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 bijection (an one-to-one and onto function) f from V_1 to 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

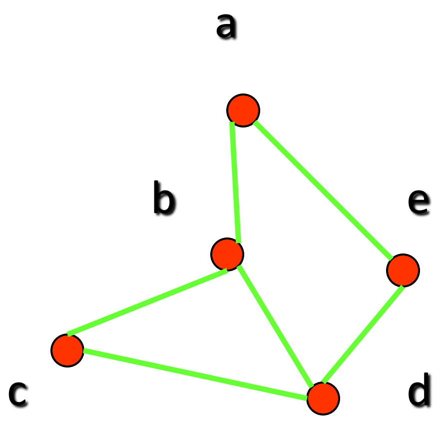
- From a visual standpoint, G_1 and G_2 are isomorphic if they can be arranged in such a way that their **displays are identical** (of course without changing adjacency).
- Unfortunately, for two simple graphs, each with n vertices, there are **n! possible isomorphisms** that we have to check in order to show that these graphs are isomorphic.
- However, showing that two graphs are **not** isomorphic can be easy.

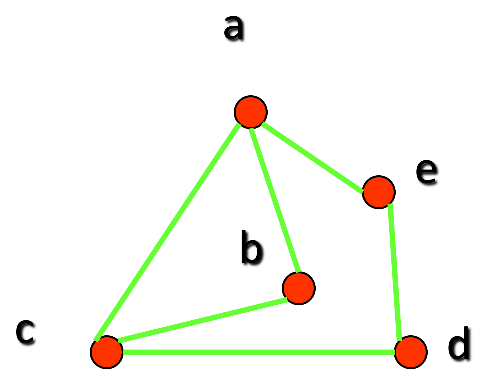
Isomorphism of Graphs

- 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?

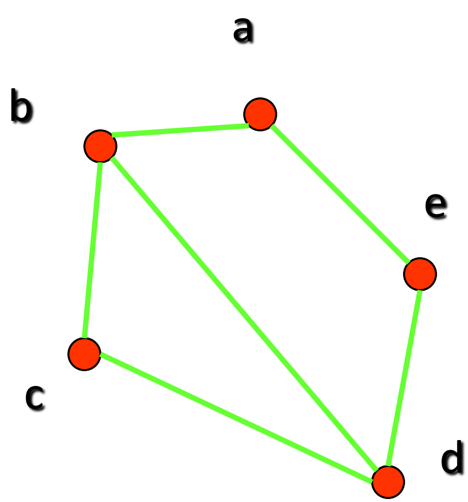


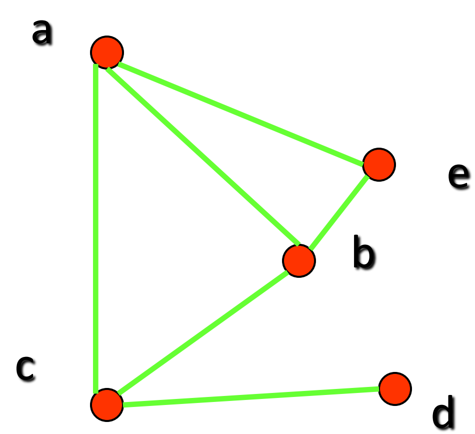


**Solution: Yes, they are isomorphic, because they can be arranged to look identical. You can see this if in the right graph you move vertex b to the left of the edge {a, c}. Then the isomorphism f from the left to the right graph is: f(a) = e, f(b) = a,
f(c) = b, f(d) = c, f(e) = d.**

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 $e_1 = \{x_0, x_1\}$, $e_2 = \{x_1, x_2\}, \ldots, e_n = \{x_{n-1}, x_n\}$, 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

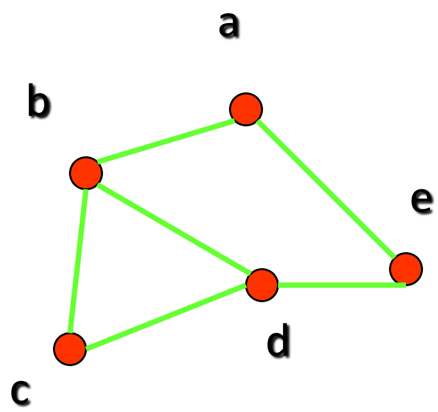
- **Definition (continued):** The path or circuit is said to **pass through** or traverse x_1, x_2, …, x_{n-1}.
- A path or circuit is **simple** if it does not contain the same edge more than once.

Connectivity

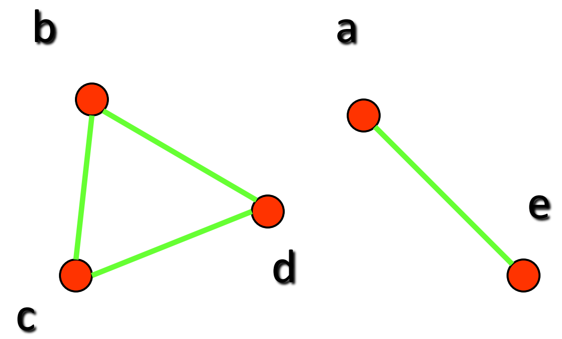
- Let us now look at something new:
- **Definition:** An undirected graph is called **connected** if there is a path between every pair of distinct vertices in the graph.
- For example, any two computers in a network can communicate if and only if the graph of this network is connected.
- **Note:** A graph consisting of only one vertex is always connected, because it does not contain any pair of distinct vertices.

Connectivity

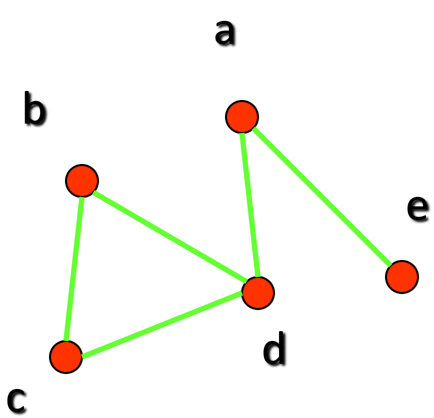
- **Example:** Are the following graphs connected?



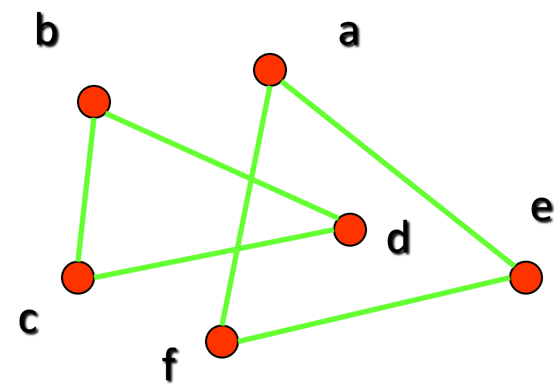
Yes.



No.



Yes.



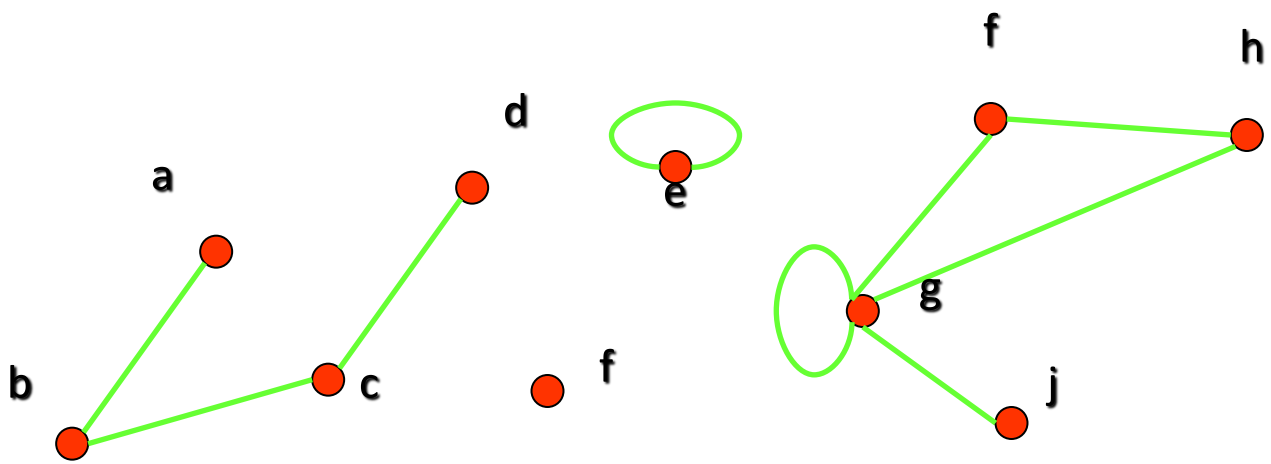
No.

Connectivity

- **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}, {f, g, h, j}.

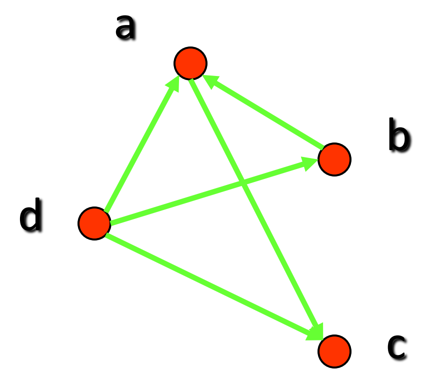
Connectivity

- **Definition:** An 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.

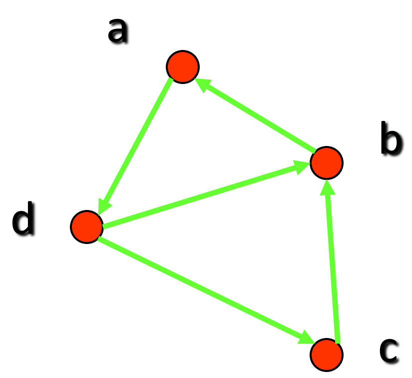
- **Definition:** An directed graph is **weakly connected** if there is a path between any two vertices in the underlying undirected graph.

Connectivity

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



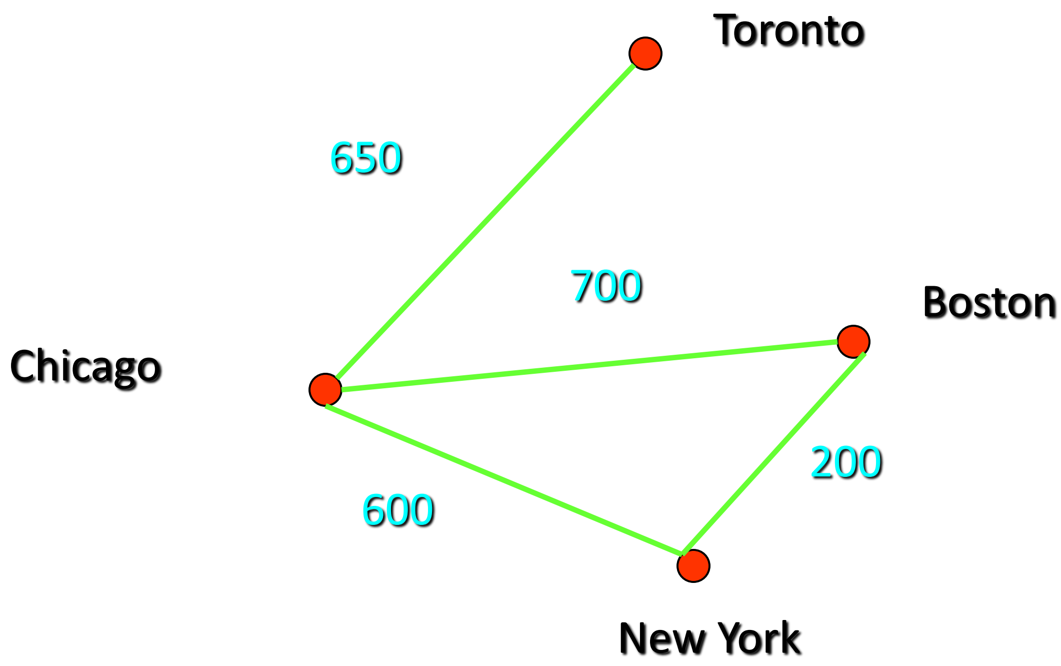
Weakly connected, because, for example, there is no path from b to d.



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

Shortest Path Problems

•We can assign weights to the edges of graphs, for example to represent the distance between cities in a railway network:



Shortest Path Problems

- Such weighted graphs can also be used to model computer networks with response times or costs as weights.
- One of the most interesting questions that we can investigate with such graphs is:
- What is the **shortest path** between two vertices in the graph, that is, the path with the **minimal sum of weights** along the way?
- This corresponds to the shortest train connection or the fastest connection in a computer network.