Discrete Mathematics and Its Applications

Lecture 7: Graphs: Representing Graphs and Graph Isomorphism

MING GAO

DaSE@ECNU
(for course related communications)
mgao@dase.ecnu.edu.cn

Dec. 27, 2018

Outline

Adjacency lists

Way i: One way to represent a graph without multiple edges is to list all the edges of this graph.

Adjacency lists

Way i: One way to represent a graph without multiple edges is to list all the edges of this graph.

Way ii: Another way to represent a graph with no multiple edges is to use **adjacency lists**, which specify the vertices that are adjacent to each vertex of the graph.

Adjacency lists

Way i: One way to represent a graph without multiple edges is to list all the edges of this graph.

Way ii: Another way to represent a graph with no multiple edges is to use **adjacency lists**, which specify the vertices that are adjacent to each vertex of the graph.

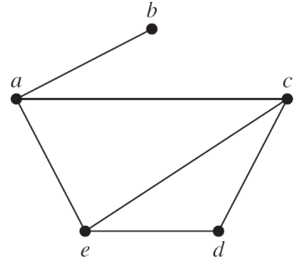


TABLE 1 An Adjacency List for a Simple Graph.

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

Adjacency lists for directed graph

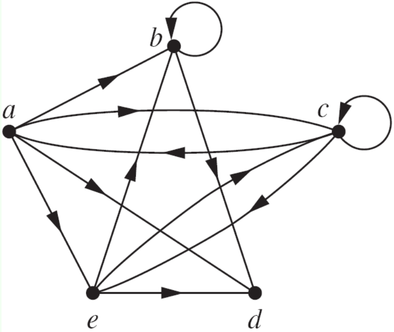


TABLE 2 An Adjacency List for a Directed Graph.

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 the vertices of $G = (V, E)$ are listed arbitrarily as $v_1, v_2, \cdots, v_n$. The **adjacency matrix** A (or A_G) of G, w.r.t. this listing of the vertices, 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, i.e., if its adjacency matrix is $A = [a_{ij}]$, then

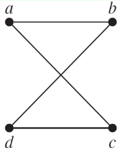
$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \text{ is an edge of } G; \\ 0, & \text{otherwise.} \end{cases}$$

Adjacency matrices

Definition

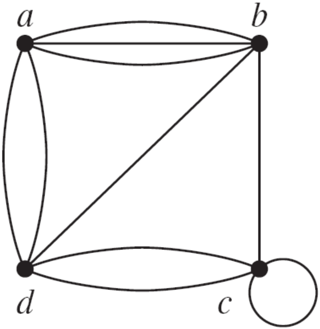
Suppose that the vertices of $G = (V, E)$ are listed arbitrarily as $v_1, v_2, \cdots, v_n$. The **adjacency matrix** A (or A_G) of G, w.r.t. this listing of the vertices, is the $n \times n$ zero-one matrix with 1 as its $(i,j)-$th entry when v_i and v_j are adjacent, and 0 as its $(i,j)-$th entry when they are not adjacent, i.e., if its adjacency matrix is $A = [a_{ij}]$, then

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



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

Adjacency matrices for pseudograph



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

Adjacency list VS. adjacency matrix

Trade-offs between adjacency lists and adjacency matrix:

Adjacency list VS. adjacency matrix

Trade-offs between adjacency lists and adjacency matrix:

- When a graph is sparse, it is usually preferable to use adjacency lists rather than an adjacency matrix to represent the graph.

Adjacency list VS. adjacency matrix

Trade-offs between adjacency lists and adjacency matrix:

- When a graph is sparse, it is usually preferable to use adjacency lists rather than an adjacency matrix to represent the graph.
- The adjacency matrix of a sparse graph is a sparse matrix, that is, a matrix with few nonzero entries, and there are special techniques for representing, and computing with, sparse matrices.

Adjacency list VS. adjacency matrix

Trade-offs between adjacency lists and adjacency matrix:

- When a graph is sparse, it is usually preferable to use adjacency lists rather than an adjacency matrix to represent the graph.
- The adjacency matrix of a sparse graph is a sparse matrix, that is, a matrix with few nonzero entries, and there are special techniques for representing, and computing with, sparse matrices.
- Suppose that a simple graph is dense, we compare the complexity of determining whether the possible edge (v_i, v_j) is present.

Adjacency list VS. adjacency matrix

Trade-offs between adjacency lists and adjacency matrix:

- When a graph is sparse, it is usually preferable to use adjacency lists rather than an adjacency matrix to represent the graph.
- The adjacency matrix of a sparse graph is a sparse matrix, that is, a matrix with few nonzero entries, and there are special techniques for representing, and computing with, sparse matrices.
- Suppose that a simple graph is dense, we compare the complexity of determining whether the possible edge (v_i, v_j) is present.
    - For case of adjacency matrix, we only need to examine the $(i, j)-$th entry in the matrix;

Adjacency list VS. adjacency matrix

Trade-offs between adjacency lists and adjacency matrix:

- When a graph is sparse, it is usually preferable to use adjacency lists rather than an adjacency matrix to represent the graph.
- The adjacency matrix of a sparse graph is a sparse matrix, that is, a matrix with few nonzero entries, and there are special techniques for representing, and computing with, sparse matrices.
- Suppose that a simple graph is dense, we compare the complexity of determining whether the possible edge (v_i, v_j) is present.
    - For case of adjacency matrix, we only need to examine the $(i, j)-$th entry in the matrix;
    - For case of adjacency list, we need to search the list of vertices adjacent to either v_i or v_j in $O(|V|)$.

Incidence matrices

Definition

Let $G = (V, E)$ be an undirected graph. Suppose that $v_1, v_2, \cdots, v_n$ are the vertices and $e_1, e_2, \cdots, e_m$ are the edges of G. Then the **incidence matrix** with respect to this ordering of V and E is the $n \times m$ matrix $M = [m_{ij}]$, where

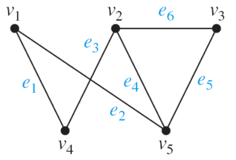
$$m_{ij} = \begin{cases} 1, & \text{when edge } e_j \text{ is incident with } v_i; \\ 0, & \text{otherwise.} \end{cases}$$

Incidence matrices

Definition

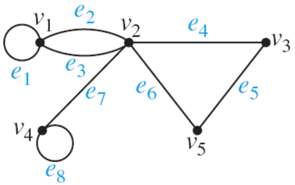
Let $G = (V, E)$ be an undirected graph. Suppose that $v_1, v_2, \cdots, v_n$ are the vertices and $e_1, e_2, \cdots, e_m$ are the edges of G. Then the **incidence matrix** with respect to this ordering of V and E is the $n \times m$ matrix $M = [m_{ij}]$, where

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



$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Incidence matrices for pseudograph



$$
\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array}
\begin{array}{c}
\begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0
\end{bmatrix}
\end{array}
$$

Random walk of a graph

Suppose that $G = (V, E)$ is a graph. Let $N(x) = \{y|(x, y) \in E\}$, and the degree of vertex x denote as $d(x) = |N(x)|$.

Random walk of a graph

Suppose that $G = (V, E)$ is a graph. Let $N(x) = \{y | (x, y) \in E\}$, and the degree of vertex x denote as $d(x) = |N(x)|$.

Markov Chain

For each $x \in V$, the transition probability matrix $P(y|x)$ is $\frac{1}{d(x)}$ if $y \in N(x)$, and $P(y|x) = 0$ otherwise. The **discrete-time Markov chain** X is a random walk on G.

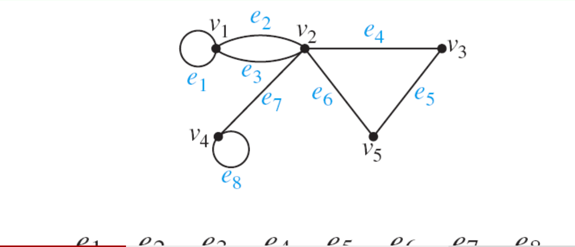
Random walk of a graph

Suppose that $G = (V, E)$ is a graph. Let $N(x) = \{y|(x, y) \in E\}$, and the degree of vertex x denote as $d(x) = |N(x)|$.

Markov Chain

For each $x \in V$, the transition probability matrix $P(y|x)$ is $\frac{1}{d(x)}$ if $y \in N(x)$, and $P(y|x) = 0$ otherwise. The **discrete-time Markov chain** X is a random walk on G.

Let $D = diag(d_1, d_2, \cdots, d_n)$ be a diagonal matrix, and $P = D^{-1}A$, which is the **transition probability matrix**.



Combinatorial Laplacian of graph

Definition

Given a graph G, (Combinatorial) Laplacian of G: $L = D - A$, i.e.,

$$L(u,v) = \begin{cases} d_v, & \text{if } u = v; \\ -1, & \text{if } u \text{ and } v \text{ are adjacent }; \\ 0, & \text{otherwise.} \end{cases}$$

Combinatorial Laplacian of graph

Definition

Given a graph G, (Combinatorial) Laplacian of G: $L = D - A$, i.e.,

$$L(u,v) = \begin{cases} d_v, & \text{if } u = v; \\ -1, & \text{if } u \text{ and } v \text{ are adjacent }; \\ 0, & \text{otherwise.} \end{cases}$$

If G is an undirected graph G, and its Laplacian matrix L with eigenvalues $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1}$, then

- L is singular and symmetric(existing $\lambda_i = 0$).
- Since row sum and column sum of L is zero, $\lambda_0 = 0$ and $v_0 = (1, 1, \cdots, 1)$.

Combinatorial Laplacian of graph

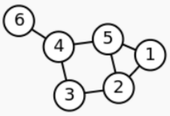
Definition

Given a graph G, (Combinatorial) Laplacian of G: $L = D - A$, i.e.,

$$L(u,v) = \begin{cases} d_v, & \text{if } u = v; \\ -1, & \text{if } u \text{ and } v \text{ are adjacent }; \\ 0, & \text{otherwise.} \end{cases}$$

If G is an undirected graph G, and its Laplacian matrix L with eigenvalues $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-1}$, then

- L is singular and symmetric(existing $\lambda_i = 0$).
- Since row sum and column sum of L is zero, $\lambda_0 = 0$ and $v_0 = (1, 1, \cdots, 1)$.

Labeled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2&0&0&0&0&0\\0&3&0&0&0&0\\0&0&2&0&0&0\\0&0&0&3&0&0\\0&0&0&0&3&0\\0&0&0&0&0&1 \end{pmatrix}$	$\begin{pmatrix} 0&1&0&0&1&0\\1&0&1&0&1&0\\0&1&0&1&0&0\\0&0&1&0&1&1\\1&1&0&1&0&0\\0&0&0&1&0&0 \end{pmatrix}$	$\begin{pmatrix} 2&-1&0&0&-1&0\\-1&3&-1&0&-1&0\\0&-1&2&-1&0&0\\0&0&-1&3&-1&-1\\-1&-1&0&-1&3&0\\0&0&0&-1&0&1 \end{pmatrix}$

Normalized Laplacian of graph

Definition

Given a graph G, normailized Laplacian of G: $\mathcal{L} = D^{-1/2} L D^{-1/2}$,

i.e., $\mathcal{L}(u,v) = \begin{cases} 1, & \text{if } u = v; \\ -\frac{1}{\sqrt{d_u d_v}}, & \text{if } u \text{ and } v \text{ are adjacent ;} \\ 0, & \text{otherwise.} \end{cases}$

Normalized Laplacian of graph

Definition

Given a graph G, normailized Laplacian of G: $\mathcal{L} = D^{-1/2} L D^{-1/2}$,

i.e., $\mathcal{L}(u,v) = \begin{cases} 1, & \text{if } u = v; \\ -\frac{1}{\sqrt{d_u d_v}}, & \text{if } u \text{ and } v \text{ are adjacent ;} \\ 0, & \text{otherwise.} \end{cases}$

- $\mathcal{L} = I - D^{-1/2} A D^{-1/2} = D^{1/2}(I - P) D^{-1/2}$. Thus, $\mathcal{L}$ is positive semidefinite and $0 \leq \lambda(\mathcal{L}) \leq 2$.

Normalized Laplacian of graph

Definition

Given a graph G, normailized Laplacian of G: $\mathcal{L} = D^{-1/2} L D^{-1/2}$,

i.e., $\mathcal{L}(u,v) = \begin{cases} 1, & \text{if } u = v; \\ -\frac{1}{\sqrt{d_u d_v}}, & \text{if } u \text{ and } v \text{ are adjacent ;} \\ 0, & \text{otherwise.} \end{cases}$

- $\mathcal{L} = I - D^{-1/2} A D^{-1/2} = D^{1/2}(I - P) D^{-1/2}$. Thus, $\mathcal{L}$ is positive semidefinite and $0 \leq \lambda(\mathcal{L}) \leq 2$.
- L is singular and symmetric, and $\lambda_0 = 0$ corresponding to eigenvector $D^{1/2} v_0^T = D^{1/2}(1, 1, \cdots, 1)^T$.

Normalized Laplacian of graph

Definition

Given a graph G, normailized Laplacian of G: $\mathcal{L} = D^{-1/2} L D^{-1/2}$,

i.e., $\mathcal{L}(u,v) = \begin{cases} 1, & \text{if } u = v; \\ -\frac{1}{\sqrt{d_u d_v}}, & \text{if } u \text{ and } v \text{ are adjacent ;} \\ 0, & \text{otherwise.} \end{cases}$

- $\mathcal{L} = I - D^{-1/2} A D^{-1/2} = D^{1/2}(I-P)D^{-1/2}$. Thus, $\mathcal{L}$ is positive semidefinite and $0 \le \lambda(\mathcal{L}) \le 2$.
- L is singular and symmetric, and $\lambda_0 = 0$ corresponding to eigenvector $D^{1/2} v_0^T = D^{1/2}(1, 1, \cdots, 1)^T$.
- P has an eigenvalue $1 - \lambda_i$, where λ_i is an eigenvalue of $\mathcal{L}$.

Normalized Laplacian of graph

Definition

Given a graph G, normailized Laplacian of G: $\mathcal{L} = D^{-1/2} L D^{-1/2}$,

i.e., $\mathcal{L}(u,v) = \begin{cases} 1, & \text{if } u = v; \\ -\frac{1}{\sqrt{d_u d_v}}, & \text{if } u \text{ and } v \text{ are adjacent }; \\ 0, & \text{otherwise.} \end{cases}$

- $\mathcal{L} = I - D^{-1/2} A D^{-1/2} = D^{1/2}(I - P)D^{-1/2}$. Thus, $\mathcal{L}$ is positive semidefinite and $0 \leq \lambda(\mathcal{L}) \leq 2$.
- L is singular and symmetric, and $\lambda_0 = 0$ corresponding to eigenvector $D^{1/2} v_0^T = D^{1/2}(1, 1, \cdots, 1)^T$.
- P has an eigenvalue $1 - \lambda_i$, where λ_i is an eigenvalue of $\mathcal{L}$.
- The regularization of graph G:
  $F^T \mathcal{L} F = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\frac{f_i}{\sqrt{d_{ii}}} - \frac{f_j}{\sqrt{d_{jj}}} \right)^2$.

Normalized Laplacian of graph

Definition

Given a graph G, normailized Laplacian of G: $\mathcal{L} = D^{-1/2}LD^{-1/2}$,

i.e., $\mathcal{L}(u,v) = \begin{cases} 1, & \text{if } u = v; \\ -\frac{1}{\sqrt{d_u d_v}}, & \text{if } u \text{ and } v \text{ are adjacent ;} \\ 0, & \text{otherwise.} \end{cases}$

- $\mathcal{L} = I - D^{-1/2}AD^{-1/2} = D^{1/2}(I-P)D^{-1/2}$. Thus, $\mathcal{L}$ is positive semidefinite and $0 \le \lambda(\mathcal{L}) \le 2$.
- L is singular and symmetric, and $\lambda_0 = 0$ corresponding to eigenvector $D^{1/2}v_0^T = D^{1/2}(1,1,\cdots,1)^T$.
- P has an eigenvalue $1-\lambda_i$, where λ_i is an eigenvalue of $\mathcal{L}$.
- The regularization of graph G: $F^T\mathcal{L}F = \frac{1}{2}\sum_{i=1}^n\sum_{j=1}^n\left(\frac{f_i}{\sqrt{d_{ii}}} - \frac{f_j}{\sqrt{d_{jj}}}\right)^2$.
- For weighted graph G, Laplacian and normalized Laplacian can be defined in a same manner.

Isomorphism of graphs

Definition

The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are **isomorphic** if there exists a 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.

Isomorphism of graphs

Definition

The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are **isomorphic** if there exists a 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.

- When two simple graphs are isomorphic, there is a one-to-one correspondence between vertices of the two graphs that preserves the adjacency relationship.

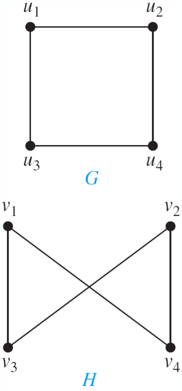
Isomorphism of graphs

Definition

The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are **isomorphic** if there exists a 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.

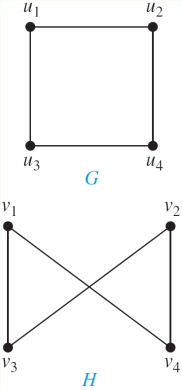
- When two simple graphs are isomorphic, there is a one-to-one correspondence between vertices of the two graphs that preserves the adjacency relationship.
- Isomorphism of simple graphs is an equivalence relation.

Example



Show that the graphs $G = (V, E)$ and $H = (W, F)$, displayed in the figure, are isomorphic.

Example



Show that the graphs $G = (V, E)$ and $H = (W, F)$, displayed in the figure, are isomorphic.

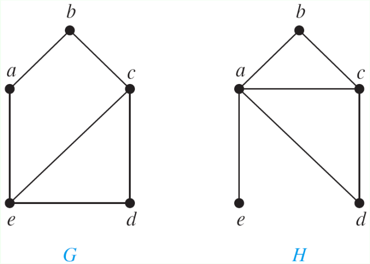
$$u_1 \to v_1$$
$$u_2 \to v_4$$
$$u_3 \to v_3$$
$$u_4 \to v_2$$

It is often difficult to determine whether two simple graphs are isomorphic. There are $n!$ possible one-to-one correspondences between the vertex sets of two simple graphs with n vertices.

How to determine?

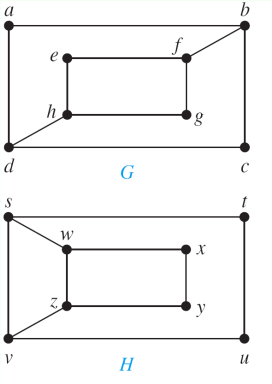
A property preserved by isomorphism of graphs is called a **graph invariant**.

- The same number of vertices;
- The same number of edges;
- The same degree for the same vertex.



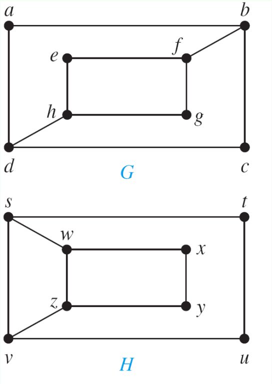
Determine whether the graphs displayed in the figure are isomorphic or not.

How to determine? Cont'd



Question: Determine whether the graphs displayed in the figure are isomorphic or not.

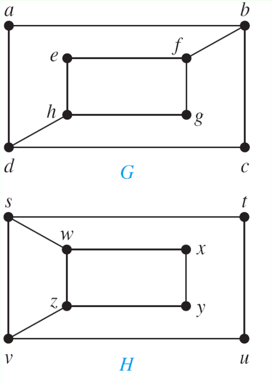
How to determine? Cont'd



Question: Determine whether the graphs displayed in the figure are isomorphic or not.

Solution: Note that because $deg(a) = 2$ in G, a must correspond to either t, u, x, or y in H, because these are the vertices of degree two in H.

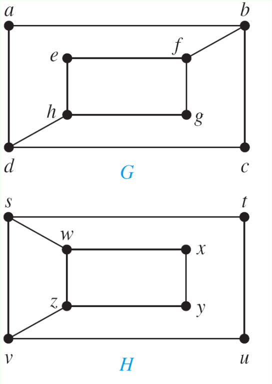
How to determine? Cont'd



Question: Determine whether the graphs displayed in the figure are isomorphic or not.

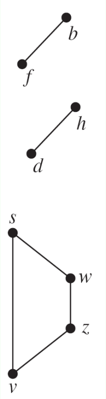
Solution: Note that because $deg(a) = 2$ in G, a must correspond to either t, u, x, or y in H, because these are the vertices of degree two in H. However, each of these four vertices in H is adjacent to another vertex of degree two in H, which is not true for a in G.

How to determine? Cont'd



Question: Determine whether the graphs displayed in the figure are isomorphic or not.

Solution: Note that because $deg(a) = 2$ in G, a must correspond to either t, u, x, or y in H, because these are the vertices of degree two 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.



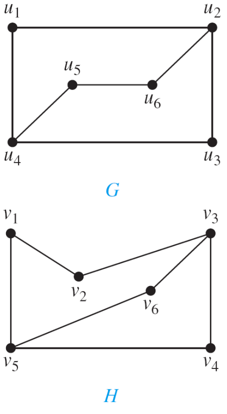
Find an efficient way to determine

Question: Determine whether the graphs displayed in the figure are isomorphic or not.

Find an efficient way to determine

Question: Determine whether the graphs displayed in the figure are isomorphic or not.

Solution:



$$
\mathbf{A}_G = \begin{array}{c|cccccc} & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 \\ \hline u_1 & 0 & 1 & 0 & 1 & 0 & 0 \\ u_2 & 1 & 0 & 1 & 0 & 0 & 1 \\ u_3 & 0 & 1 & 0 & 1 & 0 & 0 \\ u_4 & 1 & 0 & 1 & 0 & 1 & 0 \\ u_5 & 0 & 0 & 0 & 1 & 0 & 1 \\ u_6 & 0 & 1 & 0 & 0 & 1 & 0 \end{array}
$$

$$
\mathbf{A}_H = \begin{array}{c|cccccc} & v_6 & v_3 & v_4 & v_5 & v_1 & v_2 \\ \hline v_6 & 0 & 1 & 0 & 1 & 0 & 0 \\ v_3 & 1 & 0 & 1 & 0 & 0 & 1 \\ v_4 & 0 & 1 & 0 & 1 & 0 & 0 \\ v_5 & 1 & 0 & 1 & 0 & 1 & 0 \\ v_1 & 0 & 0 & 0 & 1 & 0 & 1 \\ v_2 & 0 & 1 & 0 & 0 & 1 & 0 \end{array}
$$

Paths in undirected graphs

Definition

Let n be a nonnegative integer and G a directed graph. A **path** of length n from u to v in G is a sequence of n edges $e_1, \cdots, e_n$ of G such that e_i is associated with (x_{i-1}, x_i) for $i = 1, 2, \cdots, n$, where $x_0 = u$ and $x_n = v$.

Paths in undirected graphs

Definition

Let n be a nonnegative integer and G a directed graph. A **path** of length n from u to v in G is a sequence of n edges $e_1, \cdots, e_n$ of G such that e_i is associated with (x_{i-1}, x_i) for $i = 1, 2, \cdots, n$, where $x_0 = u$ and $x_n = v$.
When there are no multiple edges in the directed graph, this path is denoted by its vertex sequence $x_0, x_1, x_2, \cdots, x_n$. A path of length greater than zero that begins and ends at the same vertex is called a **circuit or cycle**.

Paths in undirected graphs

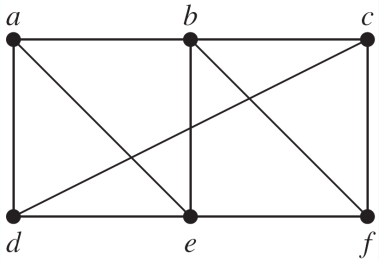
Definition

Let n be a nonnegative integer and G a directed graph. A **path** of length n from u to v in G is a sequence of n edges $e_1, \cdots, e_n$ of G such that e_i is associated with (x_{i-1}, x_i) for $i = 1, 2, \cdots, n$, where $x_0 = u$ and $x_n = v$.

When there are no multiple edges in the directed graph, this path is denoted by its vertex sequence $x_0, x_1, x_2, \cdots, x_n$. A path of length greater than zero that begins and ends at the same vertex is called a **circuit or cycle**.

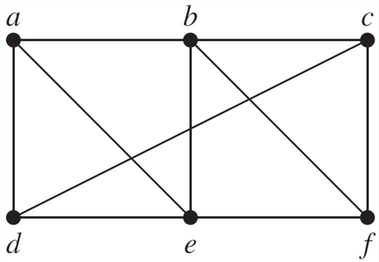
A path or circuit is called simple if it does not contain the same edge more than once.

Example



In the simple graph, a, d, c, f, e is a simple path of length 4, because $\{a, d\}, \{d, c\}$, $\{c, f\}$, and $\{f, e\}$ are all edges.

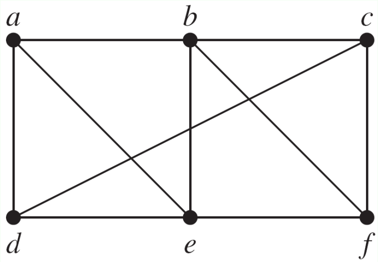
Example



In the simple graph, a, d, c, f, e is a simple path of length 4, because $\{a, d\}, \{d, c\}$, $\{c, f\}$, and $\{f, e\}$ are all edges.

However, d, e, c, a is not a path, because $\{e, c\}$ is not an edge.

Example

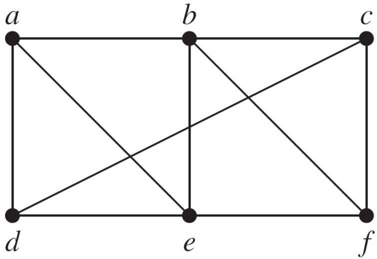


In the simple graph, a, d, c, f, e is a simple path of length 4, because $\{a, d\}, \{d, c\}$, $\{c, f\}$, and $\{f, e\}$ are all edges.

However, d, e, c, a is not a path, because $\{e, c\}$ is not an edge.

Note that b, c, f, e, b is a circuit of length 4 because $\{b, c\}$, $\{c, f\}$, $\{f, e\}$, and $\{e, b\}$ are edges, and this path begins and ends at b.

Example



In the simple graph, a, d, c, f, e is a simple path of length 4, because $\{a, d\}, \{d, c\}$, $\{c, f\}$, and $\{f, e\}$ are all edges.

However, d, e, c, a is not a path, because $\{e, c\}$ is not an edge.

Note that b, c, f, e, b is a circuit of length 4 because $\{b, c\}$, $\{c, f\}$, $\{f, e\}$, and $\{e, b\}$ are edges, and this path begins and ends at b.

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

Paths in directed graphs

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 $e_1, \cdots, e_n$ of G for which there exists a sequence $x_0 = u, x_1, \cdots, x_{n-1}, x_n = v$ of vertices such that e_i has, for $i = 1, \cdots, n$, the endpoints x_{i-1} and x_i. When the graph is simple, we denote this path by its vertex sequence $x_0, x_1, \cdots, x_n$ (because listing these vertices uniquely determines the path).

Paths in directed graphs

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 $e_1, \cdots, e_n$ of G for which there exists a sequence $x_0 = u, x_1, \cdots, x_{n-1}, x_n = v$ of vertices such that e_i has, for $i = 1, \cdots, n$, the endpoints x_{i-1} and x_i. When the graph is simple, we denote this path by its vertex sequence $x_0, x_1, \cdots, x_n$ (because listing these vertices uniquely determines the path).
The path is a **circuit** if it begins and ends at the same vertex, that is, if $u = v$, and has length greater than zero. The path or circuit is said to pass through the vertices $x_1, x_2, \cdots, x_{n-1}$ or traverse the edges $e_1, \cdots, e_n$.

Paths in directed graphs

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 $e_1, \cdots, e_n$ of G for which there exists a sequence $x_0 = u, x_1, \cdots, x_{n-1}, x_n = v$ of vertices such that e_i has, for $i = 1, \cdots, n$, the endpoints x_{i-1} and x_i.

When the graph is simple, we denote this path by its vertex sequence $x_0, x_1, \cdots, x_n$ (because listing these vertices uniquely determines the path).

The path is a **circuit** if it begins and ends at the same vertex, that is, if $u = v$, and has length greater than zero. The path or circuit is said to pass through the vertices $x_1, x_2, \cdots, x_{n-1}$ or traverse the edges $e_1, \cdots, e_n$.

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

Example

Paths in acquaintanceship graphs

In an acquaintanceship graph there is a path between two people if there is a chain of people linking these people, where two people adjacent in the chain know one another.

Example

Paths in acquaintanceship graphs

In an acquaintanceship graph there is a path between two people if there is a chain of people linking these people, where two people adjacent in the chain know one another.

Paths in collaboration graphs

In a collaboration graph, two people a and b are connected by a path when there is a sequence of people starting with a and ending with b such that the endpoints of each edge in the path are people who have collaborated.

Connectedness in undirected graphs

Definition

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

Connectedness in undirected graphs

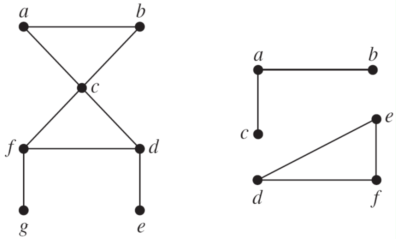
Definition

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**.

Connectedness in undirected graphs

Definition

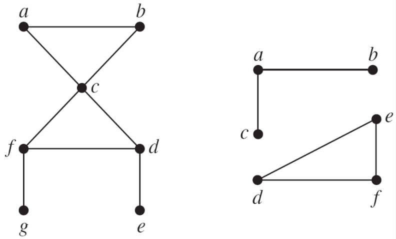
An undirected graph is called **connected** if there is a path between every pair of distinct vertices of the graph.
An undirected graph that is not connected is called **disconnected**.



Connectedness in undirected graphs

Definition

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**.

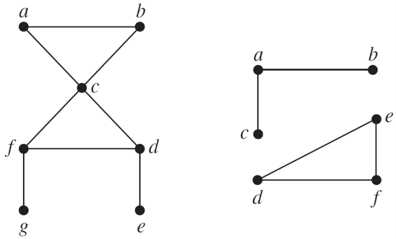


The first graph is connected, because for every pair of distinct vertices there is a path between them.

Connectedness in undirected graphs

Definition

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**.



The first graph is connected, because for every pair of distinct vertices there is a path between them.
However, the second graph is not connected. For instance, there is no path in the graph between vertices a and d.

Property

Theorem

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

Property

Theorem

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

Proof.

Let u and v be two distinct vertices of the connected undirected graph $G = (V, E)$.
Because G is connected, there is at least one path between u and v.
Let $x_0, x_1, \cdots, x_n$, where $x_0 = u$ and $x_n = v$, be the vertex sequence of a path of least length. This path of least length is simple.

Property

Theorem

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

Proof.

Let u and v be two distinct vertices of the connected undirected graph $G = (V, E)$.
Because G is connected, there is at least one path between u and v.
Let $x_0, x_1, \cdots, x_n$, where $x_0 = u$ and $x_n = v$, be the vertex sequence of a path of least length. This path of least length is simple.
To see this, suppose it is not simple. Then $x_i = x_j$ for some i and j with $0 \leq i < j$. This means that there is a path from u to v of shorter length with vertex sequence $x_0, x_1, \cdots, x_{i-1}, x_j, \cdots, x_n$ obtained by deleting the edges corresponding to the vertex sequence $x_i, \cdots, x_{j-1}$. □

Connected component

Definition

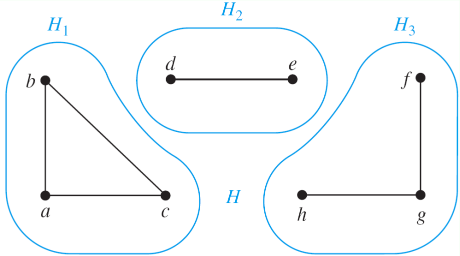
A **connected component** of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G.
That is, a connected component of a graph G is a maximal connected subgraph of G.

Connected component

Definition

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.

That is, a connected component of a graph G is a maximal connected subgraph of G.

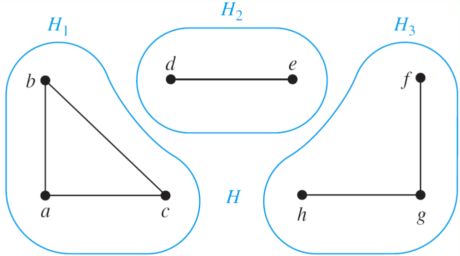


Connected component

Definition

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.

That is, a connected component of a graph G is a maximal connected subgraph of G.



The graph H is the union of three disjoint connected subgraphs H_1, H_2, and H_3. These three subgraphs are the connected components of H.

Connected components of call graphs

Two vertices x and y are in the same component of a telephone call graph when there is a sequence of telephone calls beginning at x and ending at y.

Connected components of call graphs

Two vertices x and y are in the same component of a telephone call graph when there is a sequence of telephone calls beginning at x and ending at y.

When a call graph for telephone calls made during a particular day in the AT&T network was analyzed, this graph was found to have

- 53,767,087 vertices, and more than 170 million edges;
- More than 3.7 million connected components. Most of these components were small; approximately three-fourths consisted of two vertices representing pairs of telephone numbers that called only each other.

Connected components of call graphs

Two vertices x and y are in the same component of a telephone call graph when there is a sequence of telephone calls beginning at x and ending at y.

When a call graph for telephone calls made during a particular day in the AT&T network was analyzed, this graph was found to have

- 53,767,087 vertices, and more than 170 million edges;
- More than 3.7 million connected components. Most of these components were small; approximately three-fourths consisted of two vertices representing pairs of telephone numbers that called only each other.

This graph has one huge connected component with 44,989,297 vertices comprising more than 80% of the total (giant component). Furthermore, every vertex in this component can be linked to any other vertex by a chain of no more than 20 calls.

Network robustness

Motivations

- In a phone call network, dense and frequent calls among users in the network reduce the likelihood of churn.
- In IP networks, service providers therefore aim to monitor, manage and optimize their networks to keep their networks robust.
- In social platforms, some external or internal events may be detected from the burst of user interaction networks.

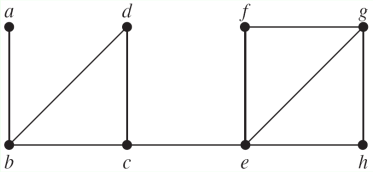
Existing measurements

- Node connectivity and edge connectivity
- Cheeger ratio, vertex expansion, and edge expansion
- Algebraic connectivity and R-energy

Cut and bridge

Definition

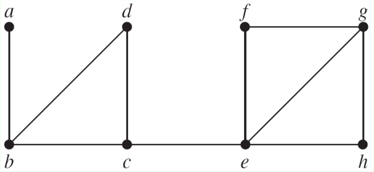
Sometimes the removal from a graph of a vertex and all incident edges produces a subgraph with more connected components. Such vertices are called **cut vertices** (or articulation points). Analogously, an edge whose removal produces a graph with more connected components than in the original graph is called a **cut edge** or **bridge**.
Note that not all graphs have cut vertices, e.g., K_n for $n \geq 3$.



Cut and bridge

Definition

Sometimes the removal from a graph of a vertex and all incident edges produces a subgraph with more connected components. Such vertices are called **cut vertices** (or articulation points). Analogously, an edge whose removal produces a graph with more connected components than in the original graph is called a **cut edge** or **bridge**.
Note that not all graphs have cut vertices, e.g., K_n for $n \geq 3$.

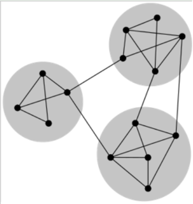


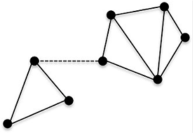
Question:
Find the cut vertices and cut edges in the left graph G.
Solution:
The cut vertices of G are b, c, and e.
The cut edges are $\{a, b\}$ and $\{c, e\}$.

Connectivity

Vertex connectivity or edge connectivity

Node connectivity (edge connectivity) $v(G)$ ($\epsilon(G)$) of a network G is defined by the minimum number of nodes (edges) that are removed to break the networks into multiple connected components.

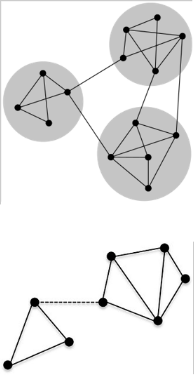




Connectivity

Vertex connectivity or edge connectivity

Node connectivity (edge connectivity) $v(G)$ ($\epsilon(G)$) of a network G is defined by the minimum number of nodes (edges) that are removed to break the networks into multiple connected components.



- The larger $v(G)$ or $\epsilon(G)$ are, the more connected we consider G to be;
- Disconnected graphs and K_1 have $v(G) = \epsilon(G) = 0$;
- We say that a graph is k-connected (or k-vertex-connected), if $v(G) \geq k$;
- $v(G) \leq \epsilon(G) \leq \min_{v \in V} deg(v)$.

Expander robustness

Definitions

Let $G = (V, E)$ be a connected and undirected network.

- $\partial(S)$ is the edge boundary of S (i.e., the set of edges with exactly one endpoint in S).
- $\partial_{out}(S)$ is the outer vertex boundary of S (i.e., the set of vertices in $V \backslash S$ with at least one neighbor in S).
- $vol(S)$ is the total degree of all vertices in S.

Expander robustness

Definitions

Let $G = (V, E)$ be a connected and undirected network.

- $\partial(S)$ is the edge boundary of S (i.e., the set of edges with exactly one endpoint in S).
- $\partial_{out}(S)$ is the outer vertex boundary of S (i.e., the set of vertices in $V \setminus S$ with at least one neighbor in S).
- $vol(S)$ is the total degree of all vertices in S.
- Cheeger ratio: $h(G) = \min\limits_{S \subset V} \dfrac{|\partial(S)|}{\min\{vol(S), vol(\overline{S})\}}$
- Vertex expansion: $h_v(G) = \min\limits_{S \subset V, 0<|S|<|V|/2} \dfrac{|\partial_{out}(S)|}{|S|}$
- Edge expansion: $h_e(G) = \min\limits_{S \subset V, 0<|S|<|V|/2} \dfrac{|\partial(S)|}{|S|}$

Laplacian robustness

Algebraic connectivity

Algebraic connectivity $\lambda(G)$ is defined by the second smallest eigenvalue of the Laplacian matrix of network G.

- $\lambda(G) \leq v(G) \leq \epsilon(G)$.
- $\lambda(G) = 0$ if G is disconnected.

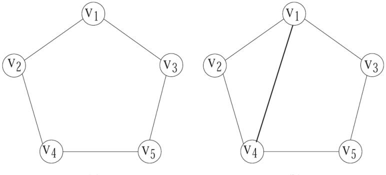
R-energy

The **robustness energy (R-energy)** of G is defined as $E(G) = \frac{1}{n-1}\sum_{i=2}^{n}(\lambda_i - \overline{\lambda})^2$, where λ_i are eigenvalues of normalized Laplacian of network G, and $\overline{\lambda} = \frac{1}{n-1}\sum_{i=2}^{n}\lambda_i$.

- $E(G) = \frac{1}{n-1}\sum_{i=1}^{n}\sum_{j\neq i}^{n}\frac{A_{ij}}{d(v_i)d(v_j)} - \frac{n}{(n-1)^2}$ (smaller is better).
- $E(G)$ is reasonable robustness metric to evaluate a disconnected network.
- $E(G)$ can be efficiently computed in $O(|V| + |E|)$.

Example of network robustness metrics

Analysis

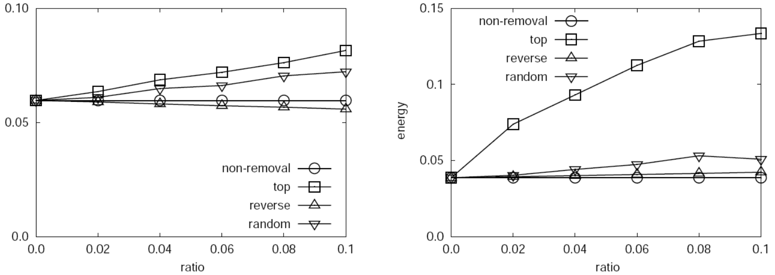


(a) (b)

networks	Connectivity			Expansion		
	node	edge	algebraic	vertex	edge	Cheeger
Figure 1(a)	2	2	1.382	1	1	0.5
Figure 1(b)	2	2	1.382	1	1	0.5

- The table illustrates that they are unreasonable to evaluate the robustness of networks.
- The R-energies of networks shown in Figures (a) and (b) are 0.222 and 0.074, respectively. Thus, R-energy is more reasonable.

R-energy: application I

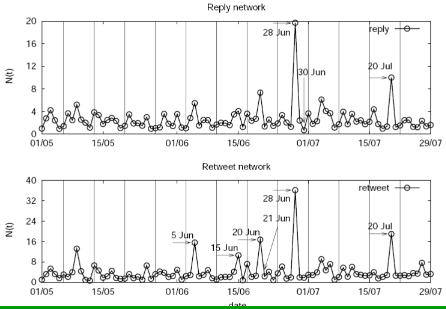


(a) CA-HepPh (b) Email-Enron

Analysis

- Networks become less robust sooner when vertices of the highest degrees are removed.
- Networks remain robust or become slightly more robust when vertices of the smallest degrees are removed.

R-energy: application II



Analysis

- On June 28, the top three words from retweets with highest frequency difference are "tax", "Obamacar" and "scotu". Actually, the Obamacare healthcare law was upheld by the Supreme Court of United States, and there were concerns about tax increase as its outcome.
- Twitter goes down in worst crash in 8 months on July 20.

Strongly and weakly connected

Definition

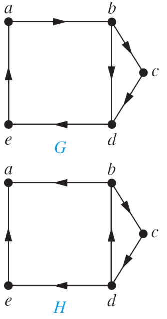
A directed graph is **strongly connected** if there is a path from a to b and from b to a whenever a and b are vertices in the graph.

Strongly and weakly connected

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.

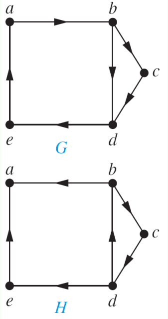
A directed graph is **weakly connected** if there is a path between every two vertices in the underlying undirected graph.



Strongly and weakly connected

Definition

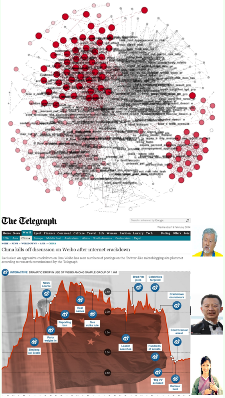
A directed graph is **strongly connected** if there is a path from a to b and from b to a whenever a and b are vertices in the graph.
A directed graph is **weakly connected** if there is a path between every two vertices in the underlying undirected graph.



- Are the directed graphs G and H shown in figure strongly connected? Are they weakly connected?
- The maximal strongly connected subgraphs, are called the strongly connected components or strong components of G.

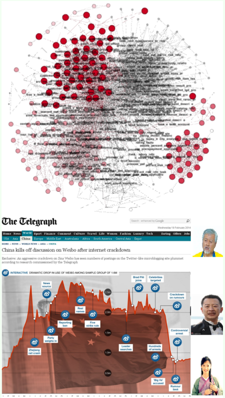
Giant component

Giant component



Giant component

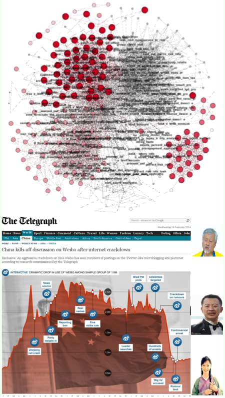
Giant component



A connected component that contains a significant fraction of all the nodes.

Giant component

Giant component

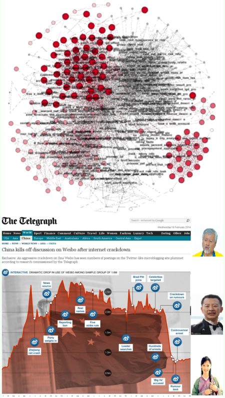


A connected component that contains a significant fraction of all the nodes.

- When a network (e.g., friendship network) contains a giant component, it almost always contains only one.

Giant component

Giant component

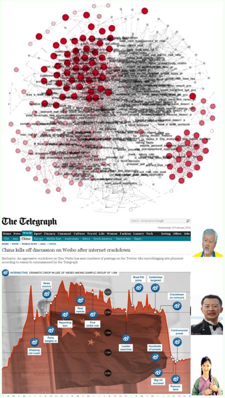


A connected component that contains a significant fraction of all the nodes.

- When a network (e.g., friendship network) contains a giant component, it almost always contains only one.
- The other connected components are very small by comparison.

Giant component

Giant component

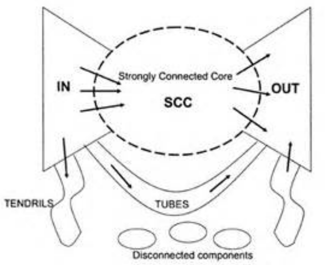


A connected component that contains a significant fraction of all the nodes.

- When a network (e.g., friendship network) contains a giant component, it almost always contains only one.
- The other connected components are very small by comparison.
- The largest connected component would break apart into three distinct components if this node were removed [related to robustness of network].

Web giant component

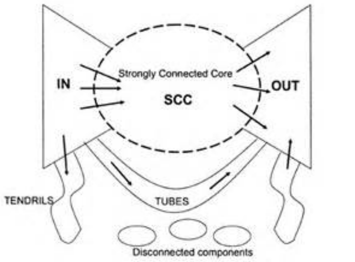
Web graph



200 M pages, 1.5 B hyperlinks

Web giant component

Web graph



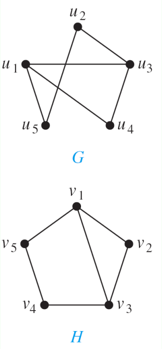
200 M pages, 1.5 B hyperlinks

Web contains a giant strongly connected component (containing home pages of many of the major commercial, governmental, and nonprofit organizations)

- IN: nodes that can reach the giant SCC but cannot be reached from it, i.e., nodes that are "upstream" of it.
- OUT: nodes that can be reached from the giant SCC but cannot reach it, i.e., nodes are "downstream" of it.

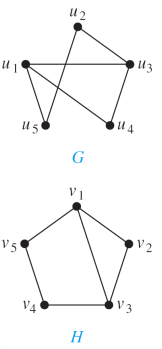
Paths and isomorphis

Paths and circuits can determine whether two graphs are isomorphic.



Paths and isomorphis

Paths and circuits can determine whether two graphs are isomorphic.



- The existence of a simple circuit of a particular length is a useful invariant;
- Paths can be used to construct mappings that may be isomorphisms;
- The paths u_1, u_4, u_3, u_2, u_5 in G and v_3, v_2, v_1, v_5, v_4 in H both go through every vertex in the graph;
-

$$
\begin{aligned}
f(u_1) &= v_3 \\
f(u_4) &= v_2 \\
f(u_3) &= v_1 \\
f(u_2) &= v_5 \\
f(u_5) &= v_4.
\end{aligned}
$$

Counting paths between vertices

Theorem

Let G be a graph with adjacency matrix A with respect to the ordering $v_1, v_2, \cdots, v_n$ of the vertices of the graph (with directed or undirected edges, with multiple edges and loops allowed). The number of different paths of length r from v_i to v_j equals the $(i, j)-$th entry of A^r for $r \in Z^+$.

Counting paths between vertices

Theorem

Let G be a graph with adjacency matrix A with respect to the ordering $v_1, v_2, \cdots, v_n$ of the vertices of the graph (with directed or undirected edges, with multiple edges and loops allowed). The number of different paths of length r from v_i to v_j equals the $(i,j)-$th entry of A^r for $r \in Z^+$.

Proof.

Basic step:

Let G be a graph with adjacency matrix A. The number of paths from v_i to v_j of length 1 is the $(i,j)-$th entry of A, because this entry is the number of edges from v_i to v_j. □

Proof Cont'd

Inductive step:

Assume that the $(i, j)-$th entry of A^r is the number of different paths of length r from v_i to v_j. This is the inductive hypothesis. Because $A^{r+1} = A^r \cdot A$, the $(i, j)-$th entry of 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 A^r. By the inductive hypothesis, b_{ik} is the number of paths of length r from v_i to v_k.

Proof Cont'd

Inductive step:

Assume that the $(i,j)-$th entry of A^r is the number of different paths of length r from v_i to v_j. This is the inductive hypothesis. Because $A^{r+1} = A^r \cdot A$, the $(i,j)-$th entry of 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 A^r. By the inductive hypothesis, b_{ik} is the number of paths of length r from v_i to v_k.

A path of length $r+1$ from v_i to v_j is made up of a path of length r from v_i to some intermediate vertex v_k, and an edge from v_k to v_j.

Proof Cont'd

Inductive step:
Assume that the $(i, j)-$th entry of A^r is the number of different paths of length r from v_i to v_j. This is the inductive hypothesis. Because $A^{r+1} = A^r \cdot A$, the $(i, j)-$th entry of 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 A^r. By the inductive hypothesis, b_{ik} is the number of paths of length r from v_i to v_k.
A path of length $r + 1$ from v_i to v_j is made up of a path of length r from v_i to some intermediate vertex v_k, and an edge from v_k to v_j. By the product rule for counting, the number of such paths is the product of the number of paths of length r from v_i to v_k, namely, b_{ik}, and the number of edges from v_k to v_j, namely, a_{kj}.

Proof Cont'd

Inductive step:

Assume that the $(i, j)-$th entry of A^r is the number of different paths of length r from v_i to v_j. This is the inductive hypothesis. Because $A^{r+1} = A^r \cdot A$, the $(i, j)-$th entry of A^{r+1} equals

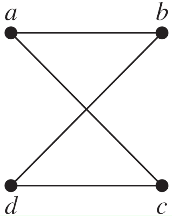
$$b_{i1}a_{1j} + b_{i2}a_{2j} + \cdots + b_{in}a_{nj},$$

where b_{ik} is the $(i, k)-$th entry of A^r. By the inductive hypothesis, b_{ik} is the number of paths of length r from v_i to v_k.

A path of length $r+1$ from v_i to v_j is made up of a path of length r from v_i to some intermediate vertex v_k, and an edge from v_k to v_j. By the product rule for counting, the number of such paths is the product of the number of paths of length r from v_i to v_k, namely, b_{ik}, and the number of edges from v_k to v_j, namely, a_{kj}. When these products are added for all possible intermediate vertices v_k, the desired result follows by the sum rule for counting.

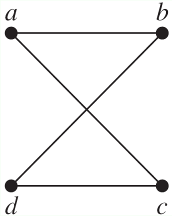
Example

How many paths of length four are there from a to d in the simple graph G in following figure?



Example

How many paths of length four are there from a to d in the simple graph G in following figure?

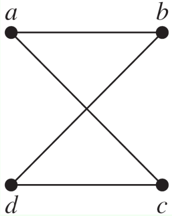


The adjacency matrix of G (ordering the vertices as a, b, c, d) is

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, A^4 = \begin{pmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{pmatrix}.$$

Example

How many paths of length four are there from a to d in the simple graph G in following figure?



The adjacency matrix of G (ordering the vertices as a, b, c, d) is

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, A^4 = \begin{pmatrix} 8 & 0 & 0 & 8 \\ 0 & 8 & 8 & 0 \\ 0 & 8 & 8 & 0 \\ 8 & 0 & 0 & 8 \end{pmatrix}.$$

Hence, the number of paths of length four from a to d is the $(1,4)-$th entry of A^4.

Take-aways

Conclusions

- Representing Graphs
  - Adjacency Matrices
  - Incidence Matrices
  - Random walk and Laplacian
- isomorphism of Graphs
- Connectivity
  - Vertex Connectivity
  - Edge Connectivity
- Connectivity
  - Paths
  - Connectedness in undirected graphs
  - How Connected is a Graph?
  - Connectedness in Directed Graphs
  - Paths and Isomorphism
  - Counting Paths Between Vertices