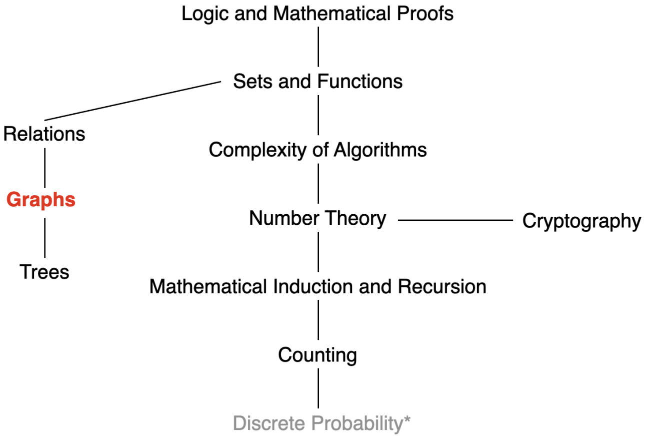
Discrete Mathematics for Computer Science

Lecture 18: Graph

Dr. Ming Tang

Department of Computer Science and Engineering
Southern University of Science and Technology (SUSTech)
Email: tangm3@sustech.edu.cn

This Lecture



Graph and terminologies, representing graphs and graph isomorphism, connectivity, Euler and Hamiliton path, ...

Bipartite Graphs

Definition: A simple graph G is bipartite if V can be partitioned into two disjoint subsets V_1 and V_2 such that every edge connects a vertex in V_1 and a vertex in V_2.

Bipartite Graphs

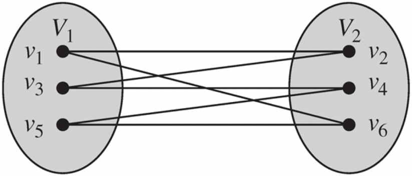
Definition: A simple graph G is bipartite if V can be partitioned into two disjoint subsets V_1 and V_2 such that every edge connects a vertex in V_1 and a vertex in V_2.

An equivalent definition of a bipartite graph is a graph where it is possible to color the vertices red or blue so that no two adjacent vertices are of the same color.

Bipartite Graphs

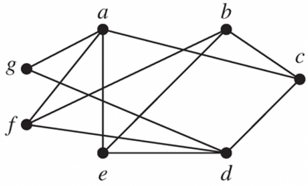
Definition: A simple graph G is bipartite if V can be partitioned into two disjoint subsets V_1 and V_2 such that every edge connects a vertex in V_1 and a vertex in V_2.

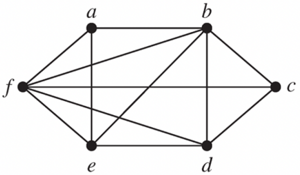
An equivalent definition of a bipartite graph is a graph where it is possible to color the vertices red or blue so that no two adjacent vertices are of the same color.



Bipartite Graphs

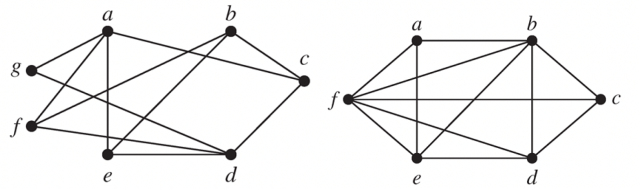
Are these graphs bipartite?





Bipartite Graphs

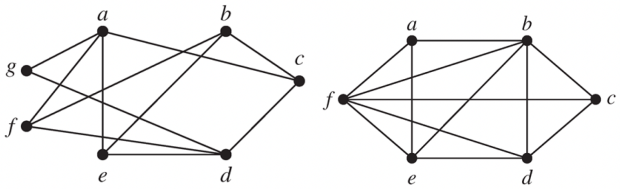
Are these graphs bipartite?



(a) Bipartite: Its vertex set is the union of two disjoint sets, $\{a, b, d\}$ and $\{c, e, f, g\}$, and each edge connects a vertex in one of these subsets to a vertex in the other subset.

Bipartite Graphs

Are these graphs bipartite?

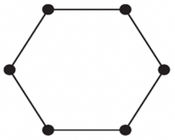


(a) Bipartite: Its vertex set is the union of two disjoint sets, $\{a, b, d\}$ and $\{c, e, f, g\}$, and each edge connects a vertex in one of these subsets to a vertex in the other subset.

(b) Not bipartite: Its vertex set cannot be partitioned into two subsets so that edges do not connect two vertices from the same subset.

Bipartite Graphs: Examples

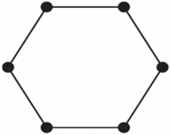
Show that C_6 is bipartite.



C_6

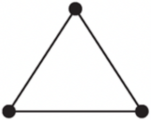
Bipartite Graphs: Examples

Show that C_6 is bipartite.



C_6

Show that C_3 is not bipartite.



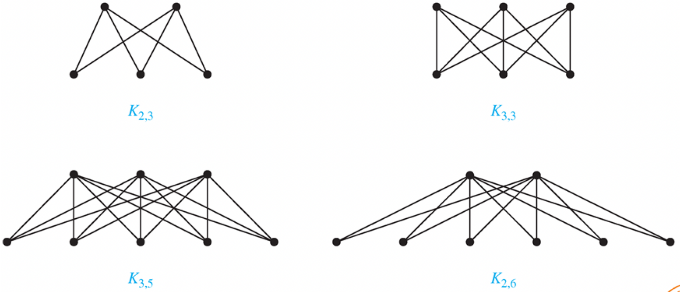
C_3

Complete Bipartite Graphs

Definition: A complete bipartite graph $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets V_1 of size m and V_2 of size n such that there is an edge from every vertex in V_1 to every vertex in V_2.

Complete Bipartite Graphs

Definition: A complete bipartite graph $K_{m,n}$ is a graph that has its vertex set partitioned into two subsets V_1 of size m and V_2 of size n such that there is an edge from every vertex in V_1 to every vertex in V_2.



Bipartite Graphs and Matchings

Given a bipartite graph, a matching is a subset of edges E such that no two edges are incident with the same vertex.

Bipartite Graphs and Matchings

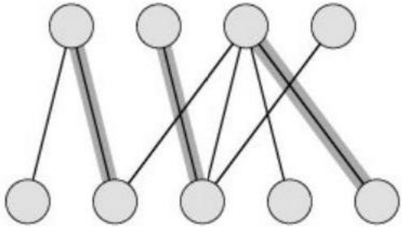
Given a bipartite graph, a matching is a subset of edges E such that no two edges are incident with the same vertex.

In other words, a matching is a subset of edges such that if $\{s, t\}$ and $\{u, v\}$ are distinct edges of the matching, then s, t, u, and v are distinct.

Bipartite Graphs and Matchings

Given a bipartite graph, a matching is a subset of edges E such that no two edges are incident with the same vertex.

In other words, a matching is a subset of edges such that if $\{s, t\}$ and $\{u, v\}$ are distinct edges of the matching, then s, t, u, and v are distinct.



Bipartite Graphs and Matchings

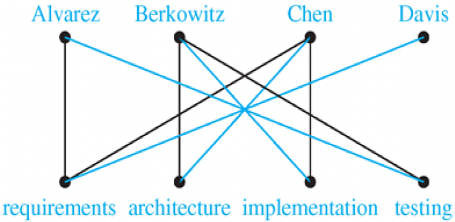
Given a bipartite graph, a matching is a subset of edges E such that no two edges are incident with the same vertex.

Job assignments: vertices represent the jobs and the employees, edges link employees with those jobs they have been trained to do. A common goal is to match jobs to employees so that the most jobs are done.

Bipartite Graphs and Matchings

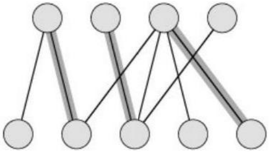
Given a bipartite graph, a matching is a subset of edges E such that no two edges are incident with the same vertex.

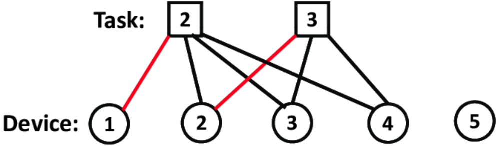
Job assignments: vertices represent the jobs and the employees, edges link employees with those jobs they have been trained to do. A common goal is to match jobs to employees so that the most jobs are done.



Bipartite Graphs and Matchings

A maximum matching is a matching with the largest number of edges.

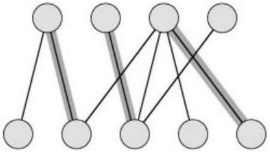


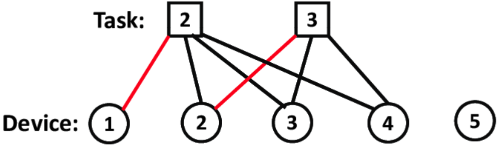


Bipartite Graphs and Matchings

A maximum matching is a matching with the largest number of edges.

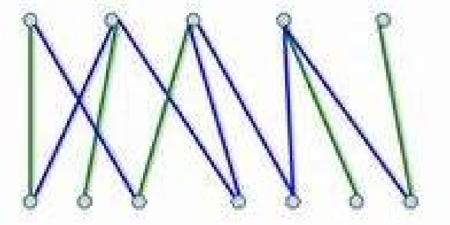
A matching M in a bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) is a complete matching from V_1 to V_2 if every vertex in V_1 is the endpoint of an edge in the matching, or equivalently, if $|M| = |V_1|$.

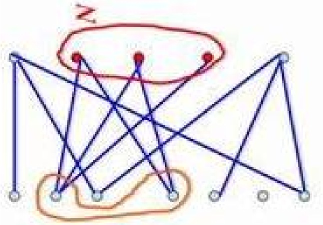




Hall's Theorem: Example

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.





Proof of Hall's Theorem

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.

Proof: "only if"

Proof of Hall's Theorem

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.

Proof: "only if"

Suppose that there is a complete matching M from V_1 to V_2. Consider an arbitrary subset $A \subseteq V_1$.

Proof of Hall's Theorem

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.

Proof: "only if"

Suppose that there is a complete matching M from V_1 to V_2. Consider an arbitrary subset $A \subseteq V_1$.

Then, for every vertex $v \in A$, there is an edge in M connecting v to a vertex in V_2.

Proof of Hall's Theorem

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.

Proof: "only if"

Suppose that there is a complete matching M from V_1 to V_2. Consider an arbitrary subset $A \subseteq V_1$.

Then, for every vertex $v \in A$, there is an edge in M connecting v to a vertex in V_2.

Thus, there are at least as many vertices in V_2 that are neighbors of vertices in V_1 as there are vertices in V_1.

Proof of Hall's Theorem

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.

Proof: "only if"

Suppose that there is a complete matching M from V_1 to V_2. Consider an arbitrary subset $A \subseteq V_1$.

Then, for every vertex $v \in A$, there is an edge in M connecting v to a vertex in V_2.

Thus, there are at least as many vertices in V_2 that are neighbors of vertices in V_1 as there are vertices in V_1.

Hence, $|N(A)| \geq |A|$.

Proof of Hall's Theorem

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.

Proof: "if",

Proof of Hall's Theorem

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.

Proof: "if", use strong induction to prove it.

Proof of Hall's Theorem

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.

Proof: "if", use strong induction to prove it.

Basic Step: $|V_1| = 1$

Proof of Hall's Theorem

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.

Proof: "if", use strong induction to prove it.

Basic Step: $|V_1| = 1$

Inductive hypothesis: Let k be a positive integer. If $G = (V, E)$ is a bipartite graph with bipartition (V_1, V_2), and $|V_1| = j \leq k$, then there is a complete mathching M from V_1 to V_2 whenever the condition that $|N(A)| \geq |A|$ for all $A \subseteq V_1$ is met.

Proof of Hall's Theorem

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.

Proof: "if", use strong induction to prove it.

Basic Step: $|V_1| = 1$

Inductive hypothesis: Let k be a positive integer. If $G = (V, E)$ is a bipartite graph with bipartition (V_1, V_2), and $|V_1| = j \leq k$, then there is a complete mathching M from V_1 to V_2 whenever the condition that $|N(A)| \geq |A|$ for all $A \subseteq V_1$ is met.

Inductive step: Suppose that $H = (W, F)$ is a bipartite graph with bipartition (W_1, W_2) and $|W_1| = k + 1$.

Proof of Hall's Theorem

Inductive hypothesis: Let $|V_1| = j \leq k$. There is a complete mathching M from V_1 to V_2 whenever $|N(A)| \geq |A|$ for all $A \subseteq V_1$.

Inductive step: Suppose that $H = (W, F)$ is a bipartite graph with bipartition (W_1, W_2) and $|W_1| = k + 1$.

Suppose $|N(A)| \geq |A|$ for all $A \subseteq W_1$. Prove there exists a complete matching. There are two cases:

Proof of Hall's Theorem

Inductive hypothesis: Let $|V_1| = j \leq k$. There is a complete mathching M from V_1 to V_2 whenever $|N(A)| \geq |A|$ for all $A \subseteq V_1$.

Inductive step: Suppose that $H = (W, F)$ is a bipartite graph with bipartition (W_1, W_2) and $|W_1| = k + 1$.

Suppose $|N(A)| \geq |A|$ for all $A \subseteq W_1$. Prove there exists a complete matching. There are two cases:

(i) For all integers j with $1 \leq j \leq k$, the vertices in every set of j elements from W_1 are adjacent to at least $j + 1$ elements of W_2.

Proof of Hall's Theorem

Inductive hypothesis: Let $|V_1| = j \leq k$. There is a complete mathching M from V_1 to V_2 whenever $|N(A)| \geq |A|$ for all $A \subseteq V_1$.

Inductive step: Suppose that $H = (W, F)$ is a bipartite graph with bipartition (W_1, W_2) and $|W_1| = k + 1$.

Suppose $|N(A)| \geq |A|$ for all $A \subseteq W_1$. Prove there exists a complete matching. There are two cases:

(i) For all integers j with $1 \leq j \leq k$, the vertices in every set of j elements from W_1 are adjacent to at least $j + 1$ elements of W_2.

(ii) For some integer j with $1 \leq j \leq k$, there is a subset W_1' of j vertices such that there are exactly j neighbors of these vertices in W_2.

Proof of Hall's Theorem

Inductive hypothesis: Let $|V_1| = j \leq k$. There is a complete mathching M from V_1 to V_2 whenever $|N(A)| \geq |A|$ for all $A \subseteq V_1$.

Inductive step: (i) For all integers j with $1 \leq j \leq k$, the vertices in every set of j elements from W_1 are adjacent to at least $j+1$ elements of W_2.

Proof of Hall's Theorem

Inductive hypothesis: Let $|V_1| = j \leq k$. There is a complete mathching M from V_1 to V_2 whenever $|N(A)| \geq |A|$ for all $A \subseteq V_1$.

Inductive step: (i) For all integers j with $1 \leq j \leq k$, the vertices in every set of j elements from W_1 are adjacent to at least $j+1$ elements of W_2.

- Let A be such a subset of W_1 with j elements, where $1 \leq j \leq k$;
- $|N(A)| \geq |A| + 1$ for all A.

Proof of Hall's Theorem

Inductive hypothesis: Let $|V_1| = j \leq k$. There is a complete mathching M from V_1 to V_2 whenever $|N(A)| \geq |A|$ for all $A \subseteq V_1$.

Inductive step: (i) For all integers j with $1 \leq j \leq k$, the vertices in every set of j elements from W_1 are adjacent to at least $j+1$ elements of W_2.

- Let A be such a subset of W_1 with j elements, where $1 \leq j \leq k$;
- $|N(A)| \geq |A| + 1$ for all A.

We select a vertex $v \in W_1$ and an element $w \in N(\{v\})$. The inductive hypothesis tells us there is a complete matching from $W_1 - \{v\}$ to $W_2 - \{w\}$.

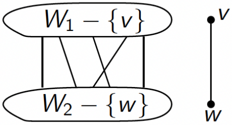
Proof of Hall's Theorem

Inductive hypothesis: Let $|V_1| = j \leq k$. There is a complete mathching M from V_1 to V_2 whenever $|N(A)| \geq |A|$ for all $A \subseteq V_1$.

Inductive step: (i) For all integers j with $1 \leq j \leq k$, the vertices in every set of j elements from W_1 are adjacent to at least $j + 1$ elements of W_2.

- Let A be such a subset of W_1 with j elements, where $1 \leq j \leq k$;
- $|N(A)| \geq |A| + 1$ for all A.

We select a vertex $v \in W_1$ and an element $w \in N(\{v\})$. The inductive hypothesis tells us there is a complete matching from $W_1 - \{v\}$ to $W_2 - \{w\}$.



Proof of Hall's Theorem

Inductive step: (ii) For some integer j with $1 \leq j \leq k$, there is a subset W_1' of j vertices such that they have exactly j neighbors in W_2.

Proof of Hall's Theorem

Inductive step: (ii) For some integer j with $1 \leq j \leq k$, there is a subset W_1' of j vertices such that they have exactly j neighbors in W_2.

- Let A be such a subset of W_1 with j elements, where $1 \leq j \leq k$;
- $|N(A)| = |A|$ for some A, i.e., W_1'.

Let W_2' be the set of these neighbors. Then by i.h., there is a complete matching from W_1' to W_2'.

Proof of Hall's Theorem

Inductive step: (ii) For some integer j with $1 \leq j \leq k$, there is a subset W_1' of j vertices such that they have exactly j neighbors in W_2.

- Let A be such a subset of W_1 with j elements, where $1 \leq j \leq k$;
- $|N(A)| = |A|$ for some A, i.e., W_1'.

Let W_2' be the set of these neighbors. Then by i.h., there is a complete matching from W_1' to W_2'.

Now consider $K = (W_1 - W_1', W_2 - W_2')$.

Proof of Hall's Theorem

Inductive step: (ii) For some integer j with $1 \leq j \leq k$, there is a subset W_1' of j vertices such that they have exactly j neighbors in W_2.

- Let A be such a subset of W_1 with j elements, where $1 \leq j \leq k$;
- $|N(A)| = |A|$ for some A, i.e., W_1'.

Let W_2' be the set of these neighbors. Then by i.h., there is a complete matching from W_1' to W_2'.

Now consider $K = (W_1 - W_1', W_2 - W_2')$. We will show that the condition $|N(A)| \geq |A|$ is met for all subsets A of $W_1 - W_1'$.

Proof of Hall's Theorem

Inductive step: (ii) For some integer j with $1 \leq j \leq k$, there is a subset W_1' of j vertices such that they have exactly j neighbors in W_2.

- Let A be such a subset of W_1 with j elements, where $1 \leq j \leq k$;
- $|N(A)| = |A|$ for some A, i.e., W_1'.

Let W_2' be the set of these neighbors. Then by i.h., there is a complete matching from W_1' to W_2'.

Now consider $K = (W_1 - W_1', W_2 - W_2')$. We will show that the condition $|N(A)| \geq |A|$ is met for all subsets A of $W_1 - W_1'$. If not,

- There is a subset B of t vertices with $1 \leq t \leq k + 1 - j$ such that $|N(B)| < t$

Proof of Hall's Theorem

Inductive step: (ii) For some integer j with $1 \leq j \leq k$, there is a subset W_1' of j vertices such that they have exactly j neighbors in W_2.

- Let A be such a subset of W_1 with j elements, where $1 \leq j \leq k$;
- $|N(A)| = |A|$ for some A, i.e., W_1'.

Let W_2' be the set of these neighbors. Then by i.h., there is a complete matching from W_1' to W_2'.

Now consider $K = (W_1 - W_1', W_2 - W_2')$. We will show that the condition $|N(A)| \geq |A|$ is met for all subsets A of $W_1 - W_1'$. If not,

- There is a subset B of t vertices with $1 \leq t \leq k + 1 - j$ such that $|N(B)| < t$
- Adding those deleted j vertices, $|N(B)| + j < t + j$. Contradiction.

Proof of Hall's Theorem

Inductive step: (ii) For some integer j with $1 \leq j \leq k$, there is a subset W_1' of j vertices such that they have exactly j neighbors in W_2.

- Let A be such a subset of W_1 with j elements, where $1 \leq j \leq k$;
- $|N(A)| = |A|$ for some A, i.e., W_1'.

Let W_2' be the set of these neighbors. Then by i.h., there is a complete matching from W_1' to W_2'.

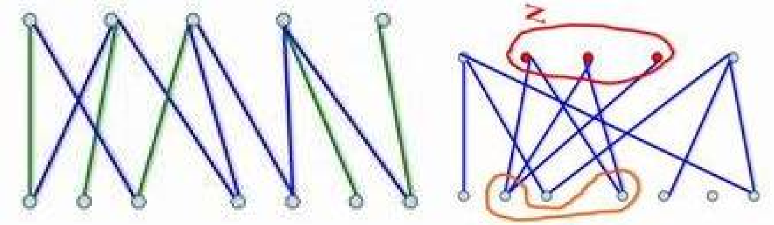
Now consider $K = (W_1 - W_1', W_2 - W_2')$. We will show that the condition $|N(A)| \geq |A|$ is met for all subsets A of $W_1 - W_1'$. If not,

- There is a subset B of t vertices with $1 \leq t \leq k + 1 - j$ such that $|N(B)| < t$
- Adding those deleted j vertices, $|N(B)| + j < t + j$. Contradiction.

Thus, there is a complete matching from from $W_1 - W_1'$ to $W_2 - W_2'$.

Hall's Theorem: Example

Theorem (Hall's Marriage Theorem): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.



Subgraphs

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

Subgraphs

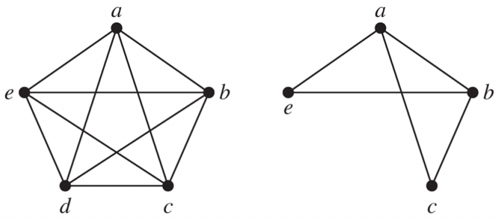
Definition: A subgraph of a graph $G = (V, E)$ is a graph (W, F), where $W \subseteq V$ and $F \subseteq E$.

A subgraph H of G is a proper subgraph of G if $H \neq G$.

Subgraphs

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

A subgraph H of G is a proper subgraph of G if $H \neq G$.

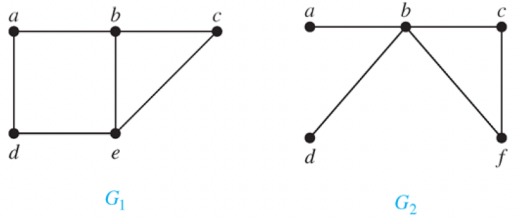


Union of Graphs

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 $E_1 \cup E_2$, denoted by $G_1 \cup G_2$.

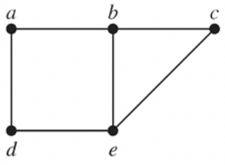
Union of Graphs

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 $E_1 \cup E_2$, denoted by $G_1 \cup G_2$.

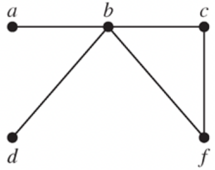


Union of Graphs

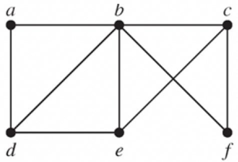
Definition: The union of two simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the simple graph with vertex set $V_1 \cup V_2$ and edge set $E_1 \cup E_2$, denoted by $G_1 \cup G_2$.



G_1

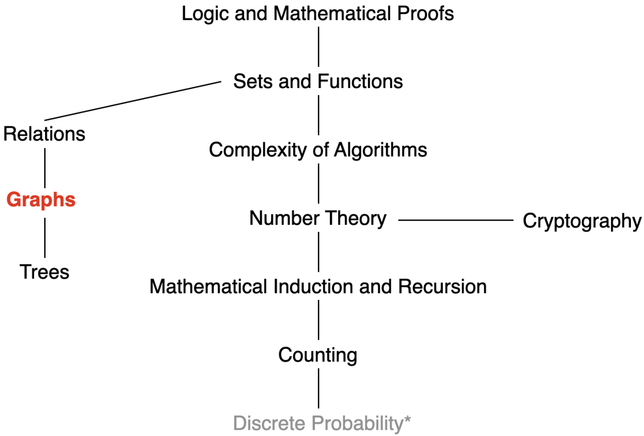


G_2



$G_1 \cup G_2$

This Lecture



Graph and terminologies, representing graphs and graph isomorphism, connectivity, Euler and Hamiliton path, ...

Representation of Graphs

To represent a graph, we may use adjacency lists, adjacency matrices, and incidence matrices.

Representation of Graphs

To represent a graph, we may use adjacency lists, adjacency matrices, and incidence matrices.

Definition: An adjacency list can be used to represent a graph with no multiple edges by specifying the vertices that are adjacent to each vertex of the graph.

Representation of Graphs

To represent a graph, we may use adjacency lists, adjacency matrices, and incidence matrices.

Definition: An adjacency list can be used to represent a graph with no multiple edges by specifying the vertices that are adjacent to each vertex of the graph.

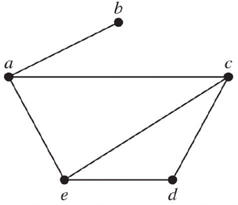


TABLE 1 An Adjacency List for a Simple Graph.

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

Representation of Graphs

Definition: An adjacency list can be used to represent a graph with no multiple edges by specifying the vertices that are adjacent to each vertex of the graph.

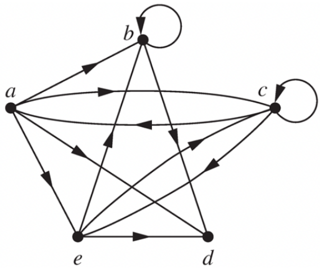


TABLE 2 An Adjacency List for a Directed Graph.

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

Adjacency Matrices

Definition: Suppose that $G = (V, E)$ is a simple graph with $|V| = n$. Arbitrarily list the vertices of G as $v_1, v_2, \ldots, v_n$. The adjacency matrix $\mathbf{A}_G$ of G, is the $n \times n$ zero-one matrix with 1 as its (i, j)-th entry when v_i and v_j are adjacent, and 0 as its (i, j)-th entry when they are not adjacent.

Adjacency Matrices

Definition: Suppose that $G = (V, E)$ is a simple graph with $|V| = n$. Arbitrarily list the vertices of G as $v_1, v_2, \ldots, v_n$. The adjacency matrix $\mathbf{A}_G$ of G, is the $n \times n$ zero-one matrix with 1 as its (i, j)-th entry when v_i and v_j are adjacent, and 0 as its (i, j)-th entry when they are not adjacent.

$$\mathbf{A}_G = [a_{ij}]_{n\times n}, \text{ where}$$

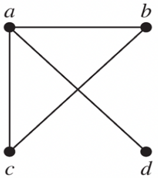
$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \text{ is an edge of } G, \\ 0 & \text{otherwise.} \end{cases}$$

Adjacency Matrices

Definition: Suppose that $G = (V, E)$ is a simple graph with $|V| = n$. Arbitrarily list the vertices of G as $v_1, v_2, \ldots, v_n$. The adjacency matrix $\mathbf{A}_G$ of G, is the $n \times n$ zero-one matrix with 1 as its (i, j)-th entry when v_i and v_j are adjacent, and 0 as its (i, j)-th entry when they are not adjacent.

$$\mathbf{A}_G = [a_{ij}]_{n\times n}, \text{ where}$$

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

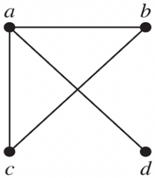


Adjacency Matrices

Definition: Suppose that $G = (V, E)$ is a simple graph with $|V| = n$. Arbitrarily list the vertices of G as $v_1, v_2, \ldots, v_n$. The adjacency matrix $\mathbf{A}_G$ of G, is the $n \times n$ zero-one matrix with 1 as its (i, j)-th entry when v_i and v_j are adjacent, and 0 as its (i, j)-th entry when they are not adjacent.

$$\mathbf{A}_G = [a_{ij}]_{n\times n}, \text{ where}$$

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



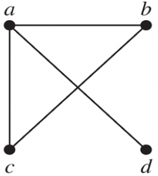
$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Adjacency Matrices

Definition: Suppose that $G = (V, E)$ is a simple graph with $|V| = n$. Arbitrarily list the vertices of G as $v_1, v_2, \ldots, v_n$. The adjacency matrix $\mathbf{A}_G$ of G, is the $n \times n$ zero-one matrix with 1 as its (i, j)-th entry when v_i and v_j are adjacent, and 0 as its (i, j)-th entry when they are not adjacent.

$$\mathbf{A}_G = [a_{ij}]_{n\times n}, \text{ where}$$

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



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

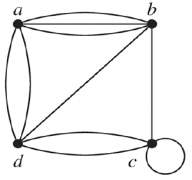
Directed graph?

Adjacency Matrices

Adjacency matrices can also be used to represent graphs with loops and multiple edges. The matrix is no longer a zero-one matrix.

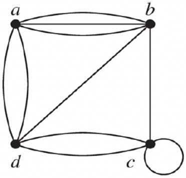
Adjacency Matrices

Adjacency matrices can also be used to represent graphs with loops and multiple edges. The matrix is no longer a zero-one matrix.



Adjacency Matrices

Adjacency matrices can also be used to represent graphs with loops and multiple edges. The matrix is no longer a zero-one matrix.



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

Incidence Matrices

Definition: Let $G = (V, E)$ be an undirected graph with vertices v_1, v_2, . . . , v_n and edges e_1, e_2, . . . , e_m. The incidence matrix with respect to the ordering of V and E is the $n \times m$ matrix $\mathbf{M} = [m_{ij}]_{n \times m}$, where

Incidence Matrices

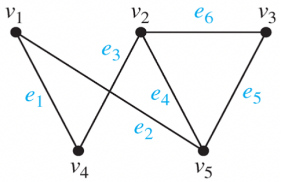
Definition: Let $G = (V, E)$ be an undirected graph with vertices v_1, v_2, . . . , v_n and edges e_1, e_2, ... , e_m. The incidence matrix with respect to the ordering of V and E is the $n \times m$ matrix $\mathbf{M} = [m_{ij}]_{n\times m}$, where

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

Incidence Matrices

Definition: Let $G = (V, E)$ be an undirected graph with vertices $v_1, v_2, \ldots, v_n$ and edges $e_1, e_2, \ldots, e_m$. The incidence matrix with respect to the ordering of V and E is the $n \times m$ matrix $\mathbf{M} = [m_{ij}]_{n \times m}$, where

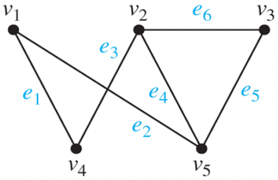
$$m_{ij} = \begin{cases} 1 & \text{if edge } e_j \text{ is incident with } v_i, \\ 0 & \text{otherwise.} \end{cases}$$



Incidence Matrices

Definition: Let $G = (V, E)$ be an undirected graph with vertices v_1, v_2, . . . , v_n and edges e_1, e_2, . . . , e_m. The incidence matrix with respect to the ordering of V and E is the $n \times m$ matrix $\mathbf{M} = [m_{ij}]_{n \times m}$, where

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

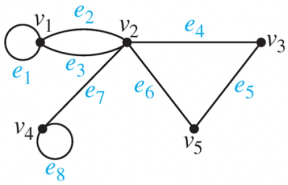


$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Incidence Matrices

Definition: Let $G = (V, E)$ be an undirected graph with vertices v_1, v_2, . . . , v_n and edges e_1, e_2, . . . , e_m. The incidence matrix with respect to the ordering of V and E is the $n \times m$ matrix $\mathbf{M} = [m_{ij}]_{n \times m}$, where

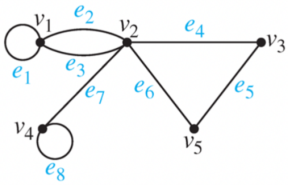
$$m_{ij} = \begin{cases} 1 & \text{if edge } e_j \text{ is incident with } v_i, \\ 0 & \text{otherwise.} \end{cases}$$



Incidence Matrices

Definition: Let $G = (V, E)$ be an undirected graph with vertices $v_1, v_2, \ldots, v_n$ and edges $e_1, e_2, \ldots, e_m$. The incidence matrix with respect to the ordering of V and E is the $n \times m$ matrix $\mathbf{M} = [m_{ij}]_{n\times m}$, where

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



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

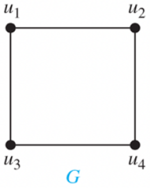
Isomorphism of Graphs

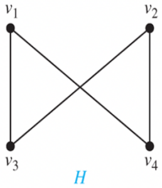
Definition: The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there is a one-to-one and onto function from V_1 to V_2 with the property that a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2, for all a and b in V_1. Such a function is called an isomorphism.

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 one-to-one and onto function 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 is called an isomorphism.

Are the two graphs isomorphic?

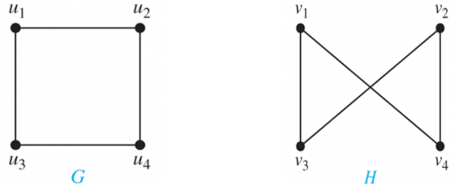




Isomorphism of Graphs

Definition: The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there is a one-to-one and onto function 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 is called an isomorphism.

Are the two graphs isomorphic?



- Define a one-to-one correspondence: $f(u_1) = v_1$, $f(u_2) = v_4$, $f(u_3) = v_3$, and $f(u_4) = v_2$.
- Check their adjacent matrics.

Isomorphism of Graphs

It is usually difficult to determine whether two simple graphs are isomorphic using brute force since there are $n!$ possible one-to-one correspondences.

Isomorphism of Graphs

It is usually difficult to determine whether two simple graphs are isomorphic using brute force since there are $n!$ possible one-to-one correspondences.

Sometimes it is not difficult to show that two graphs are not isomorphic. We can achieve this by checking some graph invariants.

Isomorphism of Graphs

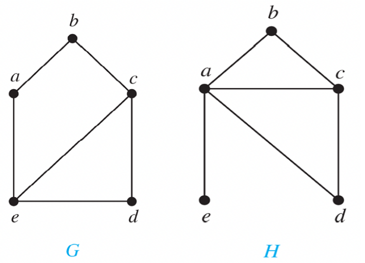
It is usually difficult to determine whether two simple graphs are isomorphic using brute force since there are $n!$ possible one-to-one correspondences.

Sometimes it is not difficult to show that two graphs are not isomorphic. We can achieve this by checking some graph invariants.

Useful graph invariants include the number of vertices, number of edges, degree sequence, etc.

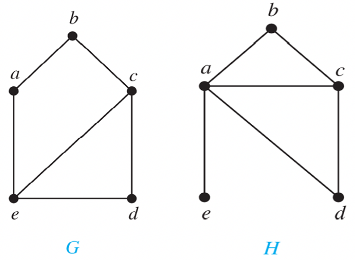
Isomorphism of Graphs: Example

Determine whether these two graphs are isomorphic.



Isomorphism of Graphs: Example

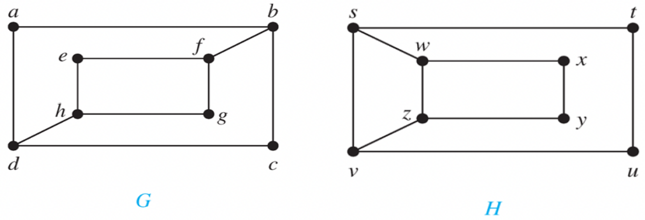
Determine whether these two graphs are isomorphic.



H has a vertex of degree one, namely, e, whereas G has no vertices of degree one. It follows that G and H are not isomorphic.

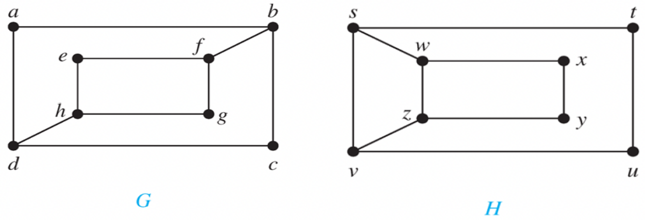
Isomorphism of Graphs: Example

Determine whether these two graphs are isomorphic.



Isomorphism of Graphs: Example

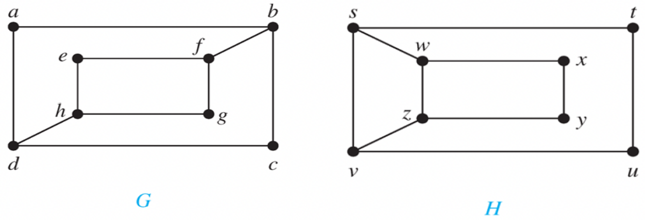
Determine whether these two graphs are isomorphic.



G and H are not isomorphic. This is because $deg(a) = 2$ in G, and a must correspond to either t, u, x, or y in H.

Isomorphism of Graphs: Example

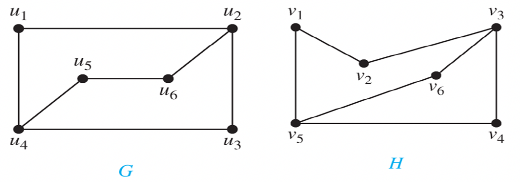
Determine whether these two graphs are isomorphic.



G and H are not isomorphic. This is because $deg(a) = 2$ in G, and a must correspond to either t, u, x, or y in H. However, each of these four vertices in H is adjacent to another vertex of degree two in H, which is not true for a in G.

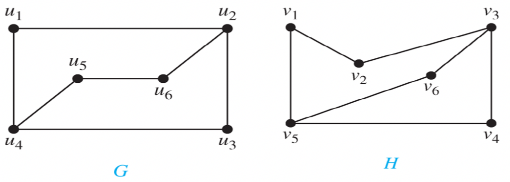
Isomorphism of Graphs: Example

Determine whether these two graphs are isomorphic.



Isomorphism of Graphs: Example

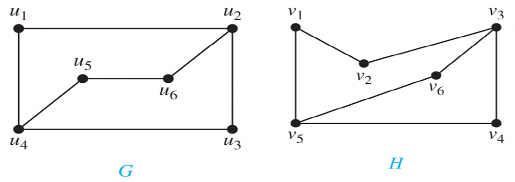
Determine whether these two graphs are isomorphic.



Because many isomorphic invariants (e.g., number of vertices/edges, degree) agree, G and H may be isomorphic. We now will define a function f:

Isomorphism of Graphs: Example

Determine whether these two graphs are isomorphic.

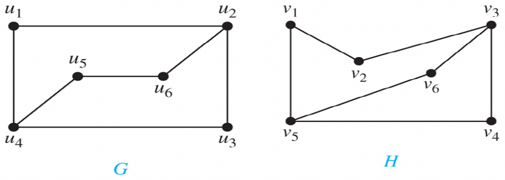


Because many isomorphic invariants (e.g., number of vertices/edges, degree) agree, G and H may be isomorphic. We now will define a function f:

- $f(u_1)$ can be either v_4 or v_6, because u_1 is not adjacent to any other vertex of degree two. We arbitrarily set $f(u_1) = v_6$.

Isomorphism of Graphs: Example

Determine whether these two graphs are isomorphic.

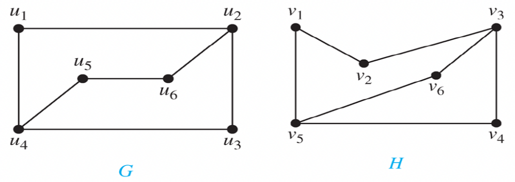


Because many isomorphic invariants (e.g., number of vertices/edges, degree) agree, G and H may be isomorphic. We now will define a function f:

- $f(u_1)$ can be either v_4 or v_6, because u_1 is not adjacent to any other vertex of degree two. We arbitrarily set $f(u_1) = v_6$.
- u_2 is adjacent to u_1, so $f(u_2)$ can be either v_3 or v_5. We arbitrarily set $f(u_2) = v_3$.

Isomorphism of Graphs: Example

Determine whether these two graphs are isomorphic.

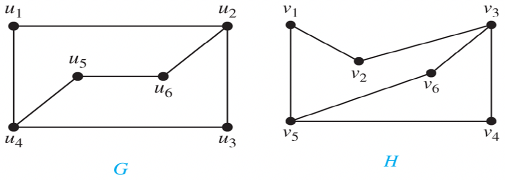


Because many isomorphic invariants (e.g., number of vertices/edges, degree) agree, G and H may be isomorphic. We now will define a function f:

- $f(u_1)$ can be either v_4 or v_6, because u_1 is not adjacent to any other vertex of degree two. We arbitrarily set $f(u_1) = v_6$.
- u_2 is adjacent to u_1, so $f(u_2)$ can be either v_3 or v_5. We arbitrarily set $f(u_2) = v_3$.
- ...

Isomorphism of Graphs: Example

Determine whether these two graphs are isomorphic.

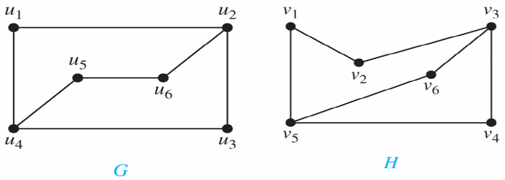


Because many isomorphic invariants (e.g., number of vertices/edges, degree) agree, G and H may be isomorphic. We now will define a function f:

- $f(u_1)$ can be either v_4 or v_6, because u_1 is not adjacent to any other vertex of degree two. We arbitrarily set $f(u_1) = v_6$.
- u_2 is adjacent to u_1, so $f(u_2)$ can be either v_3 or v_5. We arbitrarily set $f(u_2) = v_3$.
- ...
- $f(u_3) = v_4$, $f(u_4) = v_5$, $f(u_5) = v_1$, and $f(u_6) = v_2$.

Isomorphism of Graphs: Example

Determine whether these two graphs are isomorphic.

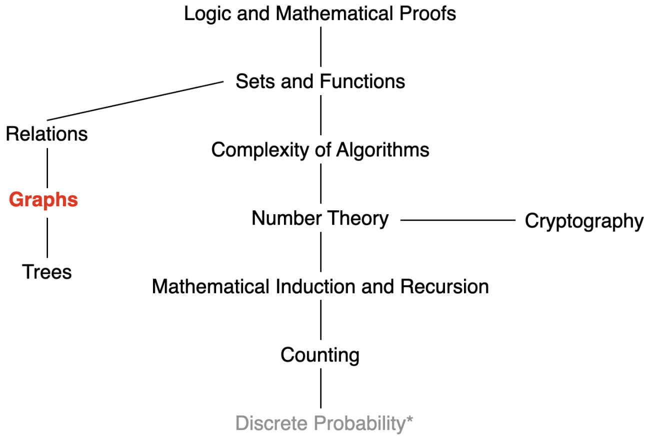


$f(u_1) = v_6,\ f(u_2) = v_3,\ f(u_3) = v_4,\ f(u_4) = v_5,\ f(u_5) = v_1,\ f(u_6) = v_2.$

$$\mathbf{A}_H = \begin{array}{cc} & \begin{array}{cccccc} v_6 & v_3 & v_4 & v_5 & v_1 & v_2 \end{array} \\ \begin{array}{c} v_6 \\ v_3 \\ v_4 \\ v_5 \\ v_1 \\ v_2 \end{array} & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}. \quad \mathbf{A}_G = \begin{array}{cc} & \begin{array}{cccccc} u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \end{array} \\ \begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{array} & \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array},$$

We conclude that f is an isomorphism, so G and H are isomorphic.

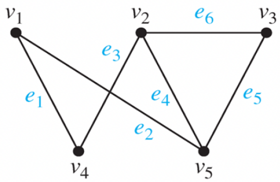
This Lecture



Graph and terminologies, representing graphs and graph isomorphism, connectivity, Euler and Hamiliton path, ...

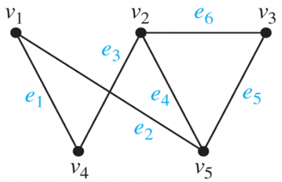
Path: Undirected Graph

Definition: Let n be a nonnegative integer and G an undirected graph. A path of length n from u to v in G is a sequence of n edges $e_1, e_2, \ldots, e_n$ of G for which there exists a sequence $x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$ of vertices such that e_i has the endpoints x_{i-1} and x_i for $i = 1, \ldots, n$.



Path: Undirected Graph

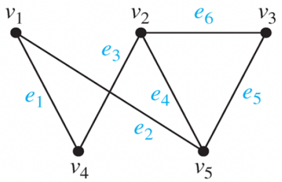
Definition: Let n be a nonnegative integer and G an undirected graph. A path of length n from u to v in G is a sequence of n edges $e_1, e_2, \ldots, e_n$ of G for which there exists a sequence $x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$ of vertices such that e_i has the endpoints x_{i-1} and x_i for $i = 1, \ldots, n$.



The path is a circuit if it begins and ends at the same vertex, i.e., if $u = v$, and has length greater than zero.

Path: Undirected Graph

Definition: Let n be a nonnegative integer and G an undirected graph. A path of length n from u to v in G is a sequence of n edges $e_1, e_2, \ldots,$ e_n of G for which there exists a sequence $x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$ of vertices such that e_i has the endpoints x_{i-1} and x_i for $i = 1, \ldots, n$.

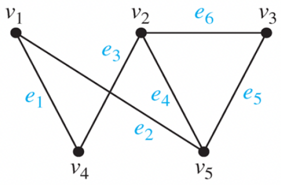


The path is a circuit if it begins and ends at the same vertex, i.e., if $u = v$, and has length greater than zero.

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

Path: Undirected Graph

Definition: Let n be a nonnegative integer and G an undirected graph. A path of length n from u to v in G is a sequence of n edges $e_1, e_2, \ldots,$ e_n of G for which there exists a sequence $x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$ of vertices such that e_i has the endpoints x_{i-1} and x_i for $i = 1, \ldots, n$.

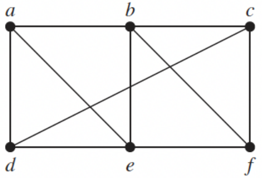


The path is a circuit if it begins and ends at the same vertex, i.e., if $u = v$, and has length greater than zero.

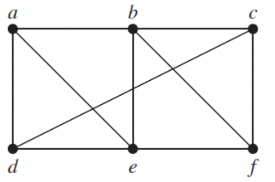
A path or circuit is simple if it does not contain the same edge more than once.

Length of a path = the number of edges on path

Path: Undirected Graph

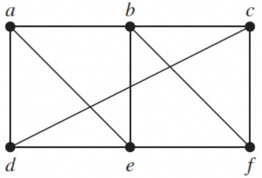


Path: Undirected Graph



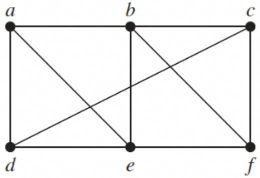
- a, d, c, f, e is a simple path of length 4.

Path: Undirected Graph



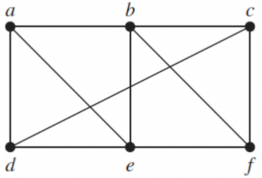
- a, d, c, f, e is a simple path of length 4.
- d, e, c, a is not a path, because $\{e, c\}$ is not an edge.

Path: Undirected Graph



- a, d, c, f, e is a simple path of length 4.
- d, e, c, a is not a path, because $\{e, c\}$ is not an edge.
- b, c, f, e, b is a circuit of length 4.

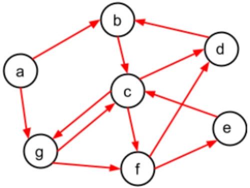
Path: Undirected Graph



- a, d, c, f, e is a simple path of length 4.
- d, e, c, a is not a path, because $\{e, c\}$ is not an edge.
- b, c, f, e, b is a circuit of length 4.
- The path a, b, e, d, a, b, which is of length 5, is not simple because it contains the edge $\{a, b\}$ twice.

Path: Directed Graph

Definition: Let n be a nonnegative integer and G an directed graph. A path of length n from u to v in G is a sequence of n edges $e_1, e_2, \ldots,$ e_n of G for which there exists a sequence $x_0 = u, x_1, \ldots, x_{n-1}, x_n = v$ of vertices such that e_i is associated with initial vertex x_{i-1} and terminal vertex x_i for $i = 1, \ldots, n$.



A path of length greater than zero that begins and ends at the same vertex is called a circuit or cycle.

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

Connectivity

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

Connectivity

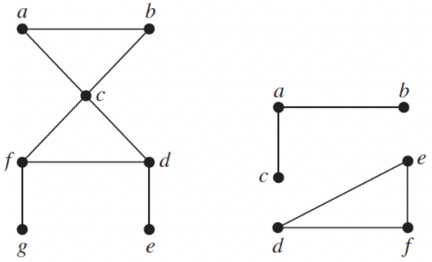
An undirected graph is called connected if there is a path between every pair of distinct vertices of the graph.

An undirected graph that is not connected is called disconnected.

Connectivity

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

An undirected graph that is not connected is called disconnected.

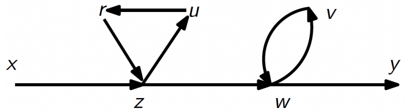


Connectivity

Lemma: If there is a path between two distinct vertices x and y of a graph G, then there is a simple path between x and y in G.

Connectivity

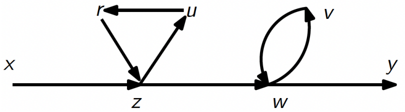
Lemma: If there is a path between two distinct vertices x and y of a graph G, then there is a simple path between x and y in G.
Proof: Just delete cycles (loops).



Path from x to y: $x, z, u, r, z, u, r, z, w, v, w, y$.
Path from x to y: x, z, w, y.

Connectivity

Lemma: If there is a path between two distinct vertices x and y of a graph G, then there is a simple path between x and y in G.
Proof: Just delete cycles (loops).



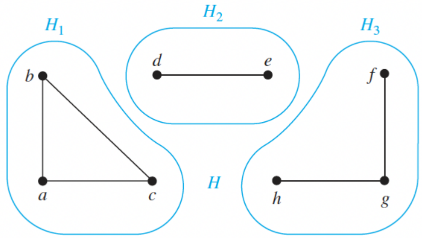
Path from x to y: $x, z, u, r, z, u, r, z, w, v, w, y$.
Path from x to y: x, z, w, y.

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

Connectivity

A connected component of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G.

A graph G that is not connected has two or more connected components that are disjoint and have G as their union.



Connectedness in Directed Graphs

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

Connectedness in Directed Graphs

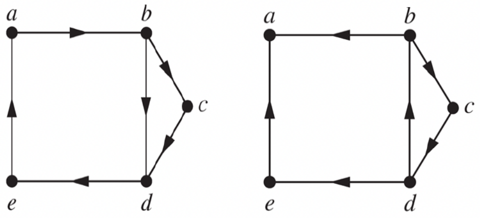
Definition: A directed graph is strongly connected if there is a path from a to b and a path from b to a whenever a and b are vertices in the graph.

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

Connectedness in Directed Graphs

Definition: A directed graph is strongly connected if there is a path from *a* to *b* and a path 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 every two vertices in the underlying undirected graph.

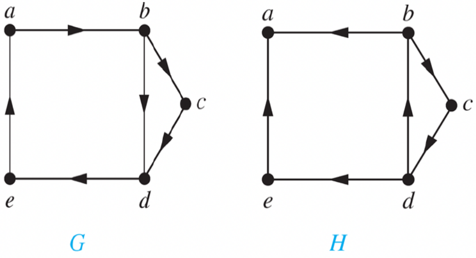


G H

Connectedness in Directed Graphs

Definition: A directed graph is strongly connected if there is a path from a to b and a path 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 every two vertices in the underlying undirected graph.



G is strongly connected; H is weakly connected.

Cut Vertices and Cut Edges

Sometimes the removal from a graph of a vertex and all incident edges disconnect the graph.

Cut Vertices and Cut Edges

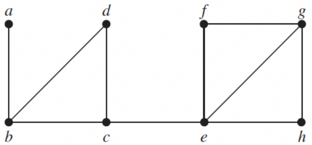
Sometimes the removal from a graph of a vertex and all incident edges disconnect the graph.

Such vertices are called cut vertices. Similarly we may define cut edges.

Cut Vertices and Cut Edges

Sometimes the removal from a graph of a vertex and all incident edges disconnect the graph.

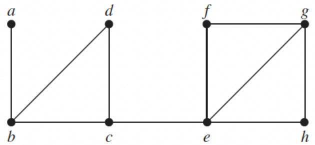
Such vertices are called cut vertices. Similarly we may define cut edges.



Cut Vertices and Cut Edges

Sometimes the removal from a graph of a vertex and all incident edges disconnect the graph.

Such vertices are called cut vertices. Similarly we may define cut edges.

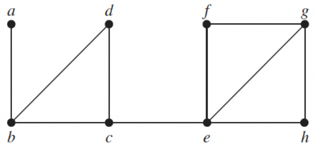


The cut vertices are b, c, and e.

Cut Vertices and Cut Edges

Sometimes the removal from a graph of a vertex and all incident edges disconnect the graph.

Such vertices are called cut vertices. Similarly we may define cut edges.



The cut vertices are b, c, and e.

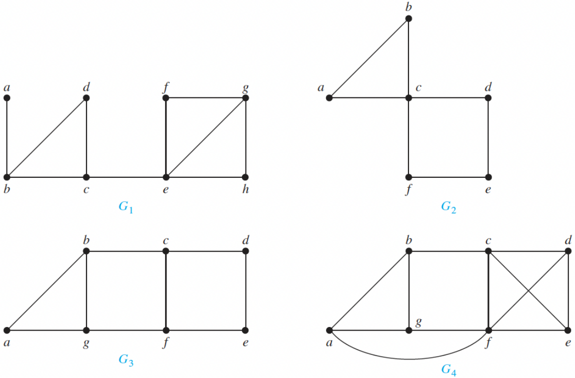
The cut edges are $\{a, b\}$ and $\{c, e\}$.

Cut Vertices and Cut Edges

A set of edges E' is called an edge cut of G if the subgraph $G - E'$ is disconnected. The edge connectivity $\lambda(G)$ is the minimum number of edges in an edge cut of G.

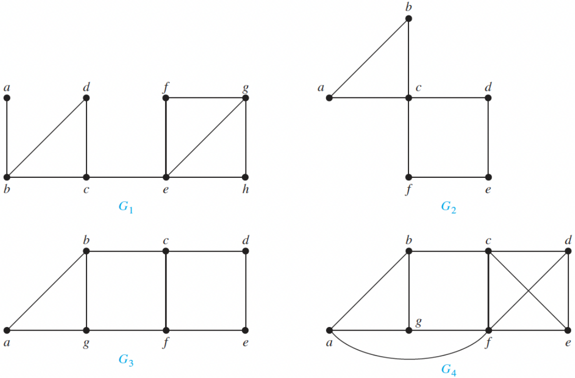
Cut Vertices and Cut Edges

A set of edges E' is called an edge cut of G if the subgraph $G - E'$ is disconnected. The edge connectivity $\lambda(G)$ is the minimum number of edges in an edge cut of G.



Cut Vertices and Cut Edges

A set of edges E' is called an edge cut of G if the subgraph $G - E'$ is disconnected. The edge connectivity $\lambda(G)$ is the minimum number of edges in an edge cut of G.



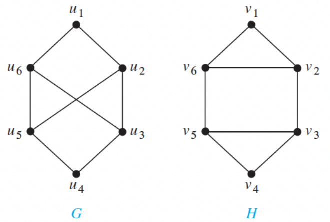
$\lambda(G_1) = 1$; $\lambda(G_2) = 2$; $\lambda(G_3) = 2$; $\lambda(G_4) = 3$

Paths and Isomorphism

The existence of a simple circuit of length k is isomorphic invariant. This can be used to construct mappings that may be isomorphisms.

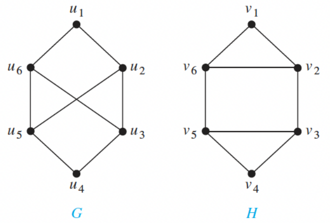
Paths and Isomorphism

The existence of a simple circuit of length k is isomorphic invariant. This can be used to construct mappings that may be isomorphisms.



Paths and Isomorphism

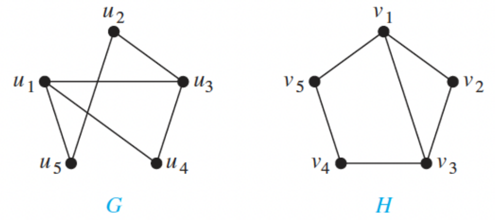
The existence of a simple circuit of length k is isomorphic invariant. This can be used to construct mappings that may be isomorphisms.



Not isomorphic. H has a simple circuit of length three, namely, v_1, v_2, v_6, v_1, whereas G has no simple circuit of length three.

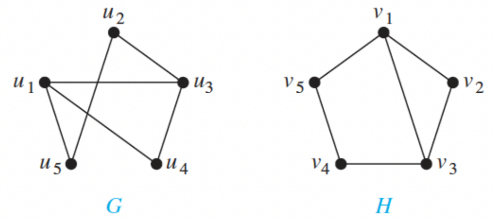
Paths and Isomorphism

The existence of a simple circuit of length k is isomorphic invariant. This can be used to construct mappings that may be isomorphisms.



Paths and Isomorphism

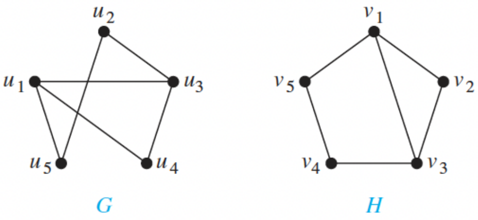
The existence of a simple circuit of length k is isomorphic invariant. This can be used to construct mappings that may be isomorphisms.



Because many isomorphic invariants (e.g., number of vertices/edges, degree, circuit) agree, G and H may be isomorphic.

Paths and Isomorphism

The existence of a simple circuit of length k is isomorphic invariant. This can be used to construct mappings that may be isomorphisms.



Because many isomorphic invariants (e.g., number of vertices/edges, degree, circuit) agree, G and H may be isomorphic. Let $f(u_1) = v_3$, $f(u_4) = v_2$, $f(u_3) = v_1$, $f(u_2) = v_5$, and $f(u_5) = v_4$. We can show that f is an isomorphism, so G and H are isomorphic.

Counting Paths between Vertices

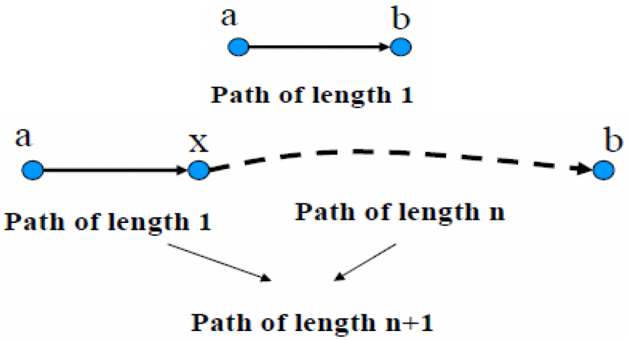
Theorem: Let G be a graph with adjacency matrix $\mathbf{A}$ with respect to the ordering $v_1, v_2, \ldots, v_n$ of vertices. The number of different paths of length r from v_i to v_j, where r is a positive integer, equals the (i, j)-th entry of $\mathbf{A}^r$.

Note: with directed or undirected edges, multiple edges and loops allowed

Recap: Path Length

Theorem: Let R be relation on a set A. There is a path of length n from a to b if and only if $(a, b) \in R^n$. (Boolean product.)

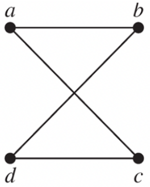
Proof (by induction):



Recall that $R^{n+1} = R^n \circ R$

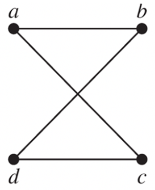
Counting Paths between Vertices:

How many paths of length 4 are there from a to d in the graph G?



Counting Paths between Vertices:

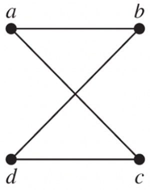
How many paths of length 4 are there from a to d in the graph G?



$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Counting Paths between Vertices:

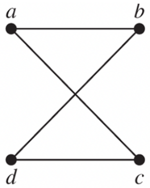
How many paths of length 4 are there from a to d in the graph G?



$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad \mathbf{A}^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

Counting Paths between Vertices:

How many paths of length 4 are there from a to d in the graph G?



$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad \mathbf{A}^4 = \begin{bmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{bmatrix}$$

a, b, a, b, d; a, b, a, c, d; a, b, d, b, d; a, b, d, c, d;
a, c, a, b, d; a, c, a, c, d; a, c, d, b, d; a, c, d, c, d

Counting Paths between Vertices

Theorem: The number of different paths of length r from v_i to v_j, where r is a positive integer, equals the (i, j)-th entry of $\mathbf{A}^r$.

Proof (by induction):

Counting Paths between Vertices

Theorem: The number of different paths of length r from v_i to v_j, where r is a positive integer, equals the (i, j)-th entry of $\mathbf{A}^r$.

Proof (by induction):

- Basic Step: The number of paths from v_i to v_j of length 1 is the (i, j)-th entry of $\mathbf{A}$.

Counting Paths between Vertices

Theorem: The number of different paths of length r from v_i to v_j, where r is a positive integer, equals the (i, j)-th entry of $\mathbf{A}^r$.

Proof (by induction):

- Basic Step: The number of paths from v_i to v_j of length 1 is the (i, j)-th entry of $\mathbf{A}$.
- Inductive hypothesis: Assume that the (i, j)-th entry of $\mathbf{A}^r$ is the number of different paths of length r from v_i to v_j.

Counting Paths between Vertices

Theorem: The number of different paths of length r from v_i to v_j, where r is a positive integer, equals the (i, j)-th entry of $\mathbf{A}^r$.

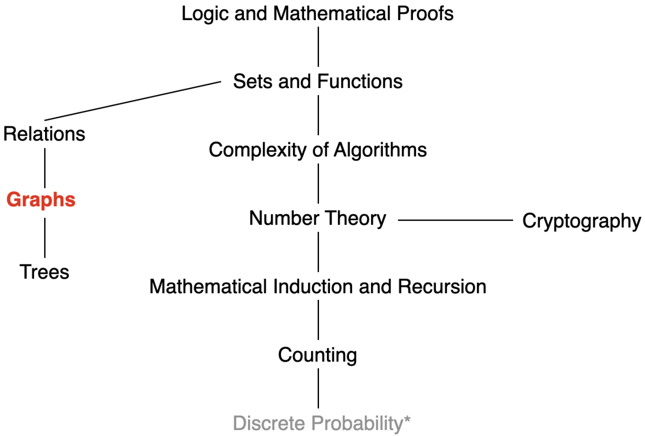
Proof (by induction):

- Basic Step: The number of paths from v_i to v_j of length 1 is the (i, j)-th entry of $\mathbf{A}$.
- Inductive hypothesis: Assume that the (i, j)-th entry of $\mathbf{A}^r$ is the number of different paths of length r from v_i to v_j.
- Inductive Step: $\mathbf{A}^{r+1} = \mathbf{A}^r\mathbf{A}$. The (i, j)-th entry of $\mathbf{A}^{r+1}$ equals

$$b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{in}a_{nj},$$

  where b_{ik} is the (i, k)-th entry of $\mathbf{A}^r$. By the inductive hypothesis, b_{ik} is the number of paths of length r from v_i to v_k.
- Inductive Conclusion: (i, j)-th entry of $\mathbf{A}^{r+1}$ counts all paths with length $r + 1$ for all possible intermediate vertices v_k.

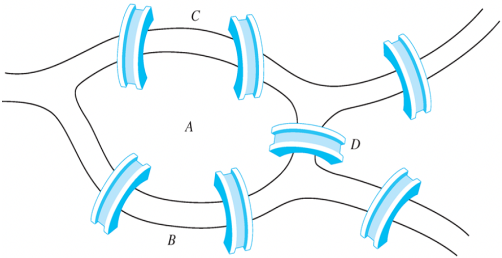
Next Lecture



Graph and terminologies, representing graphs and graph isomorphism, connectivity, Euler and Hamiliton path, ...

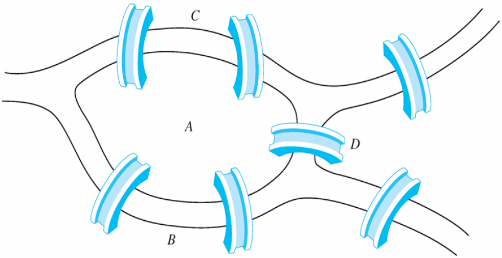
Euler Paths

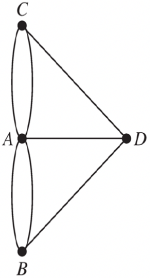
Königsberg seven-bridge problem: People wondered whether it was possible to start at some location in the town, travel across all the bridges once without crossing any bridge twice, and return to the starting point.



Euler Paths

Königsberg seven-bridge problem: People wondered whether it was possible to start at some location in the town, travel across all the bridges once without crossing any bridge twice, and return to the starting point.





Euler Paths and Circuits

Definition: An Euler circuit in a graph G is a simple circuit containing every edge of G. An Euler path in G is a simple path containing every edge of G.

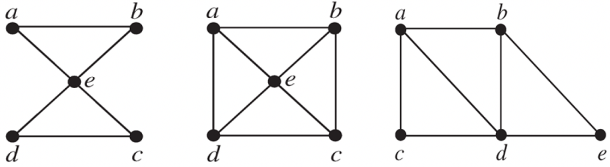
Recall that a path or circuit is simple if it does not contain the same edge more than once.

Euler Paths and Circuits

Definition: An Euler circuit in a graph G is a simple circuit containing every edge of G. An Euler path in G is a simple path containing every edge of G.

Recall that a path or circuit is simple if it does not contain the same edge more than once.

Example: Which of the undirected graphs have an Euler circuit? Of those that do not, which have an Euler path?



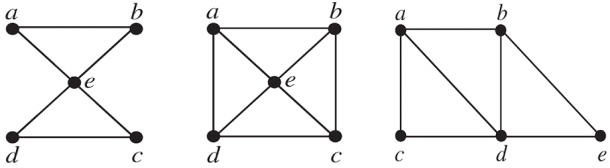
G_1 G_2 G_3

Euler Paths and Circuits

Definition: An Euler circuit in a graph G is a simple circuit containing every edge of G. An Euler path in G is a simple path containing every edge of G.

Recall that a path or circuit is simple if it does not contain the same edge more than once.

Example: Which of the undirected graphs have an Euler circuit? Of those that do not, which have an Euler path?



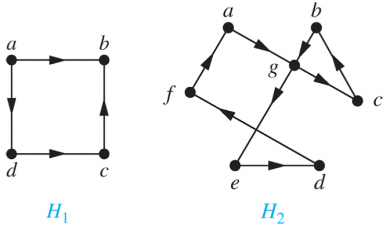
G_1 G_2 G_3

G_1: an Euler circuit, e.g., a, e, c, d, e, b, a;
G_2: neither; G_3: an Euler path, e.g., a, c, d, e, b, d, a, b

Euler Paths and Circuits

Definition: An Euler circuit in a graph G is a simple circuit containing every edge of G. An Euler path in G is a simple path containing every edge of G.

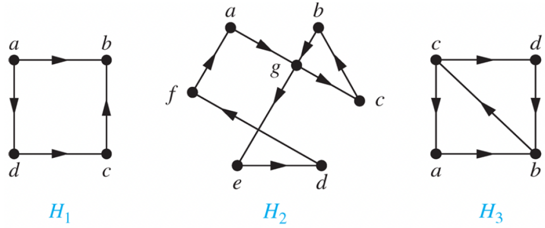
Example: Which of the directed graphs have an Euler circuit? Of those that do not, which have an Euler path?



Euler Paths and Circuits

Definition: An Euler circuit in a graph G is a simple circuit containing every edge of G. An Euler path in G is a simple path containing every edge of G.

Example: Which of the directed graphs have an Euler circuit? Of those that do not, which have an Euler path?



H_1: neither; H_2: an Euler circuit, e.g., a, g, c, b, g, e, d, f, a; H_3: an Euler path, e.g., c, a, b, c, d, b

Necessary Conditions for Euler Circuits and Paths

Euler Circuit $\Rightarrow$ The degree of every vertex must be even

- Each time the circuit passes through a vertex, it contributes two to the vertex's degree.
- The circuit starts with a vertex a and ends at a, then contributes two to $deg(a)$.

Euler Path $\Rightarrow$ The graph has exactly two vertices of odd degree

- The initial vertex and the final vertex of an Euler path have odd degree.