Graphs

CS 220 — Applied Discrete Mathematics

Draft

Graphs

Definition (Graph)

A **graph** G is a pair (V, E), where

- V is a set of **vertices** or **nodes**
- E is a set of **edges**, where each edge joins two vertices

There are two main types of graphs: directed and undirected.
(There are other kinds, but those are less common.)

In addition, graphs may be augmented with additional information.

Directed Graphs

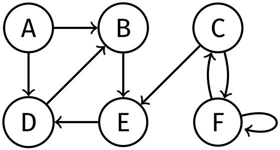
Definition (Directed Graph)

A **directed graph** or **digraph** G is a pair (V, E), where

- V is a set of vertices
- $E \subseteq V \times V$ is a set of edges

That is, an edge is an ordered pair; the edge (u, v) goes from vertex u to vertex v. Self-loops are allowed.

We have used directed graphs before to represent relations.



Undirected Graphs

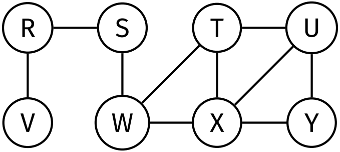
Definition (Undirected Graph)

A **undirected graph** G is a pair (V, E), where

- V is a set of vertices
- $E \subseteq \{\{u, v\} \mid u, v \in V,\ u \neq v\}$ is a set of edges

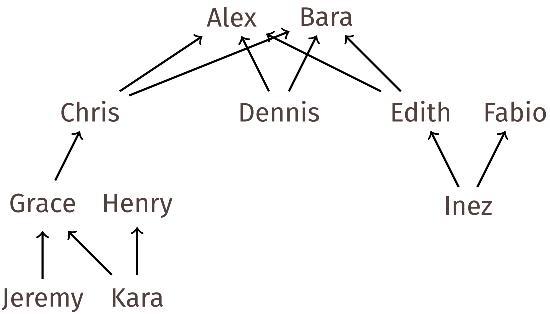
That is, an edge is an unordered pair—that is, a set of two distinct vertices.
The edge $\{u, v\}$ is the same as $\{v, u\}$; it connects u and v without direction.

Self-loops are not allowed.

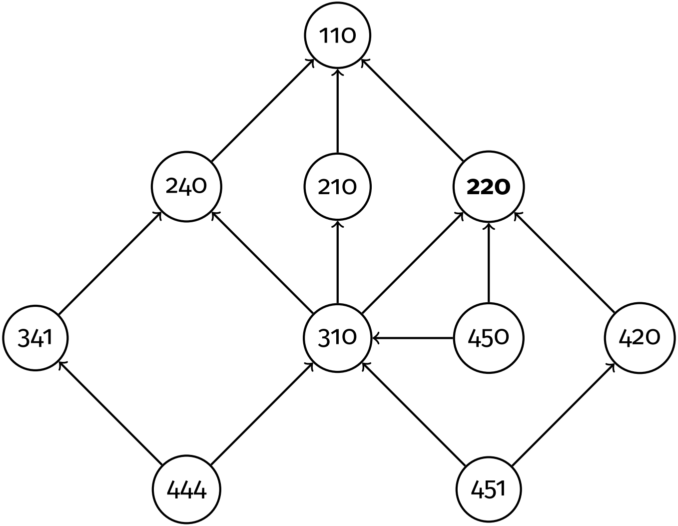


Undirected graph edges are sometimes written as (u, v) anyway, in a misuse of notation.

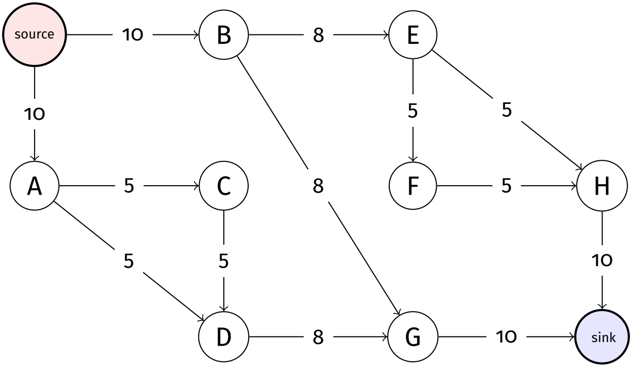
Applications: Ancestry Graph



Applications: Dependency Graph

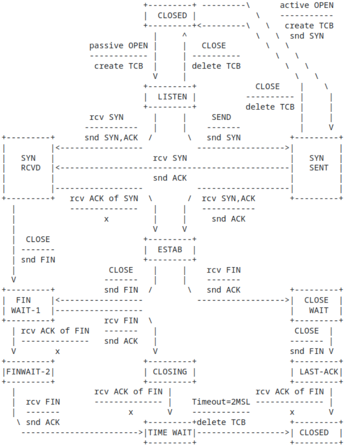


Applications: Flow Network

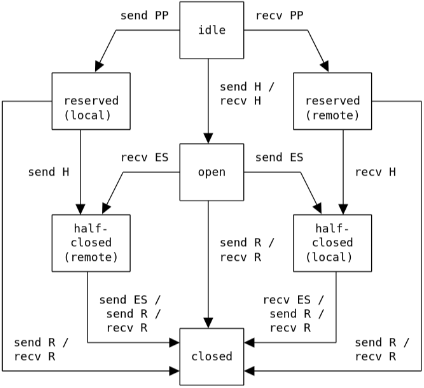


This is a “flow network”: a directed graph with distinguished *source* and *sink* vertices and a *capacity function* that assigns each edge a nonnegative real capacity.

Applications: Finite State Machines

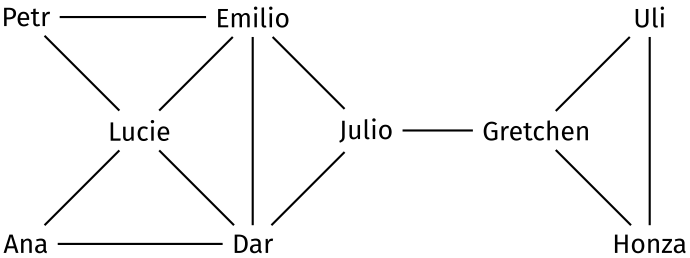


RFC 793: Transmission Control Protocol (1981)

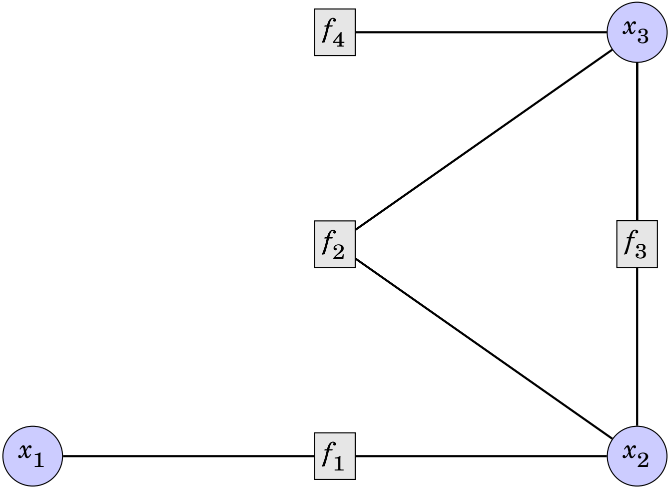


RFC 9113: HTTP/2, §5.1

Applications: Social Network



Applications: Factor Graph



Directed Graphs

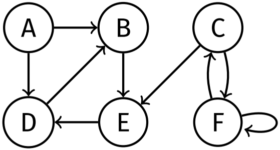
Basic Terminology

Definitions (Adjacent, Initial/Terminal Vertex)

Let G be a directed graph, and suppose (u, v) is an edge in G.

Then we say that

- u is **adjacent to** v
- v is **adjacent from** u
- u is the **initial vertex** of (u, v)
- v is the **terminal vertex** of (u, v)

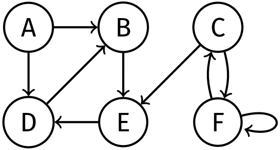


In-Degree and Out-Degree

Definitions (In-Degree, Out-Degree)

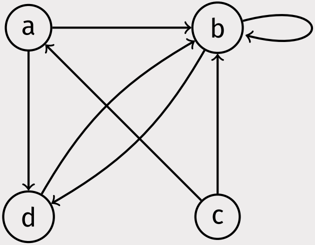
Let G be a directed graph, and let v be a vertex in G.

- The **in-degree** of v, denoted by $\deg^{-}(v)$, is the number of edges with v as their terminal vertex (that is, edges coming **in** to v).
- The **out-degree** of v, denoted by $\deg^{+}(v)$, is the number of edges with v as their initial vertex (that is, edges going **out** from v).



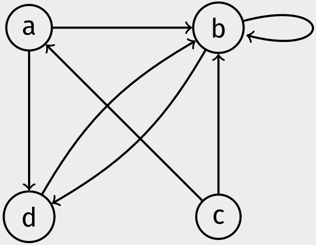
Example: In-Degree and Out-Degree

What are the in-degrees and out-degrees of the vertices a, b, c, d?



Example: In-Degree and Out-Degree

What are the in-degrees and out-degrees of the vertices a, b, c, d?



$$\deg^-(a) = 1 \qquad \deg^+(a) = 2$$
$$\deg^-(b) = 4 \qquad \deg^+(b) = 2$$
$$\deg^-(c) = 0 \qquad \deg^+(c) = 2$$
$$\deg^-(d) = 2 \qquad \deg^+(d) = 1$$

Degrees and Edges

Theorem

Let $G = (V, E)$ be a directed graph. Then:

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

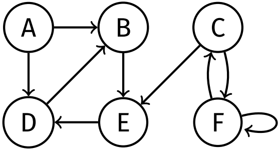
This is easy to see, because every new edge increases both the sum of in-degrees and the sum of out-degrees by one.

Walks, Trails, Paths, Circuits, and Cycles

Definitions (*)

Let $G = (V, E)$ be a directed graph.

- A **walk** from x_0 to x_n is a sequence of vertices $x_0, \ldots, x_n$ such that each $(x_k, x_{k+1}) \in E$. The **length** of a walk is the number of edges it contains. So the length of the walk $x_0, \ldots, x_n$ is n.
- A **trail** is a walk that does not use any edge more than once.
- A **path** is a walk that does not use any vertex more than once.
- A **circuit** is a trail that starts and ends with the same vertex.
- A **cycle** is a circuit with at least one edge that does not repeat any vertex except the start and end vertex (which appear twice).



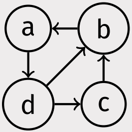
Strongly Connected

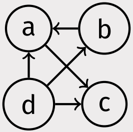
Definitions (Strongly Connected)

A directed graph is **strongly connected** if there is a path from u to v for every pair of vertices u and v in the graph.

Note: There is always a 0-length path from u to itself.

Examples





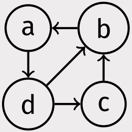
Strongly Connected

Definitions (Strongly Connected)

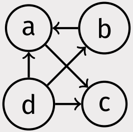
A directed graph is **strongly connected** if there is a path from u to v for every pair of vertices u and v in the graph.

Note: There is always a 0-length path from u to itself.

Examples



strongly connected



not strongly connected
eg, no path from c to a

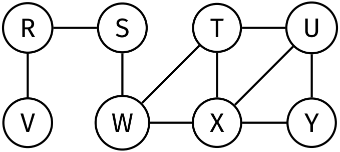
Undirected Graphs

Basic Terminology

Definitions (Adjacent, Incident, ...)

Let G be an undirected graph, and suppose $\{u, v\}$ is an edge in G.
Then we say

- u and v are **adjacent**
- u and v are both **incident** with the edge
- u and v are the **endpoints** of the edge



Degree

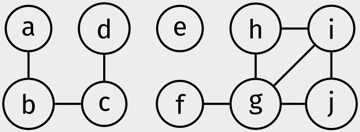
Definitions (Degree, ...)

The **degree** of a vertex in an undirected graph is the number of edges incident with it. The degree of the vertex v is denoted by $\deg(v)$.

A vertex of degree 0 is called **isolated**.
A vertex of degree 1 is called **pendant**.

Example

Which vertices in the following graph are isolated? Which are pendant? Which vertices have the the maximum degree, and what is it?



The Handshaking Theorem

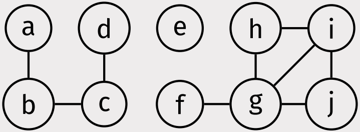
Theorem (Handshaking Theorem)

Let $G = (V, E)$ be an undirected graph. Then $\sum_{v \in V} \deg(v) = 2|E|$.

Corollary

An undirected graph has an even number of vertices of odd degree.

Example



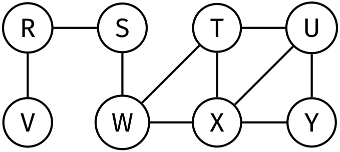
There are 9 edges, and the sum of all vertex degrees is 18.
Which vertices have odd degree?

Walks, Trails, Paths, Circuits, and Cycles

Definitions (*)

Let $G = (V, E)$ be an undirected graph.

- A **walk** from x_0 to x_n is a sequence of vertices $x_0, \dots, x_n$ such that each $\{x_k, x_{k+1}\} \in E$. The **length** of a walk is the number of edges it contains. So the length of the walk $x_0, \dots, x_n$ is n.
- A **trail** is a walk that does not use any edge more than once.
- A **path** is a walk that does not use any vertex more than once.
- A **circuit** is a trail that starts and ends with the same vertex.
- A **cycle** is a circuit with at least one edge that does not repeat any vertex except the start and end vertex (which appear twice).



Connected Graphs

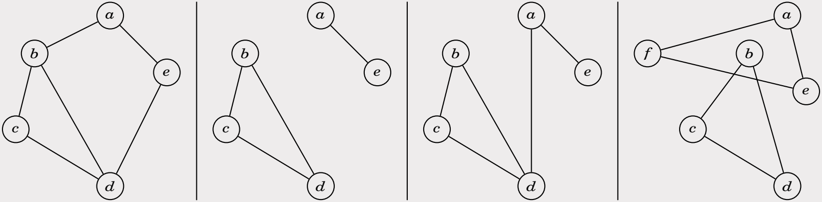
Definition (Connected)

An undirected graph is **connected** if there is a path between every pair of distinct vertices in the graph. Otherwise it is **disconnected**.

Note: A graph with only one vertex is always connected, because it does not contain any pair of distinct vertices. So is the empty graph.

Examples

Which of these graphs are connected?



Connected Components

Definition (Connected Component)

Let $G = (V, E)$ be an undirected graph.

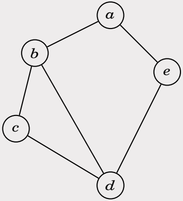
There is a partition of the vertices V into n sets $\{V_1, \ldots, V_n\}$ such that

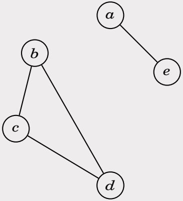
- Every V_k is connected. That is, for each $k \in [1 \,..\, n]$, there is a path between every pair of distinct vertices in V_k.
- Every V_i is disconnected from every other V_j. That is, for every $i, j \in [1 \,..\, n]$ where $i \neq j$, there is no path from any vertex in V_i to any vertex in V_j.

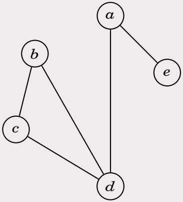
Each V_k is a **connected component** of G. If G is connected, $n = 1$.

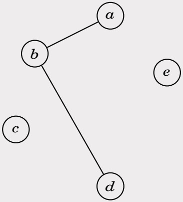
Exercise: Connected Components

What is the set of connected components of each graph below?







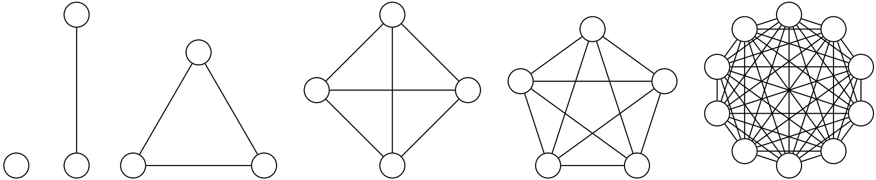


Special Undirected Graphs

Complete Graph

Definition (Complete Graph)

The **complete graph** on n vertices, denoted by K_n, is the undirected graph that contains exactly one edge between each pair of distinct vertices.

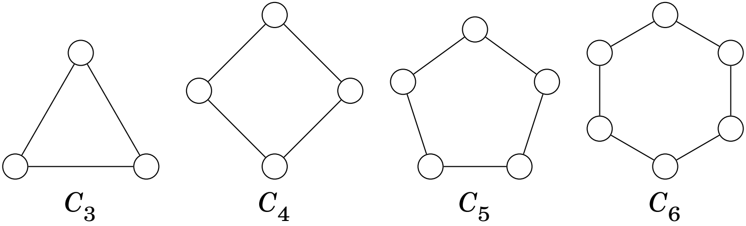


K_1 K_2 K_3 K_4 K_5 K_{10}

Cyclic Graph

Definition (Cyclic Graph)

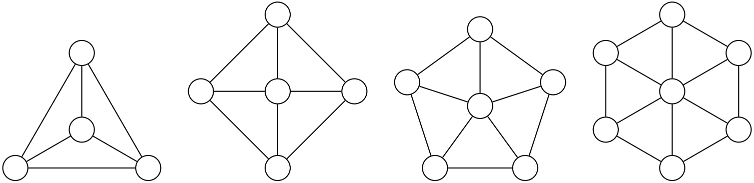
The **cyclic graph** C_n where $n \geq 3$, consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.



Wheel

Definition (Wheel)

The **wheel** W_n, for $n \geq 3$, is obtained by adding an additional vertex to the cyclic graph C_n and adding an edge connecting this new vertex to each of the n vertices in C_n.

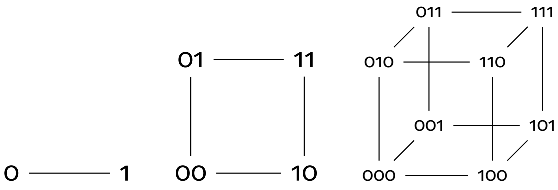


W_3 W_4 W_5 W_6

n-Cube

Definition (n-Cube)

The **n-cube**, denoted by Q_n, is the undirected graph that has vertices representing 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 position.



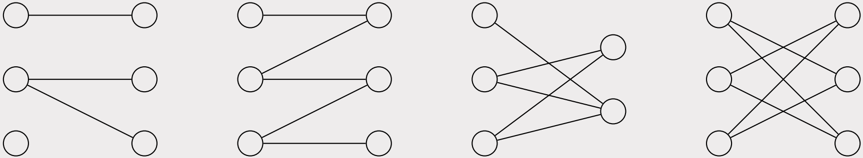
Q_1 $\quad$ Q_2 $\quad$ Q_3

Bipartite Graphs

Definition (Bipartite)

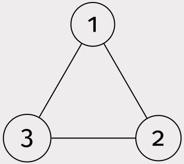
An undirected graph is called **bipartite** if its vertex set V can be partitioned into two disjoint sets V_1 and V_2 such that every edge in the graph connects a vertex in V_1 with a vertex in V_2. That is, no edge in G connects either two vertices in V_1 or two vertices in V_2.

Examples

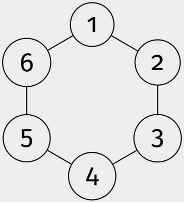


Examples: Bipartite Graphs

Example: is C_3 bipartite?

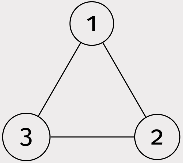


Example: is C_6 bipartite?



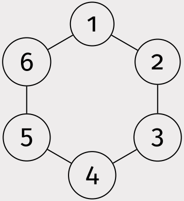
Examples: Bipartite Graphs

Example: is C_3 bipartite?



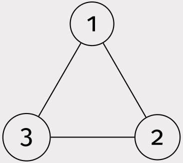
No, because there is no way to partition the vertices into two sets so that there are no edges with both endpoints in the same set.

Example: is C_6 bipartite?



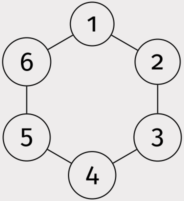
Examples: Bipartite Graphs

Example: is C_3 bipartite?

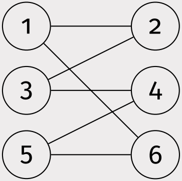


No, because there is no way to partition the vertices into two sets so that there are no edges with both endpoints in the same set.

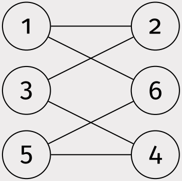
Example: is C_6 bipartite?



Yes, because we can display C_6 like this:



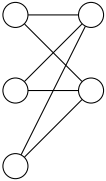
or



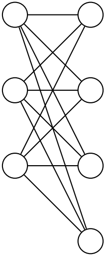
Complete Bipartite Graph

Definition (Complete Bipartite Graph)

The **complete bipartite graph** $K_{m,n}$ is the graph that has its vertex set partitioned into two subsets of m and n vertices, respectively. Two vertices are connected if and only if they are in different subsets.



$K_{3,2}$



$K_{3,4}$

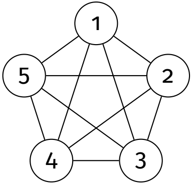
Relationships Between Graphs

Subgraphs

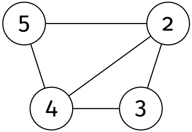
Definition (Subgraph)

A **subgraph** of a directed or undirected graph $G = (V, E)$ is any graph $G' = (V', E')$ where $V' \subseteq V$ and $E' \subseteq E$.

Note: Of course, G' must be a valid graph, so we cannot remove any endpoints of remaining edges when creating G'.



K_5

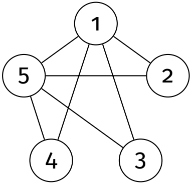


a subgraph of K_5

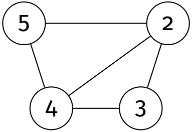
Definition (Union of Graphs)

The **union** of two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the undirected graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$.

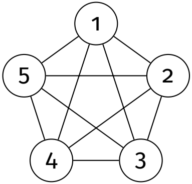
The union of G_1 and G_2 is denoted by $G_1 \cup G_2$.



G_1



G_2



$G_1 \cup G_2 = K_5$

Graph Isomorphism

Definition (Graph Isomorphism)

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be undirected graphs.
Then G_1 and G_2 are **isomorphic** if there is a bijection $f : V_1 \to V_2$ such that $\{u, v\} \in E_1$ if and only if $\{f(u), f(v)\} \in E_2$.

Visually, two graphs are isomorphic if they can be arranged so they look identical except for the labeling of the vertices.

Unfortunately, checking whether graphs are isomorphic can be expensive:

If G_1 and G_2 are two undirected graphs, each with n vertices, then there are $P(n, n) = n!$ different bijections from G_1 to G_2.

Disproving Graph Isomorphism

Deciding whether two graphs are isomorphic is expensive in general, because there are many potential vertex bijections to search.

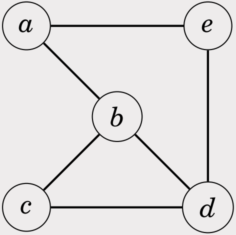
But there are shortcuts that quickly answer “no” for *some* pairs of graphs.

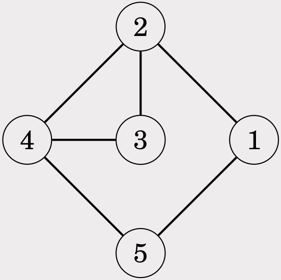
Two graphs cannot be isomorphic if

- they have different numbers of vertices
- they have different numbers of edges
- they have different numbers of vertices of degree d
- one graph has a path of length n and the other does not
- one graph has a cycle of length n and the other does not

Examples: Graph Isomorphism

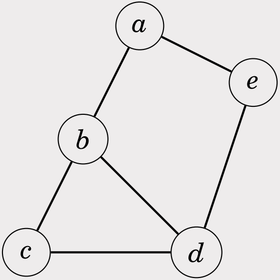
Are the following graphs isomorphic?

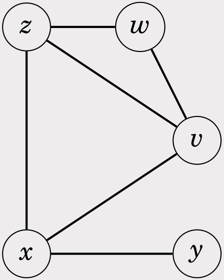




Example: Graph Isomorphism (2)

Are the following graphs isomorphic?





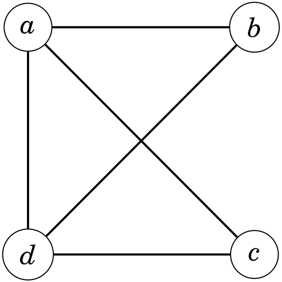
Representing Graphs

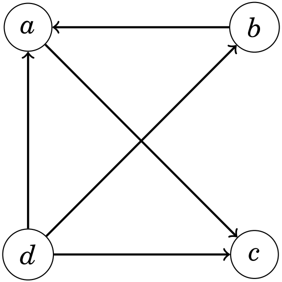
Graph Representations

There are two main ways to represent a graph's edges:

- adjacency lists
- adjacency matrices

Adjacency Lists

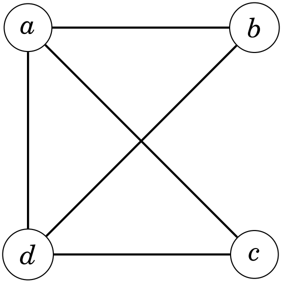


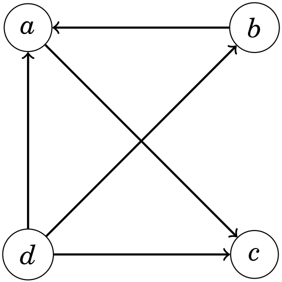


Vertex	Adjacent Vertices
a	b, c, d
b	a, d
c	a, d
d	a, b, c

Vertex	Adjacent Vertices
a	c
b	a
c	
d	a, b, c

Adjacency Matrices





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

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

Trees

Trees

Definition (Tree, Forest)

A **tree** is a undirected graph that is connected and contains no cycles.

A **forest** is a undirected graph that contains no cycles. It can be partitioned into a set of trees.

Theorem

An undirected graph is a tree if and only if there is a unique path between any two vertices in the graph.

Theorem

A tree with n vertices has $n - 1$ edges.

Spanning Trees

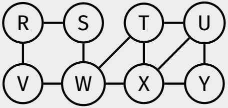
Definition (Spanning Tree)

Let G be an undirected graph. A **spanning tree** of G is a subgraph of G that is a tree and contains every vertex of G.

Theorem

Let G be an undirected graph. There is a spanning tree of G if and only if G is connected. The spanning tree is not necessarily unique.

Example



Spanning Trees

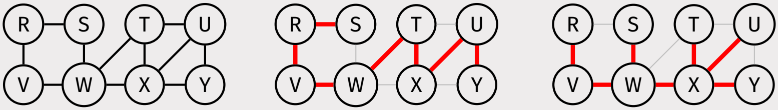
Definition (Spanning Tree)

Let G be an undirected graph. A **spanning tree** of G is a subgraph of G that is a tree and contains every vertex of G.

Theorem

Let G be an undirected graph. There is a spanning tree of G if and only if G is connected. The spanning tree is not necessarily unique.

Example



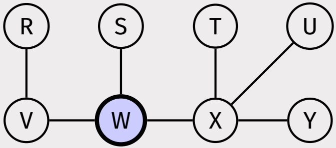
Rooted Trees

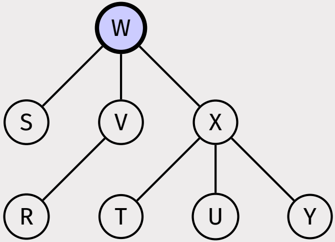
Definition (Rooted Tree)

A **rooted tree** is a tree with a distinguished **root** vertex.

Since there is a unique path from the root to every vertex in the graph, a rooted tree has “directions” (toward the root, away from the root), even though it is an undirected graph.

Example





Rooted Tree Relationships

Definitions (Descendant, Ancestor)

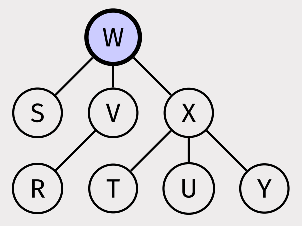
Let $T = (V, E)$ be a rooted tree, and let $u, v \in V$.

If the unique path from u to the root contains v, then we say

- u is a **descendant** of v, and
- v is an **ancestor** of u.

If furthermore $\{u, v\} \in E$, then

- u is a **child** of v, and
- v is the **parent** of u.



Every vertex is a descendant and ancestor of itself.
The root vertex is an ancestor of every vertex.
Every vertex is a descendant of the root vertex.

Summary

Topics

- directed graph, undirected graph
- adjacent, in-degree, out-degree, degree
- walk, trail, path, circuit, cycle
- strongly connected, connected, connected component
- complete graph, cyclic graph, wheel, n-cube
- bipartite graphs
- subgraph, graph union
- graph isomorphism
- graph representations
- tree, forest
- spanning tree, rooted tree