Graphs & Trees (Lecture – 3)

Dr. Nirnay Ghosh

Graph Representation (*Contd…*)

- **<u>Adjacency matrix for directed graph</u>**:
  - The matrix for a directed graph $G = (V, E)$ has a 1 in its (i, j)th position if there is an edge from v_i to v_j, where $v_1, v_2, \ldots, v_n$ is an arbitrary listing of the vertices of the directed graph.
  - The adjacency matrix for a directed graph does not have to be symmetric, because there may not be an edge from v_j to v_i when there is an edge from v_i to v_j
  - If $\mathbf{A} = [aij]$ is the adjacency matrix for the directed graph with respect to this listing of the vertices, then

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

  - <u>Trade-off between Adjacency lists and Adjacency matrices:</u>
    - Sparse graph: adjacency list
    - Dense graph: adjacency matrix

- **<u>Incidence matrix:</u>** Let $G = (V, E)$ be an undirected graph. Suppose that $v_1, v_2, \ldots, v_n$ are the vertices and $e_1, e_2, \ldots, e_m$ 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

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

Isomorphism of Graphs

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*.* Two simple graphs that are not isomorphic are called *nonisomorphic*.

- *-The word isomorphism comes from the Greek roots *isos* for "equal" and *morphe* for "form".
- If two simple graphs are isomorphic, there is a one-to-one correspondence between vertices of the two graphs that preserves the adjacency relationship.

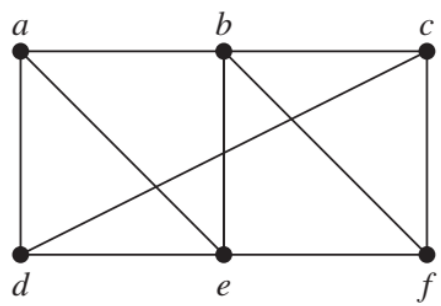
- Determining whether two simple graphs are isomorphic?
  - Hard problem, as there can be *n!* possible one-to-one correspondence between two simple graphs with *n* vertices
  - Easier to show if the graphs are *not* isomorphic by checking the following attributes of the *graph invariant* property:
    - Same number of vertices
    - Same number of edges
    - Degrees of the vertices must be the same

Connectivity

- Path:

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 $e_1, \ldots, e_n$ 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, for $i = 1, \ldots, n$, the endpoints x_{i-1} and x_i. When the graph is simple, we denote this path by its vertex sequence $x_0, x_1, \ldots, x_n$ (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 $x_1, x_2, \ldots, x_{n-1}$ or *traverse* the edges $e_1, e_2, \ldots, e_n$. A path or circuit is *simple* if it does not contain the same edge more than once.

- Informally, a path is a sequence of edges that begins at a vertex of a graph and travels from vertex to vertex along edges of the graph.
- As the path travels along its edges, it visits the vertices along this path, that is, the endpoints of these edges.

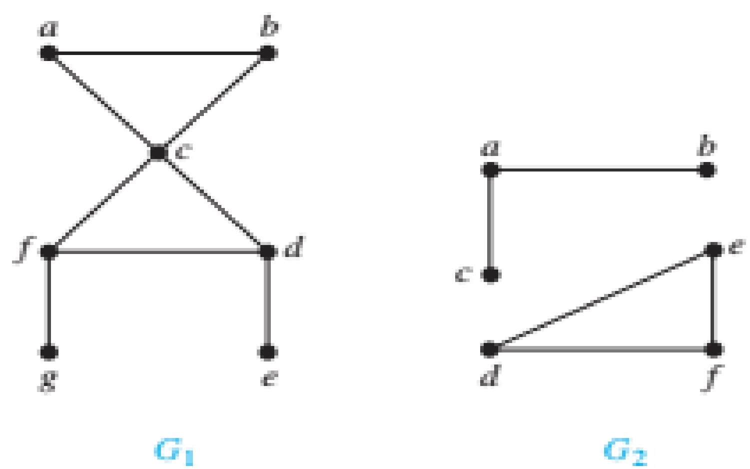


A Simple Graph

Connectivity (*Contd…*)

- Connectedness in Undirected Graphs

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*. We say that we *disconnect* a graph when we remove vertices or edges, or both, to produce a disconnected subgraph.



Connected and Disconnected Graph

- Theorem:

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

Connectivity (*Contd…*)

- Connected component
  - 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.
  - It is a maximal 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.

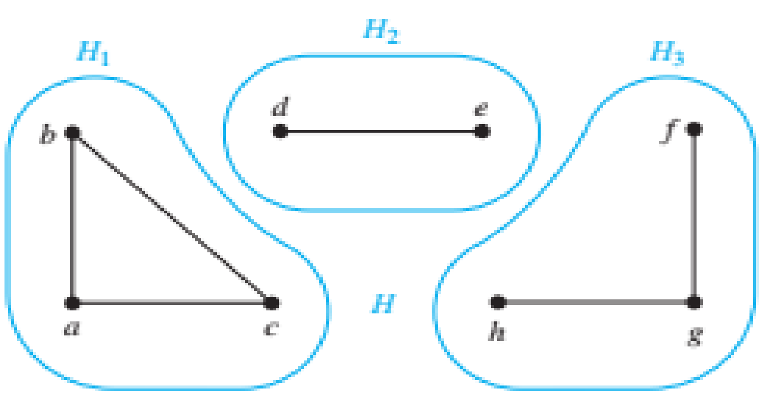


FIGURE 3 **The Graph H and Its Connected Components H_1, H_2, and H_3.**

Connectivity (*Contd…*)

- How connected in a graph?
  - **Cut vertex**: removal of a cut vertex from a connected graph produces a subgraph that is not connected.
  - **Cut edge or bridge**: an edge whose removal produces a graph with more connected components than in the original graph.
    - Example: In a computer network, a cut vertex is an essential router and cut edge is an essential link that cannot fail for a computer network to be operational.
- **Vertex cut**: A subset *V'* of the vertex set *V* of $G = (V, E)$ is a *vertex cut, or separating set*, if $G - V'$ is disconnected.
  - For complete graph, no minimum vertex cut exist.
  - ***Vertex connectivity*** of a noncomplete graph *G*, denoted by $\kappa(G)$, is the minimum number of vertices in a ***vertex cut***