Graph

Section 10 in the texbook

Introduction to Graphs

Definition: A **simple graph** G = (V, E) consists of V, a nonempty set of vertices, and E, a set of **unordered pairs** of distinct elements of V called edges.

A simple graph is just like a directed graph, but with no specified direction of its edges.

Sometimes we want to model **multiple connections** between vertices, which is impossible using simple graphs.

In these cases, we have to use **multigraphs**.

Introduction to Graphs

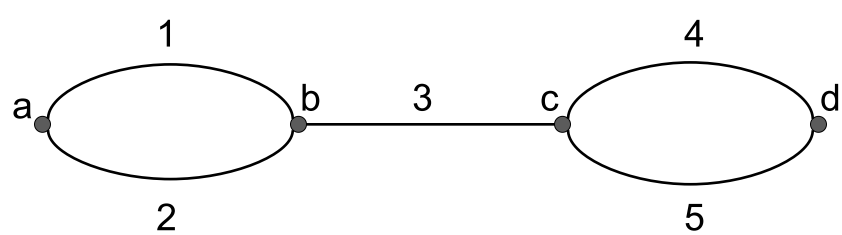
Definition: A **multigraph** G = (V, E) consists of a set V of vertices, a set E of edges, and a function f: $E \rightarrow \{\{u, v\} \mid u, v \in V, u \neq v\}$. The edges e_1 and e_2 are called **multiple or parallel edges** if $f(e_1) = f(e_2)$.

Note:

- Edges in multigraphs are not necessarily defined as pairs, but can be of any type.
- No loops are allowed in multigraphs (u ≠ v).

Introduction to Graphs

Example: A multigraph G with vertices V = {a, b, c, d}, edges {1, 2, 3, 4, 5} and function f with $f(1) = \{a, b\}, f(2) = \{a, b\}, f(3) = \{b, c\}, f(4) = \{c, d\}$ and $f(5) = \{c, d\}$:



Introduction to Graphs

If we want to define loops, we need the following type of graph:

Definition: A **pseudograph** G = (V, E) consists of a set V of vertices, a set E of edges, and a function $f\colon\ E \rightarrow \ \{\{u, v\} \mid u, v \in V\}$.

An edge e is a loop if $f(e)\ =\ \{u, u\}$ for some u∈V.

Introduction to Graphs

Here is a type of graph that we already know:

Definition: A **directed graph** G = (V, E) consists of a set V of vertices and a set E of edges that are ordered pairs of elements in V.

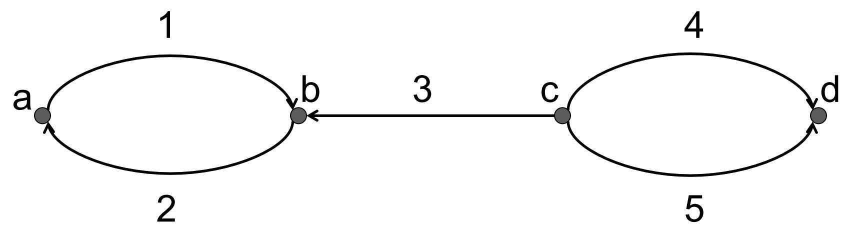
… leading to a new type of graph:

Definition: A **directed multigraph** G = (V, E) consists of a set V of vertices, a set E of edges, and a function $f\ :\ E \rightarrow \{(u, v) \mid u, v \in V\}$.

The edges e_1 and e_2 are called **multiple edges** if $f(e_1)\ =\ f(e_2)$.

Introduction to Graphs

Example: A directed multigraph G with vertices V = {a, b, c, d}, edges {1, 2, 3, 4, 5} and function f with $f(1) = (a,b), f(2) = (b,a), f(3) = (c,b),\ f(4) = (c,d)$ and $f(5) = (c,d)$:



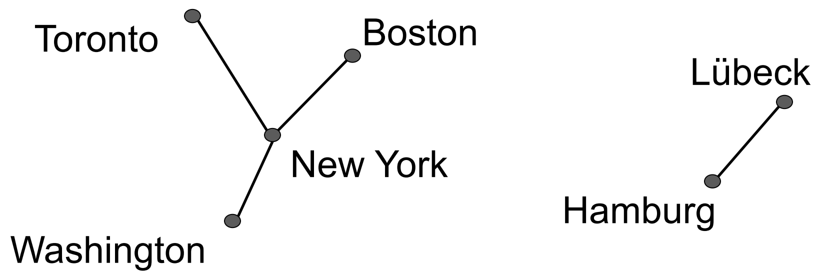
Introduction to Graphs

Type	Edge	Multiple Edges Allowed	Allow loops
Simple graph	Undirected	No	No
Multigraph	Undirected	Yes	No
Pseudo graph	Undirected	Yes	Yes
Simple directed graph	Directed	No	No
Directed multigraph	Directed	Yes	Yes
Mixgraph	Directed & Undirected	Yes	Yes

Graph Models

Example I: How can we represent a network of (bi-directional) railways connecting a set of cities?

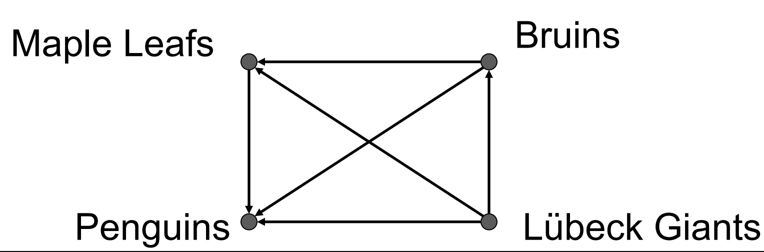
We should use a **simple graph** with an edge {a, b} indicating a direct train connection between cities a and b.



Graph Models

Example II: In a round-robin tournament, each team plays against each other team exactly once. How can we represent the results of the tournament (which team beats which other team)?

We should use a **directed graph** with an edge (a, b) indicating that team a beats team b.



Graph Terminology

Definition: Two vertices u and v in an undirected graph G are called **adjacent** (or **neighbors**) in G if $\{u, v\}$ is an edge in G.

If $e = \{u, v\}$, the edge e is called **incident with** the vertices u and v. The edge e is also said to **connect** u and v.

The vertices u and v are called **endpoints** of the edge $\{u, v\}$.

Graph Terminology

Definition: The **degree** of a vertex in an undirected graph is the number of edges incident with it, except that a loop at a vertex contributes twice to the degree of that vertex.

In other words, you can determine the degree of a vertex in a displayed graph by **counting the lines** that touch it.

The degree of the vertex v is denoted by $\deg(\boldsymbol{v})$**.**

Graph Terminology

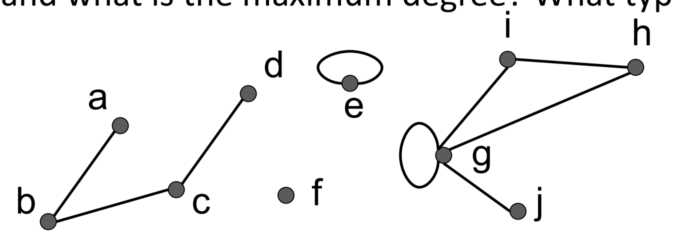
A vertex of degree 0 is called **isolated**, since it is not adjacent to any vertex.

Note: A vertex with a **loop** at it has at least degree 2 and, by definition, is **not isolated**, even if it is not adjacent to any **other** vertex.

A vertex of degree 1 is called **pendant**. It is adjacent to exactly one other vertex.

Graph Terminology

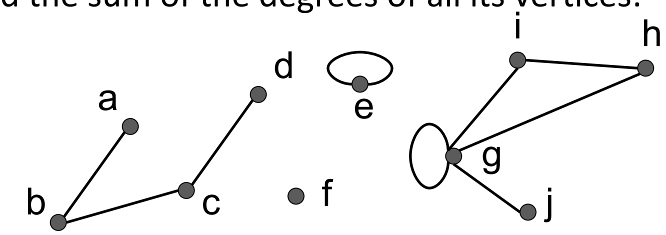
Example: Which vertices in the following graph are isolated, which are pendant, and what is the maximum degree? What type of graph is it?



Solution: Vertex f is isolated, and vertices a, d and j are pendant. The maximum degree is deg(g) = 5.
This graph is a pseudograph (undirected, loops).

Graph Terminology

Let us look at the same graph again and determine the number of its edges and the sum of the degrees of all its vertices:



Result: There are 9 edges, and the sum of all degrees is 18. This is easy to explain: Each new edge increases the sum of degrees by exactly two.

Graph Terminology

The Handshaking Theorem: Let G = (V, E) be an undirected graph with e edges. Then $2e = \sum_{v \in V} \deg(v)$

Note: This theorem holds even if multiple edges and/or loops are present.

Example: How many edges are there in a graph with 10 vertices, each of degree 6?

Solution: The sum of the degrees of the vertices is 6·10 = 60. According to the Handshaking Theorem, it follows that 2e = 60, so there are 30 edges.

Graph Theorems

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

Idea: There are three possibilities for adding an edge to connect two vertices in the graph:

Before:		**After:**
Both vertices have even degree	⇨	Both vertices have odd degree
Both vertices have odd degree	⇨	Both vertices have even degree
One vertex has odd degree, the other even	⇨	One vertex has even degree, the other odd

Graph Theorems

There are two possibilities for adding a loop to a vertex in the graph:

Before:		**After:**
The vertex has even degree	⇨	The vertex has even degree
The vertex has odd degree	⇨	The vertex has odd degree

Graph Terminology

So if there is an even number of vertices of odd degree in the graph, it will still be even after adding an edge.

Therefore, since an undirected graph with **no edges** has an even number of vertices with odd degree (zero), the same must be true for **any** undirected graph.

Graph Terminology

Definition: When (u, v) is an edge of the graph G with directed edges, u is said to be **adjacent to** v, and v is said to be **adjacent from** u.

The vertex u is called the **initial vertex** of (u, v), and v is called the **terminal vertex** of (u, v).

The initial vertex and terminal vertex of a loop are the same.

Graph Terminology

Definition: In a graph with directed edges, the **in-degree** of a vertex v, denoted by $\deg^{-}(\boldsymbol{v})$, is the number of edges with v as their **terminal vertex**.

The **out-degree** of v, denoted by $\deg^{+}(\boldsymbol{v})$, is the number of edges with v as their initial vertex.

Question: How does adding a loop to a vertex change the in-degree and out-degree of that vertex?

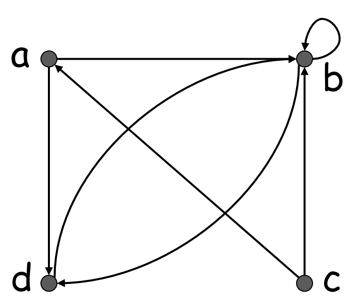
Answer: It increases both the in-degree and the out-degree by one.

Graph Terminology

Example: What are the in-degrees and out-degrees of the vertices a, b, c, d in this graph:

$\deg^{-}(a) = 1$
$\deg^{+}(a) = 2$

$\deg^{-}(b) = 4$
$\deg^{+}(b) = 2$



$\deg^{-}(d) = 2$
$\deg^{+}(d) = 1$

$\deg^{-}(c) = 0$
$\deg^{+}(c) = 2$

Graph Terminology

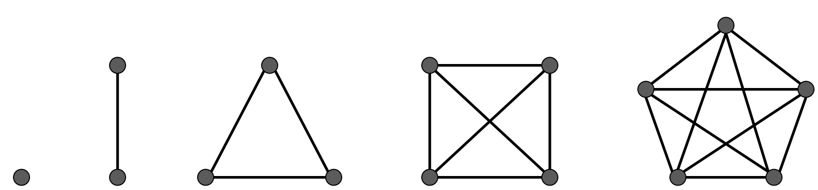
Theorem: Let G = (V, E) be a graph with directed edges. Then:

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

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

Special Graphs

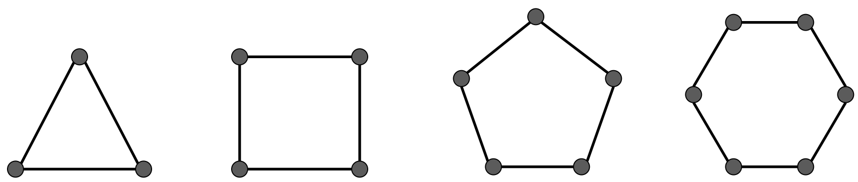
Definition: The **complete graph** on n vertices, denoted by K_n, is the simple graph that contains exactly one edge between each pair of distinct vertices.



K_1 K_2 K_3 K_4 K_5

Special Graphs

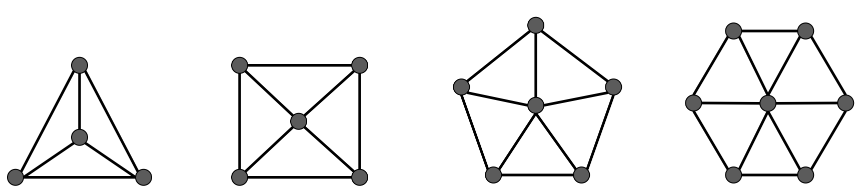
Definition: The **cycle** C_n, $n \geq 3$, consists of n vertices $v_1, v_2, \ldots, v_n$ and edges $\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{n-1}, v_n\}, \{v_n, v_1\}$.



C_3 C_4 C_5 C_6

Special Graphs

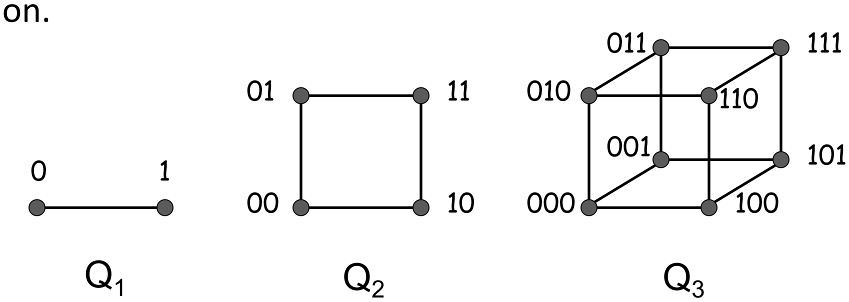
Definition: We obtain the **wheel** W_n when we add an additional vertex to the cycle C_n, for $n \geq 3$, and connect this new vertex to each of the n vertices in C_n by adding new edges.



W_3 W_4 W_5 W_6

Special Graphs

Definition: The **n-cube,** denoted by Q_n, is the 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.



UMass Boston

Special Graphs

Definition: A simple graph is called **bipartite** if its vertex set V can be partitioned into two disjoint nonempty 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 (so that no edge in G connects either two vertices in V_1 or two vertices in V_2).

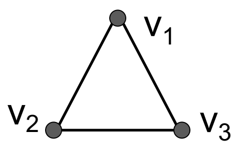
For example, consider a graph that represents each person in a mixed-doubles tennis tournament (i.e., teams consist of one female and one male player). Players of the same team are connected by edges.

This graph is **bipartite**, because each edge connects a vertex in the **subset of males** with a vertex in the **subset of females**.

UMass Boston

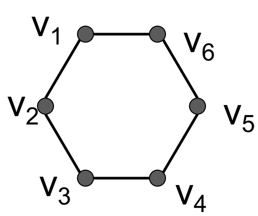
Special Graphs

Example I: Is C_3 bipartite?

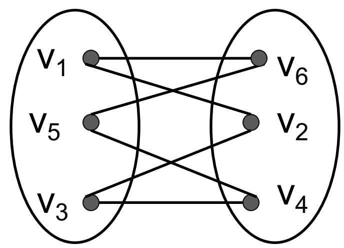


No, because there is no way to partition the vertices into two sets so that there are no edges with both endpoints in the same set.

Example II: Is C_6 bipartite?

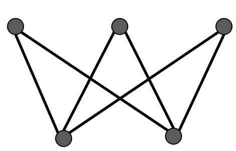


Yes, because we can display C_6 like this:

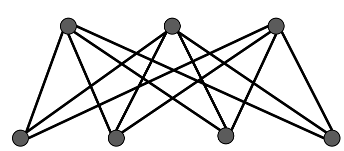


Special Graphs

Definition: 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.



$K_{3,2}$



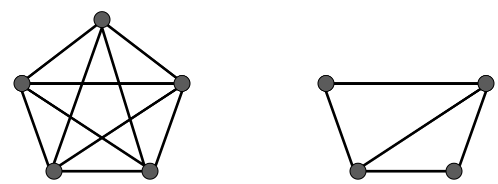
$K_{3,4}$

Operations on Graphs

Definition: A **subgraph** of a graph G = (V, E) is a graph H = (W, F) where $W \subseteq V$ and $F \subseteq E$.

Note: Of course, H is a valid graph, so we cannot remove any endpoints of remaining edges when creating H.

Example:

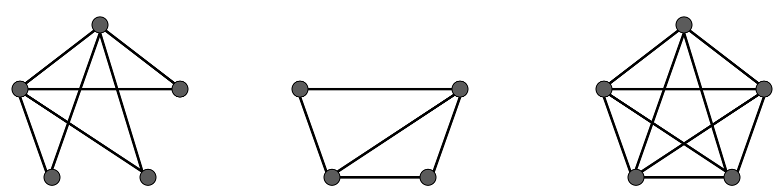


K_5 subgraph of K_5

Operations on Graphs

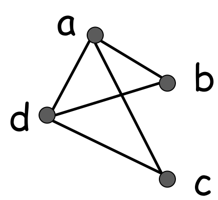
Definition: The **union** of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$.

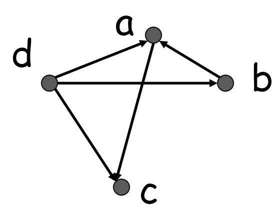
The union of G_1 and G_2 is denoted by $\mathbf{G_1 \cup G_2}$.



G_1 G_2 $G_1 \cup G_2 = K_5$

Representing Graphs





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

Initial Vertex	Terminal Vertices
a	c
b	a
c	
d	a, b, c

Representing Graphs

Definition: Let G = (V, E) be a simple graph with |V| = n. Suppose that the vertices of G are listed in arbitrary order as $v_1, v_2, \ldots, v_n$.

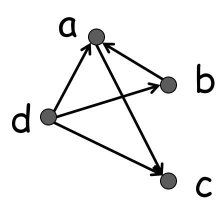
The **adjacency matrix** A (or A_G) of G, with respect to this listing of the vertices, is the n×n zero-one matrix with 1 as its (i, j)th entry when v_i and v_j are adjacent, and 0 otherwise.

In other words, for an adjacency matrix A = $[a_{ij}]$,

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of G} \\ 0 & \text{otherwise} \end{cases}$$

Representing Graphs

Example: What is the adjacency matrix A_G for the following graph G based on the order of vertices a, b, c, d ?



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

Solution:

Note: Adjacency matrices of undirected graphs are always symmetric.

Representing Graphs

For the representation of graphs with **multiple edges**, we can no longer use zero-one matrices.

Instead, we use **matrices of natural numbers**.

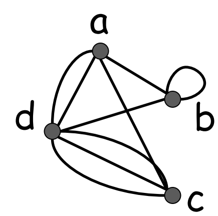
The (i, j)th entry of such a matrix equals the **number of edges** that are associated with $\{v_i, v_j\}$.

Representing Graphs

Example: What is the adjacency matrix A_G for the following graph G based on the order of vertices a, b, c, d ?

Answer:

$$A_G = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 3 \\ 2 & 1 & 3 & 0 \end{bmatrix}$$



Note: For undirected graphs, adjacency matrices are symmetric.

Representing Graphs

Definition: Let G = (V, E) be an undirected graph with |V| = n and |E| = m. Suppose that the vertices and edges of G are listed in arbitrary order as $v_1, v_2, \dots, v_n$ and $e_1, e_2, \dots, e_m$, respectively.

The **incidence matrix** of G with respect to this listing of the vertices and edges is the n×m zero-one matrix with 1 as its (i, j)th entry when edge e_j is incident with vertex v_i, and 0 otherwise.

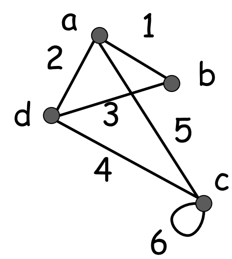
In other words, for an incidence matrix M = $[m_{ij}]$,

$$m_{ij} = \begin{cases} 1 & \text{if edge } e_j \text{ is an incident with } v_i \\ 0 & \text{otherwise} \end{cases}$$

Representing Graphs

Example: What is the incidence matrix M for the following graph G based on the order of vertices a, b, c, d and edges 1, 2, 3, 4, 5, 6?

Answer: $$M = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$$



Note: Incidence matrices of directed graphs contain two 1s per column for edges connecting two vertices and one 1 per column for loops.

Isomorphism of Graphs

Definition: The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are **isomorphic** if there is a bijective function $f : V_1 \rightarrow V_2$ with the property that a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2, for all a and b in V_1.

Such a function f is called an **isomorphism**.

In other words, G_1 and G_2 are isomorphic if their vertices can be ordered in such a way that the adjacency matrices M_{G_1} and M_{G_2} are identical.

Isomorphism of Graphs

From a visual standpoint, G_1 and G_2 are isomorphic if they can be arranged in such a way that their **displays are identical** (of course without changing adjacency).

Unfortunately, for two simple graphs, each with n vertices, there are **n! possible isomorphisms** that we have to check in order to show that these graphs are isomorphic.

However, showing that two graphs are **not** isomorphic can be easy.

Isomorphism of Graphs

For this purpose, we can check **invariants**, that is, properties that two isomorphic simple graphs must both have.

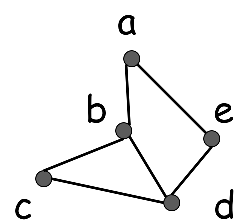
For example, they must have

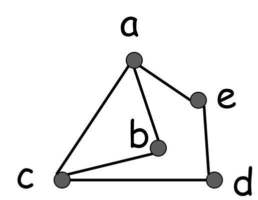
- the same number of vertices,
- the same number of edges, and
- the same degrees of vertices.

Note that two graphs that **differ** in any of these invariants are not isomorphic, but two graphs that **match** in all of them are not necessarily isomorphic.

Isomorphism of Graphs

Example I: Are the following two graphs isomorphic?

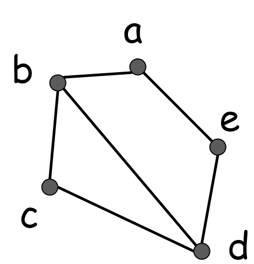


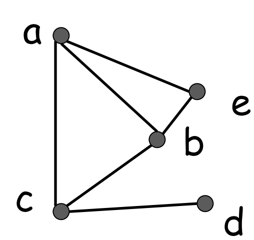


Solution: Yes, they are isomorphic, because they can be arranged to look identical. You can see this if in the right graph you move vertex b to the left of the edge {a, c}. Then the isomorphism f from the left to the right graph is: f(a) = e, f(b) = a,
f(c) = b, f(d) = c, f(e) = d.

Isomorphism of Graphs

Example II: How about these two graphs?





Solution: No, they are not isomorphic, because they differ in the degrees of their vertices.

Vertex d in right graph is of degree one, but there is no such vertex in the left graph.

Connectivity

Definition: A **path** of length n from u to v, where n is a positive integer, in an **undirected graph** is a sequence of edges $e_1, e_2, \ldots, e_n$ of the graph such that $f(e_1) = \{x_0, x_1\}$, $f(e_2) = \{x_1, x_2\}$, …, $f(e_n) = \{x_{n-1}, x_n\}$, where $x_0 = u$ and $x_n = v$. When the graph is simple, we denote this path by its **vertex sequence** $x_0, x_1, \ldots, x_n$, since it uniquely determines the path.

The path is a **circuit** if it begins and ends at the same vertex, that is, if u = v.

Connectivity

Definition (continued): The path or circuit is said to **pass through** or traverse $x_1, x_2, \ldots, x_{n-1}$. A path or circuit is **simple** if it does not contain the same edge more than once.

Connectivity

Definition: A **path** of length n from u to v, where n is a positive integer, in a **directed multigraph** is a sequence of edges $e_1, e_2, \ldots, e_n$ of the graph such that $f(e_1) = (x_0, x_1)$, $f(e_2) = (x_1, x_2)$, …, $f(e_n) = (x_{n-1}, x_n)$, where $x_0 = u$ and $x_n = v$.

When there are no multiple edges in the path, we denote this path by its **vertex sequence** $x_0, x_1, \ldots, x_n$, since it uniquely determines the path.

The path is a **circuit** if it begins and ends at the same vertex, that is, if u = v.

A path or circuit is called **simple** if it does not contain the same edge more than once.

Connectivity

Let us now look at something new:

Definition: An undirected graph is called **connected** if there is a path between every pair of distinct vertices in the graph.

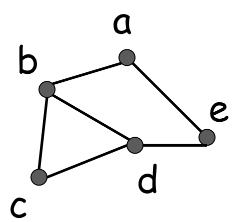
For example, any two computers in a network can communicate if and only if the graph of this network is connected.

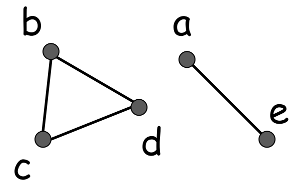
Note: A graph consisting of only one vertex is always connected, because it does not contain any pair of distinct vertices.

Connectivity

Example: Are the following graphs connected?

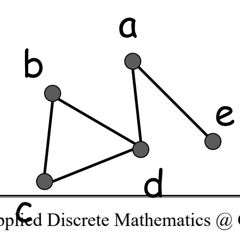
Yes.

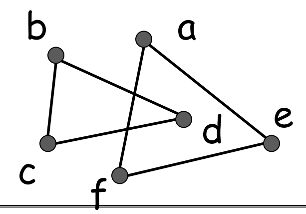




No.

Yes.





No.

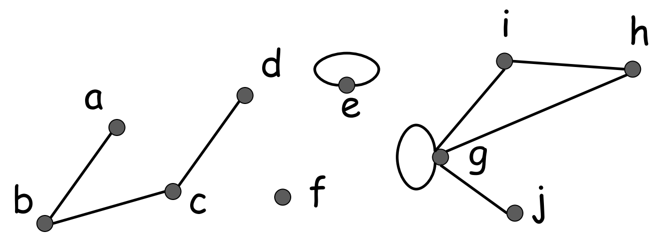
Connectivity

Theorem: There is a **simple** path between every pair of distinct vertices of a connected undirected graph.

Definition: A graph that is not connected is the union of two or more connected subgraphs, each pair of which has no vertex in common. These disjoint connected subgraphs are called the **connected components** of the graph.

Connectivity

Example: What are the connected components in the following graph?



Solution: The connected components are the graphs with vertices {a, b, c, d}, {e}, {f}, {i, g, h, j}.

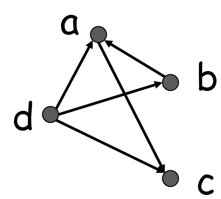
Connectivity

Definition: A directed graph is **strongly connected** if there is a path from a to b and from b to a whenever a and b are vertices in the graph.

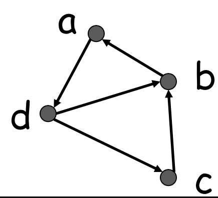
Definition: A directed graph is **weakly connected** if there is a path between any two vertices in the underlying undirected graph.

Connectivity

Example: Are the following directed graphs strongly or weakly connected?



Weakly connected, because, for example, there is no path from b to d.



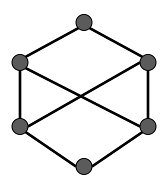
Strongly connected, because there are paths between all possible pairs of vertices.

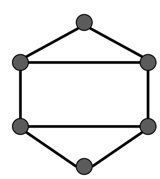
53

Connectivity

Idea: The number and size of connected components and circuits are further invariants with respect to isomorphism of simple graphs.

Example: Are these two graphs isomorphic?





Solution: No, because the right graph contains circuits of length 3, while the left graph does not.

54