Graph Isomorphism

Definition

An isomorphism from a simple graph G to a simple graph H is a bijection

$$f : V(G) \to V(H)$$

such that $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$.

We write $G \cong H$ to mean that G is isomorphic to H.

Graph Isomorphism

Definition

An isomorphism from a simple graph G to a simple graph H is a bijection

$$f : V(G) \to V(H)$$

such that $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$.

We write $G \cong H$ to mean that G is isomorphic to H.

Intuitively, G and H are isomorphic if they are drawings of the "same" graph, possibly with a different labeling of the vertices.

Graph Isomorphism

Definition

An isomorphism from a simple graph G to a simple graph H is a bijection

$$f : V(G) \to V(H)$$

such that $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$.

We write $G \cong H$ to mean that G is isomorphic to H.

Intuitively, G and H are isomorphic if they are drawings of the "same" graph, possibly with a different labeling of the vertices.

Isomorphism is an equivalence relation on the set of all graphs. We think of isomorphism classes as *unlabeled graphs*.

Graph Isomorphism

Definition

An isomorphism from a simple graph G to a simple graph H is a bijection

$$f : V(G) \to V(H)$$

such that $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$.

We write $G \cong H$ to mean that G is isomorphic to H.

Intuitively, G and H are isomorphic if they are drawings of the "same" graph, possibly with a different labeling of the vertices.

Isomorphism is an equivalence relation on the set of all graphs. We think of isomorphism classes as *unlabeled graphs*.

We can show that two graphs are isomorphic by constructing a bijection as above.

Graph Isomorphism

Definition

An isomorphism from a simple graph G to a simple graph H is a bijection

$$f : V(G) \to V(H)$$

such that $uv \in E(G)$ if and only if $f(u)f(v) \in E(H)$.

We write $G \cong H$ to mean that G is isomorphic to H.

Intuitively, G and H are isomorphic if they are drawings of the "same" graph, possibly with a different labeling of the vertices.

Isomorphism is an equivalence relation on the set of all graphs. We think of isomorphism classes as *unlabeled graphs*.

We can show that two graphs are isomorphic by constructing a bijection as above.

But how do we show that two graphs are not isomorphic?

Showing that two graphs are *not* isomorphic

To show two graphs are not isomorphic, we can find a “structural property” (preserved by isomorphisms) on which they differ.

Showing that two graphs are *not* isomorphic

To show two graphs are not isomorphic, we can find a "structural property" (preserved by isomorphisms) on which they differ.

For example:

- One is bipartite and the other is not.

Showing that two graphs are *not* isomorphic

To show two graphs are not isomorphic, we can find a "structural property" (preserved by isomorphisms) on which they differ.

For example:

- One is bipartite and the other is not.
- They have distinct complements.

Showing that two graphs are *not* isomorphic

To show two graphs are not isomorphic, we can find a "structural property" (preserved by isomorphisms) on which they differ.

For example:

- One is bipartite and the other is not.
- They have distinct complements.
- They have different order or size.

Showing that two graphs are *not* isomorphic

To show two graphs are not isomorphic, we can find a "structural property" (preserved by isomorphisms) on which they differ.

For example:

- One is bipartite and the other is not.
- They have distinct complements.
- They have different order or size.
- One graph has a vertex of degree k and the other doesn't.

Showing that two graphs are *not* isomorphic

To show two graphs are not isomorphic, we can find a “structural property” (preserved by isomorphisms) on which they differ.

For example:

- One is bipartite and the other is not.
- They have distinct complements.
- They have different order or size.
- One graph has a vertex of degree k and the other doesn't.
- Their chromatic number is not the same.

Showing that two graphs are *not* isomorphic

To show two graphs are not isomorphic, we can find a "structural property" (preserved by isomorphisms) on which they differ.

For example:

- One is bipartite and the other is not.
- They have distinct complements.
- They have different order or size.
- One graph has a vertex of degree k and the other doesn't.
- Their chromatic number is not the same.
- One contains a cycle and the other does not.

Showing that two graphs are *not* isomorphic

To show two graphs are not isomorphic, we can find a "structural property" (preserved by isomorphisms) on which they differ.

For example:

- One is bipartite and the other is not.
- They have distinct complements.
- They have different order or size.
- One graph has a vertex of degree k and the other doesn't.
- Their chromatic number is not the same.
- One contains a cycle and the other does not.
- They have different girth.

Showing that two graphs are *not* isomorphic

To show two graphs are not isomorphic, we can find a "structural property" (preserved by isomorphisms) on which they differ.

For example:

- One is bipartite and the other is not.
- They have distinct complements.
- They have different order or size.
- One graph has a vertex of degree k and the other doesn't.
- Their chromatic number is not the same.
- One contains a cycle and the other does not.
- They have different girth.

Definition

The **girth** of a graph is the length of its shortest cycle.

The Petersen graph

Definition

Let P be the simple graph whose vertex set consists of all the two-element subsets of $\{1, 2, 3, 4, 5\}$. For any two vertices A and B, let AB be an edge if and only if $A \cap B = \emptyset$.

The Petersen graph

Definition

Let P be the simple graph whose vertex set consists of all the two-element subsets of $\{1, 2, 3, 4, 5\}$. For any two vertices A and B, let AB be an edge if and only if $A \cap B = \emptyset$.

Some properties of the Petersen graph:

The Petersen graph

Definition

Let P be the simple graph whose vertex set consists of all the two-element subsets of $\{1, 2, 3, 4, 5\}$. For any two vertices A and B, let AB be an edge if and only if $A \cap B = \emptyset$.

Some properties of the Petersen graph:

- Each vertex has degree 3 (we say that P is **3-regular**).

The Petersen graph

Definition

Let P be the simple graph whose vertex set consists of all the two-element subsets of $\{1, 2, 3, 4, 5\}$. For any two vertices A and B, let AB be an edge if and only if $A \cap B = \emptyset$.

Some properties of the Petersen graph:

- Each vertex has degree 3 (we say that P is **3-regular**).
- If two vertices are adjacent they do not have a common neighbor.

The Petersen graph

Definition

Let P be the simple graph whose vertex set consists of all the two-element subsets of $\{1, 2, 3, 4, 5\}$. For any two vertices A and B, let AB be an edge if and only if $A \cap B = \emptyset$.

Some properties of the Petersen graph:

- Each vertex has degree 3 (we say that P is **3-regular**).
- If two vertices are adjacent they do not have a common neighbor.
- Non-adjacent vertices have exactly one common neighbor.

The Petersen graph

Definition

Let P be the simple graph whose vertex set consists of all the two-element subsets of $\{1, 2, 3, 4, 5\}$. For any two vertices A and B, let AB be an edge if and only if $A \cap B = \emptyset$.

Some properties of the Petersen graph:

- Each vertex has degree 3 (we say that P is **3-regular**).
- If two vertices are adjacent they do not have a common neighbor.
- Non-adjacent vertices have exactly one common neighbor.
- The girth of P is 5.

Graph decompositions

Definition

A **decomposition** of a graph is a list of subgraphs such that each edge appears in exactly one subgraph in the list.

Graph decompositions

Definition

A **decomposition** of a graph is a list of subgraphs such that each edge appears in exactly one subgraph in the list.

Definition

A simple graph is **self-complementary** if it is isomorphic to its complement.

Graph decompositions

Definition

A **decomposition** of a graph is a list of subgraphs such that each edge appears in exactly one subgraph in the list.

Definition

A simple graph is **self-complementary** if it is isomorphic to its complement.

Equivalently, an n-vertex H is self-complementary if it is possible to decompose K_n into two copies of H.

1.2 Walks, paths and cycles

Definition

A **walk** is an alternating list $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ of vertices and edges such that for $1 \leq i \leq k$, the edge e_i has endpoints v_{i-1} and v_i.

An u, v-**walk** is a walk with first vertex u and last vertex v.

1.2 Walks, paths and cycles

Definition

A **walk** is an alternating list $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ of vertices and edges such that for $1 \leq i \leq k$, the edge e_i has endpoints v_{i-1} and v_i.
An **u, v-walk** is a walk with first vertex u and last vertex v.

Definition

A walk is **closed** if the first and the last vertex are the same.

1.2 Walks, paths and cycles

Definition

A **walk** is an alternating list $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ of vertices and edges such that for $1 \leq i \leq k$, the edge e_i has endpoints v_{i-1} and v_i.
An u, v-**walk** is a walk with first vertex u and last vertex v.

Definition

A walk is **closed** if the first and the last vertex are the same.

Definition

A **path** is a walk with no repeated vertex.

1.2 Walks, paths and cycles

Definition

A **walk** is an alternating list $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ of vertices and edges such that for $1 \leq i \leq k$, the edge e_i has endpoints v_{i-1} and v_i.

An **u, v-walk** is a walk with first vertex u and last vertex v.

Definition

A walk is **closed** if the first and the last vertex are the same.

Definition

A **path** is a walk with no repeated vertex.

Definition

A closed walk with no repeated vertex other than the first and the last is called a **cycle**.

Walks, paths and cycles

For simple graphs, it is not necessary to list the edges of a walk or a path; it suffices to list the vertices.

Walks, paths and cycles

For simple graphs, it is not necessary to list the edges of a walk or a path; it suffices to list the vertices.

Definition

The **length** of a walk, path, or cycle is its number of edges.

Walks, paths and cycles

For simple graphs, it is not necessary to list the edges of a walk or a path; it suffices to list the vertices.

Definition

The **length** of a walk, path, or cycle is its number of edges.

Lemma

Every u, v-walk contains a u, v-path.

Walks, paths and cycles

For simple graphs, it is not necessary to list the edges of a walk or a path; it suffices to list the vertices.

Definition

The **length** of a walk, path, or cycle is its number of edges.

Lemma

Every u, v-walk contains a u, v-path.

Exercise: prove that if there is a path from u to v and a path from v to w, then there must be a path from u to w.

Components of a graph

Definition

A graph is **connected** if for any two vertices u, v, it has a u, v-path. Otherwise, it is **disconnected**.

Components of a graph

Definition

A graph is **connected** if for any two vertices u, v, it has a u, v-path. Otherwise, it is **disconnected**.

Definition

The **components** of a graph are its maximal connected subgraphs.

Components of a graph

Definition

A graph is **connected** if for any two vertices u, v, it has a u, v-path. Otherwise, it is **disconnected**.

Definition

The **components** of a graph are its maximal connected subgraphs.

Proposition

Every graph with n vertices and k edges has at least $n - k$ components.

Components of a graph

Definition

A graph is **connected** if for any two vertices u, v, it has a u, v-path. Otherwise, it is **disconnected**.

Definition

The **components** of a graph are its maximal connected subgraphs.

Proposition

Every graph with n vertices and k edges has at least $n - k$ components.

Deleting an edge can increase the number of components by at most one. This is not the case when deleting a vertex.

Cut-edges and cut-vertices

Definition

A **cut-edge** or **cut-vertex** of a graph is an edge or vertex, respectively, whose deletion increases the number of components.

Cut-edges and cut-vertices

Definition

A cut-edge or cut-vertex of a graph is an edge or vertex, respectively, whose deletion increases the number of components.

Notation:

- $G - e$ is the graph obtained by deleting the edge e from G. Note that when we delete an edge we do not remove its endpoints.

Cut-edges and cut-vertices

Definition

A **cut-edge** or **cut-vertex** of a graph is an edge or vertex, respectively, whose deletion increases the number of components.

Notation:

- $G - e$ is the graph obtained by deleting the edge e from G. Note that when we delete an edge we do not remove its endpoints.
- $G - v$ is the graph obtained by deleting the vertex v from G. Note that when we delete a vertex we delete all the edges incident to it.

Cut-edges and cut-vertices

Definition

A **cut-edge** or **cut-vertex** of a graph is an edge or vertex, respectively, whose deletion increases the number of components.

Notation:

- $G - e$ is the graph obtained by deleting the edge e from G. Note that when we delete an edge we do not remove its endpoints.
- $G - v$ is the graph obtained by deleting the vertex v from G. Note that when we delete a vertex we delete all the edges incident to it.
- For $M \subseteq E(G)$, $G - M$ is the graph obtained by deleting the edges in the set M from G.

Cut-edges and cut-vertices

Definition

A cut-edge or cut-vertex of a graph is an edge or vertex, respectively, whose deletion increases the number of components.

Notation:

- $G - e$ is the graph obtained by deleting the edge e from G. Note that when we delete an edge we do not remove its endpoints.
- $G - v$ is the graph obtained by deleting the vertex v from G. Note that when we delete a vertex we delete all the edges incident to it.
- For $M \subseteq E(G)$, $G - M$ is the graph obtained by deleting the edges in the set M from G.
- For $S \subseteq V(G)$, $G - S$ is the graph obtained by deleting the vertices in the set S from G.