Cycle in the graph $G = \langle V, E \rangle$

- A path $v_1, v_2, \ldots, v_k$, such that $v_k = v_1$
- Example: $[C, G, H, D]$ is a cycle



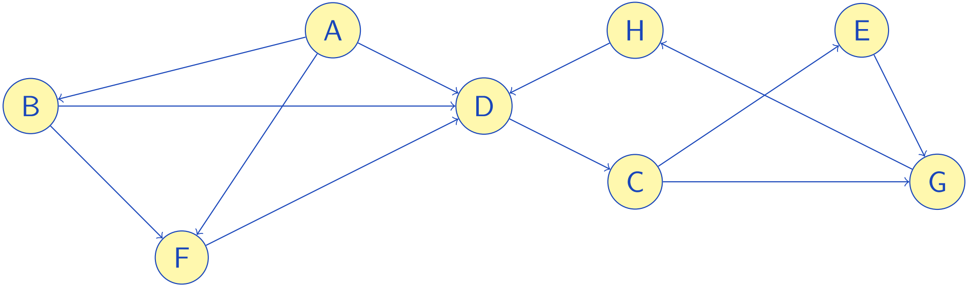
Cycle in the graph $G = \langle V, E \rangle$

- A path $v_1, v_2, \ldots, v_k$, such that $v_k = v_1$
- Example: $[C, G, H, D]$ is a cycle



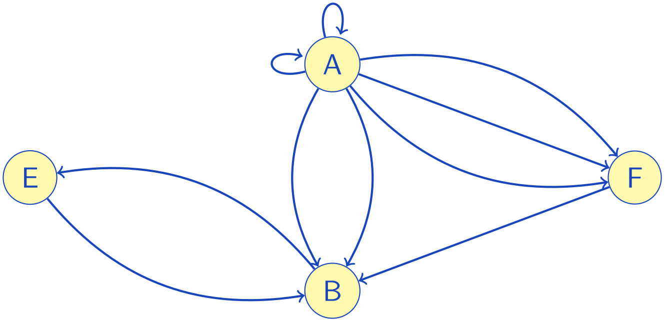
Cycle in the graph $G = \langle V, E \rangle$

- A path $v_1, v_2, \dots, v_k$, such that $v_k = v_1$
- Example: $[C, G, H, D]$ is a cycle
- Simple cycle: no vertex appears twice in the path, except for $v_1 = v_k$



Loop is an edge which connects a vertex to itself
Multiedge is an edge which appears multiple times in E

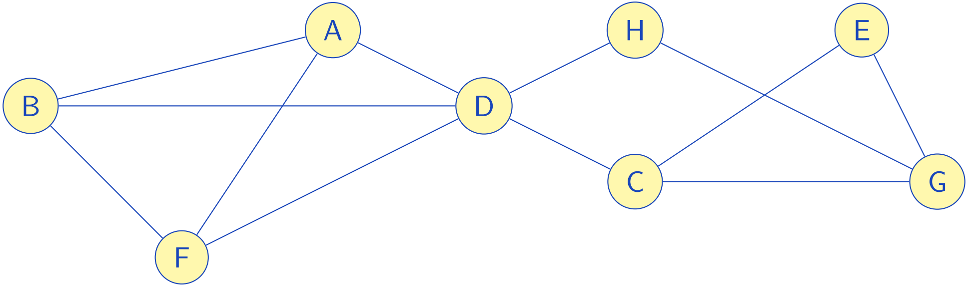
Loop is an edge which connects a vertex to itself
Multiedge is an edge which appears multiple times in E



An undirected graph is connected if there exists a path between any two vertices

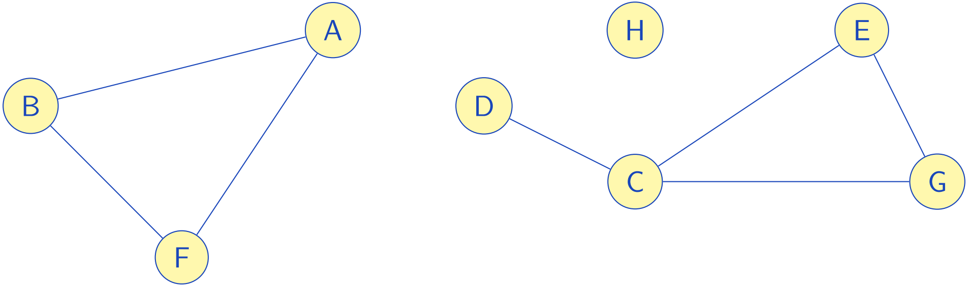
An undirected graph is connected if there exists a path between any two vertices

Example of connected graph

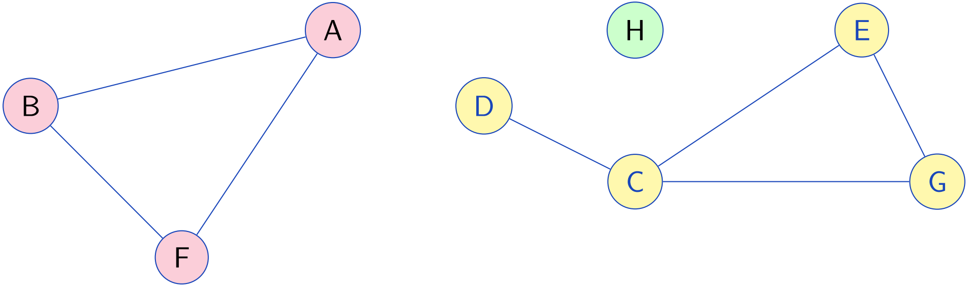


An undirected graph is connected if there exists a path between any two vertices

Example of graph which is not connected



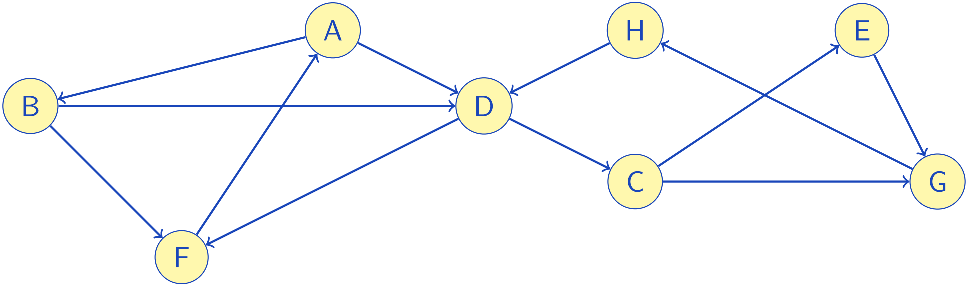
An undirected graph is connected if there exists a path between any two vertices
Example of graph which is not connected – three connected components



A directed graph is strongly connected if, for any two vertices u and v, there exists a path from u to v, and a path from v to u

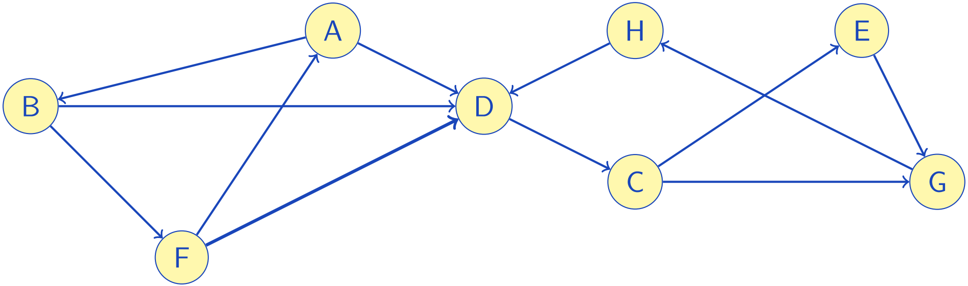
A directed graph is strongly connected if, for any two vertices u and v, there exists a path from u to v, and a path from v to u

Example of strongly connected graph



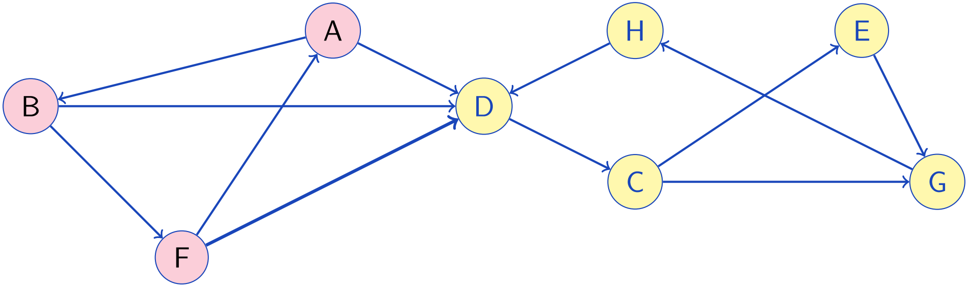
A directed graph is strongly connected if, for any two vertices u and v, there exists a path from u to v, and a path from v to u

Example of graph which is not strongly connected



A directed graph is strongly connected if, for any two vertices u and v, there exists a path from u to v, and a path from v to u

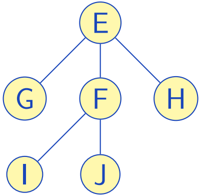
Example of graph which is not strongly connected – two strongly connected components



There exist several important types of graphs. Some of these frequently appear in programming competitions. Here they are:

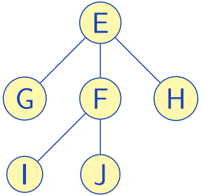
- Trees, rooted trees and forests
- Cactuses
- Complete graphs and tournaments
- Bipartite graphs
- Planar graphs

Tree: an undirected graph in which any two vertices are connected by exactly one path



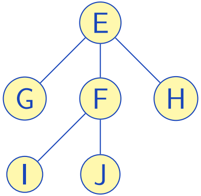
Tree: an undirected graph in which any two vertices are connected by exactly one path

- A tree does not contain any cycles. Graphs without cycles are called acyclic



Tree: an undirected graph in which any two vertices are connected by exactly one path

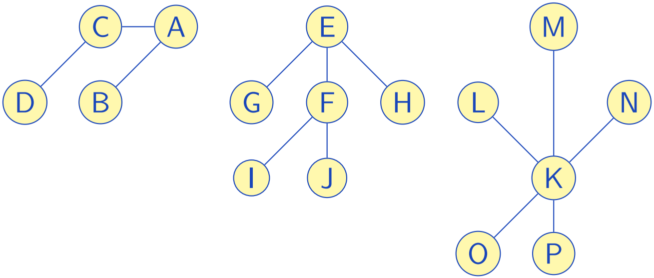
- A tree does not contain any cycles. Graphs without cycles are called acyclic
- In any tree, $|E| = |V| - 1$



Tree: an undirected graph in which any two vertices are connected by exactly one path

- A tree does not contain any cycles. Graphs without cycles are called acyclic
- In any tree, $|E| = |V| - 1$

Forest: a disjoint union of trees



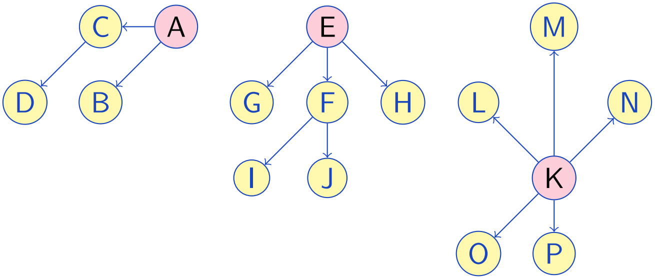
Tree: an undirected graph in which any two vertices are connected by exactly one path

- A tree does not contain any cycles. Graphs without cycles are called acyclic
- In any tree, $|E| = |V| - 1$

Forest: a disjoint union of trees

Rooted tree: a tree where one of the vertices is a root.

All edges have direction either from or towards root



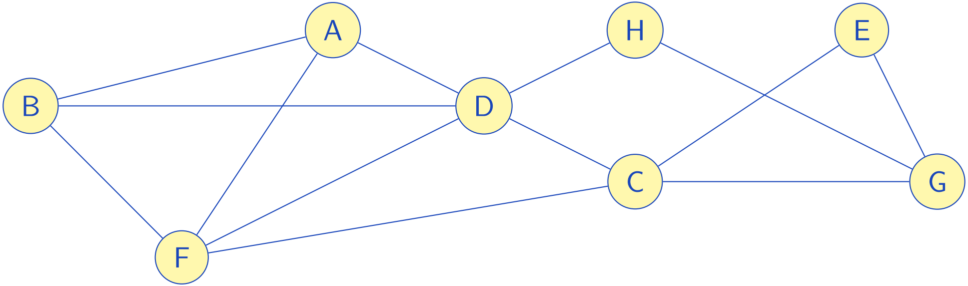
Spanning tree: a subgraph of an undirected graph G, which is a tree and contains all vertices of G

Spanning tree: a subgraph of an undirected graph G, which is a tree and contains all vertices of G

- A graph $G' = \langle V', E' \rangle$ is a subgraph of $G = \langle V, E \rangle$ if:
  - $V' \subseteq V$, $E' \subseteq E$
  - For all $(u, v) \in E'$, $u \in V'$ and $v \in V'$

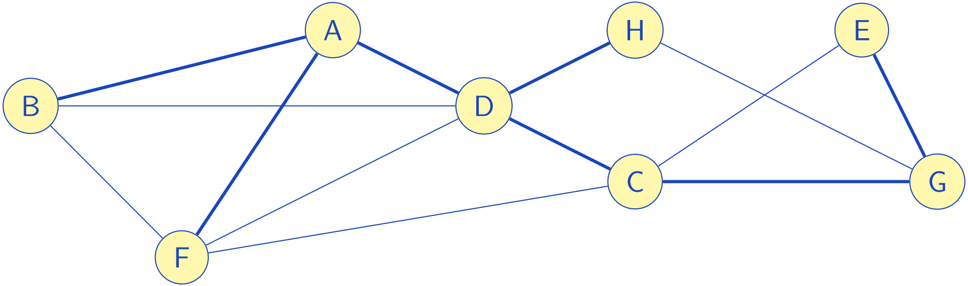
Spanning tree: a subgraph of an undirected graph G, which is a tree and contains all vertices of G

- A graph $G' = \langle V', E' \rangle$ is a subgraph of $G = \langle V, E \rangle$ if:
  - $V' \subseteq V$, $E' \subseteq E$
  - For all $(u, v) \in E'$, $u \in V'$ and $v \in V'$

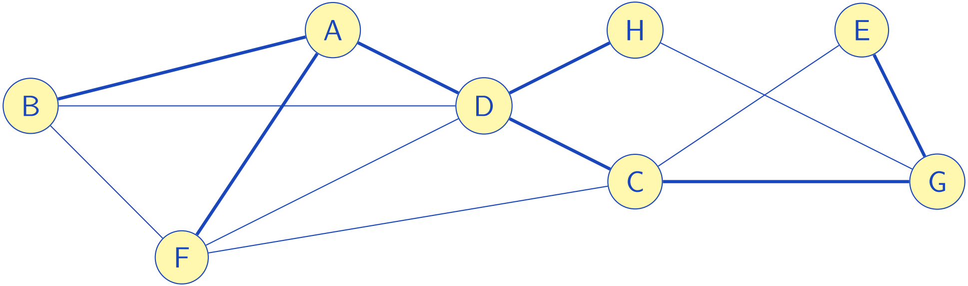


Spanning tree: a subgraph of an undirected graph G, which is a tree and contains all vertices of G

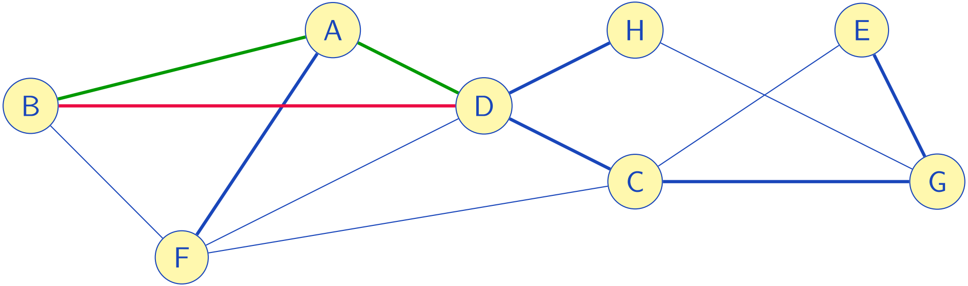
- A graph $G' = \langle V', E' \rangle$ is a subgraph of $G = \langle V, E \rangle$ if:
  - $V' \subseteq V$, $E' \subseteq E$
  - For all $(u, v) \in E'$, $u \in V'$ and $v \in V'$



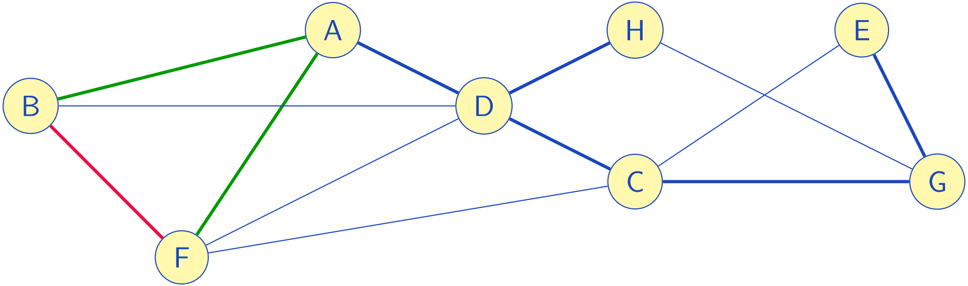
Fix a spanning tree T of graph G. Every edge of G not in T forms a cycle with T. These are the fundamental cycles with regards to T.



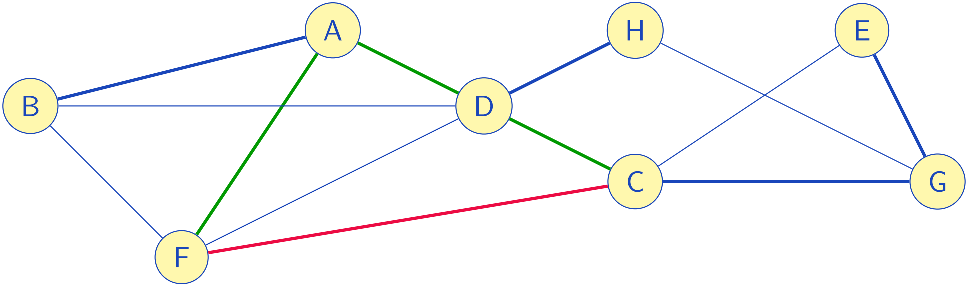
Fix a spanning tree T of graph G. Every edge of G not in T forms a cycle with T. These are the fundamental cycles with regards to T.



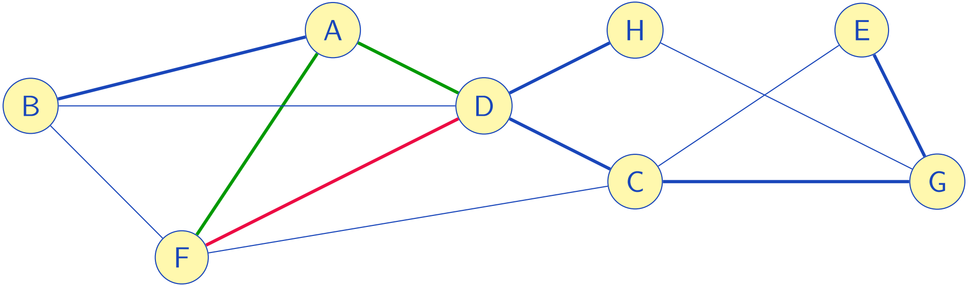
Fix a spanning tree T of graph G. Every edge of G not in T forms a cycle with T. These are the fundamental cycles with regards to T.



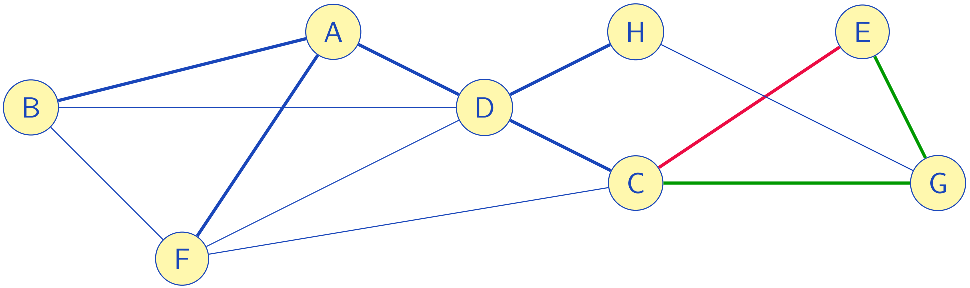
Fix a spanning tree T of graph G. Every edge of G not in T forms a cycle with T. These are the fundamental cycles with regards to T.



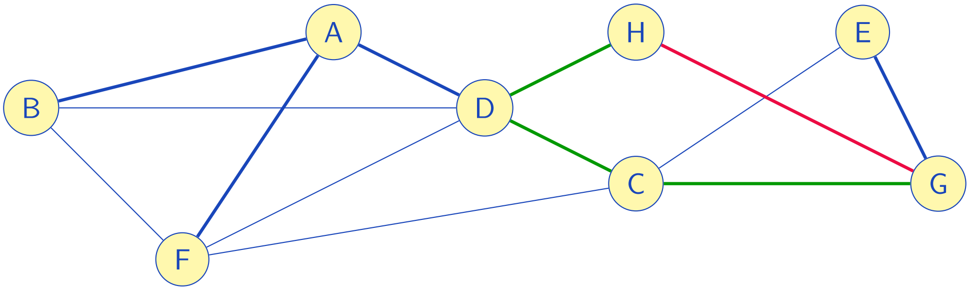
Fix a spanning tree T of graph G. Every edge of G not in T forms a cycle with T. These are the fundamental cycles with regards to T.



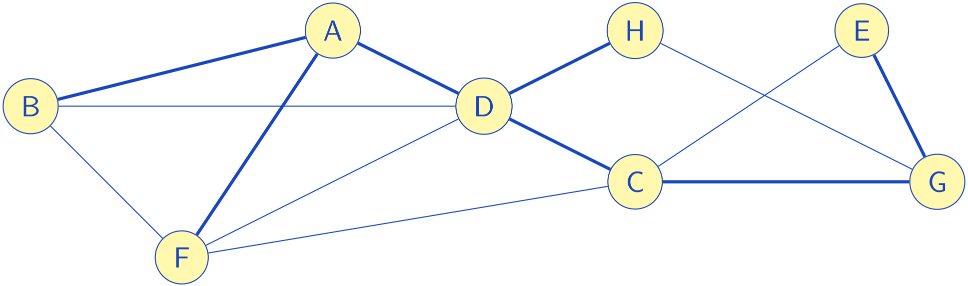
Fix a spanning tree T of graph G. Every edge of G not in T forms a cycle with T. These are the fundamental cycles with regards to T.



Fix a spanning tree T of graph G. Every edge of G not in T forms a cycle with T. These are the fundamental cycles with regards to T.



Fix a spanning tree T of graph G. Every edge of G not in T forms a cycle with T.
These are the fundamental cycles with regards to T.
Any cycle can be uniquely expressed as a combination of fundamental cycles



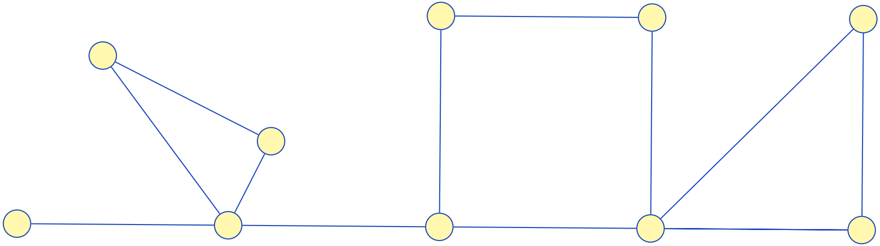
Edge-cactus: connected, undirected, any two simple cycles share at most one vertex

Edge-cactus: connected, undirected, any two simple cycles share at most one vertex

- Each edge belongs to at most one cycle

Edge-cactus: connected, undirected, any two simple cycles share at most one vertex

- Each edge belongs to at most one cycle



Edge-cactus: connected, undirected, any two simple cycles share at most one vertex

- Each edge belongs to at most one cycle

Vertex-cactus: any two simple cycles share no common vertices

Edge-cactus: connected, undirected, any two simple cycles share at most one vertex

- Each edge belongs to at most one cycle

Vertex-cactus: any two simple cycles share no common vertices

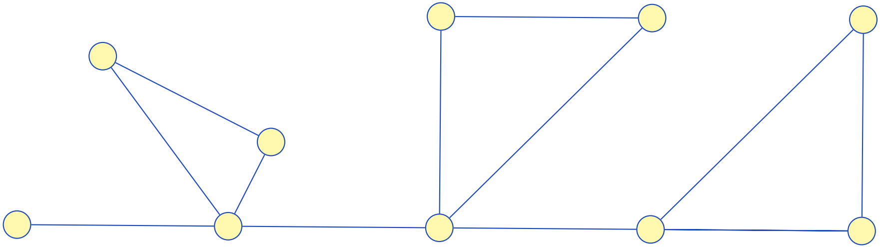
- Each vertex belongs to at most one cycle

Edge-cactus: connected, undirected, any two simple cycles share at most one vertex

- Each edge belongs to at most one cycle

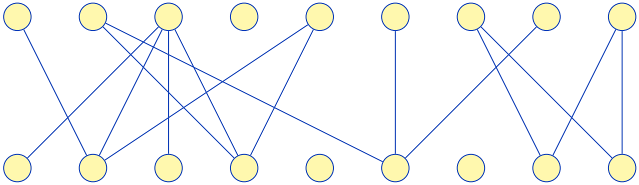
Vertex-cactus: any two simple cycles share no common vertices

- Each vertex belongs to at most one cycle



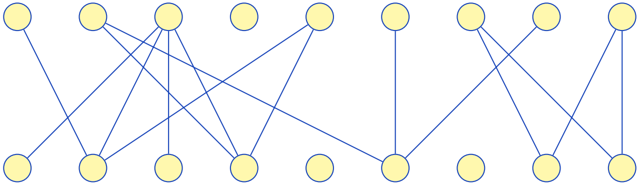
Bipartite graph: a undirected graph where $V = L \cup R$, $L \cap R = \emptyset$, such that for any edge (u, v) it holds that $u \in L$, $v \in R$

Bipartite graph: a undirected graph where $V = L \cup R$, $L \cap R = \emptyset$, such that for any edge (u, v) it holds that $u \in L$, $v \in R$



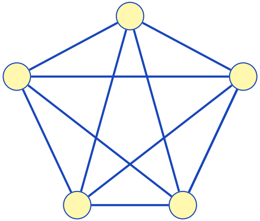
Bipartite graph: a undirected graph where $V = L \cup R$, $L \cap R = \emptyset$, such that for any edge (u, v) it holds that $u \in L$, $v \in R$

- In any bipartite graph, every loop has an even length

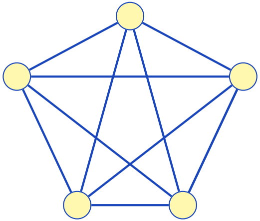


Complete graph K_n: an undirected graph with edges between all pairs of vertices

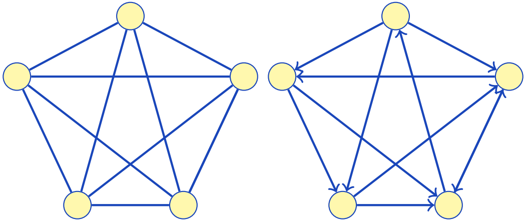
Complete graph K_n: an undirected graph with edges between all pairs of vertices



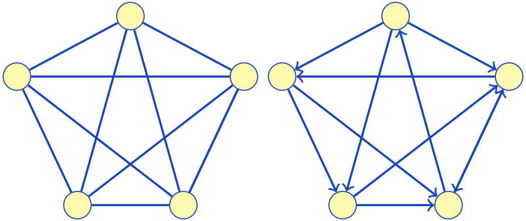
Complete graph K_n: an undirected graph with edges between all pairs of vertices
Tournament T_n: a directed graph with either (u, v) or (v, u) for all u and v



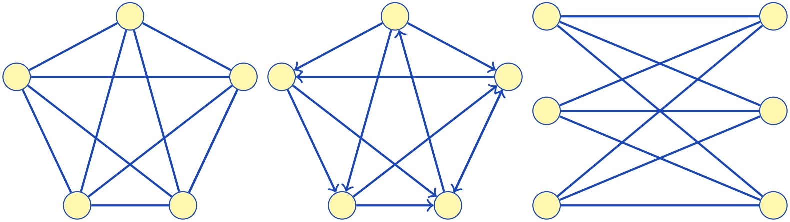
Complete graph K_n: an undirected graph with edges between all pairs of vertices
Tournament T_n: a directed graph with either (u, v) or (v, u) for all u and v



Complete graph K_n: an undirected graph with edges between all pairs of vertices
Tournament T_n: a directed graph with either (u, v) or (v, u) for all u and v
Complete bipartite graph $K_{n,m}$: an undirected graph with edges between all pairs of vertices from different parts

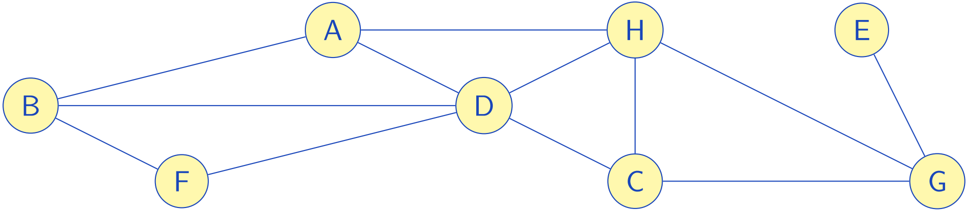


Complete graph K_n: an undirected graph with edges between all pairs of vertices
Tournament T_n: a directed graph with either (u, v) or (v, u) for all u and v
Complete bipartite graph $K_{n,m}$: an undirected graph with edges between all pairs of vertices from different parts



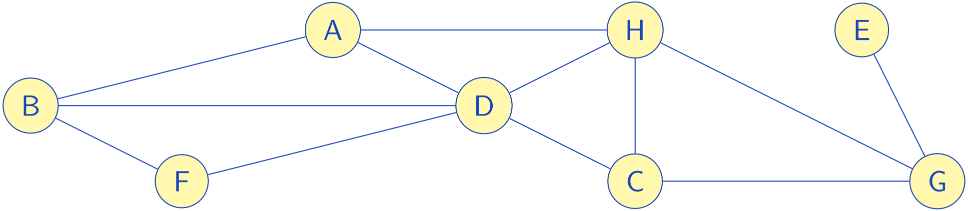
Planar graph: a graph which can be laid out on the plane without edges intersections

Planar graph: a graph which can be laid out on the plane without edges intersections



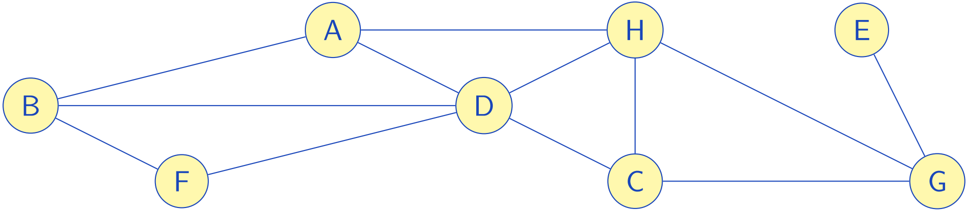
Planar graph: a graph which can be laid out on the plane without edges intersections

- Face: a piece of plane bounded by a loop in the graph never intersected by edges



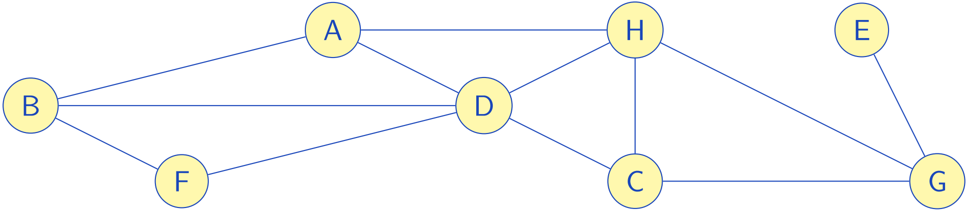
Planar graph: a graph which can be laid out on the plane without edges intersections

- Face: a piece of plane bounded by a loop in the graph never intersected by edges
- Fact 1 (Euler characteristic): $|V| - |E| + |F| = 2$.



Planar graph: a graph which can be laid out on the plane without edges intersections

- Face: a piece of plane bounded by a loop in the graph never intersected by edges
- Fact 1 (Euler characteristic): $|V| - |E| + |F| = 2$.
- Fact 2: If $|V| \geq 3$ then $|E| < 3|V| - 6$



Planar graph: a graph which can be laid out on the plane without edges intersections

- Face: a piece of plane bounded by a loop in the graph never intersected by edges
- Fact 1 (Euler characteristic): $|V| - |E| + |F| = 2$.
- Fact 2: If $|V| \geq 3$ then $|E| < 3|V| - 6$
- Fact 3 (Kuratowski): The graph is planar if and only if it doesn't contain subgraphs which are subdivision of K_5 or $K_{3,3}$. Subdivision of the graph is introducing some number of vertices in each edge.