UNIVERSITY OF REGINA

CS310-002 DISCRETE COMPUTATIONAL STRUCTURES

ANDRÉ E. DOS SANTOS
dossantos@cs.uregina.ca
andreeds.github.io

CS310-002

DISCRETE COMPUTATIONAL STRUCTURES

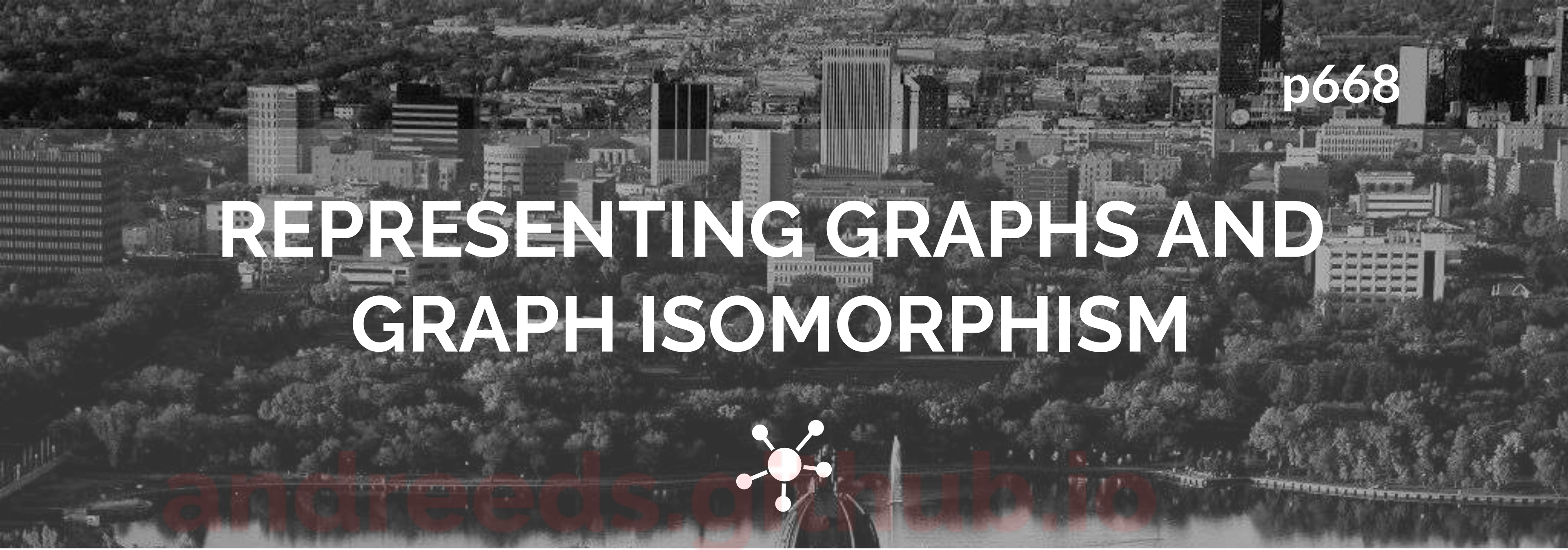
GRAPHS

ANDRÉ E. **DOS SANTOS**

dossantos@cs.uregina.ca

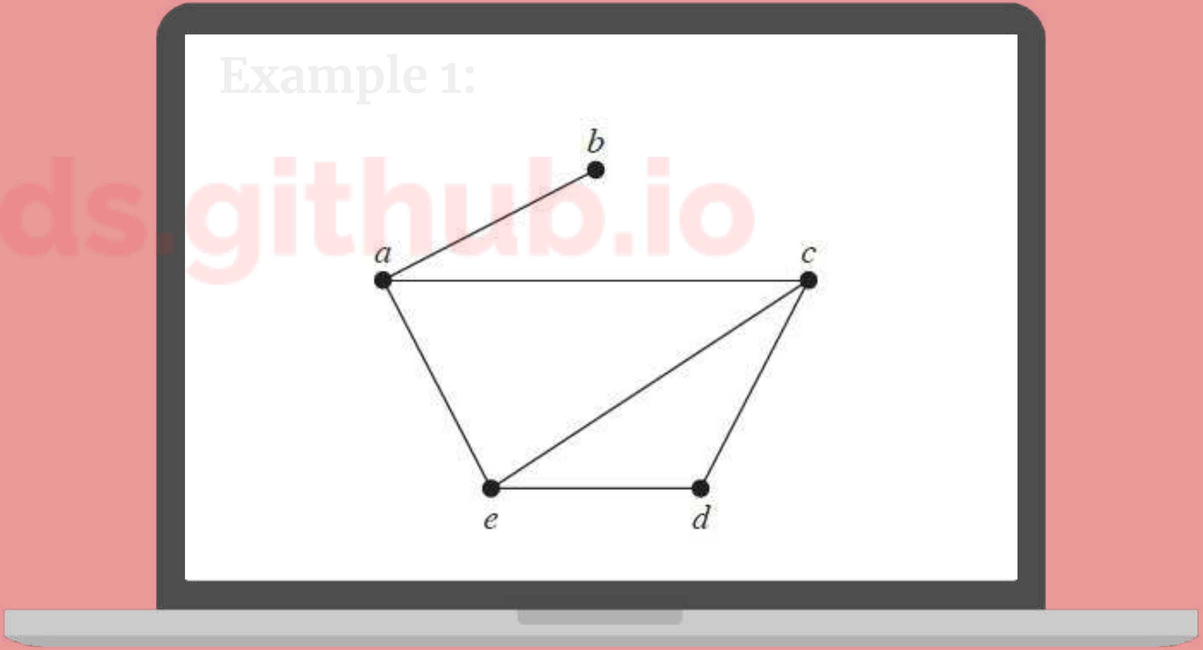
andreeds.github.io

REPRESENTING GRAPHS AND GRAPH ISOMORPHISM



REPRESENTING GRAPHS

- **Adjacency lists**
- Adjacency matrices
- Incidence matrices

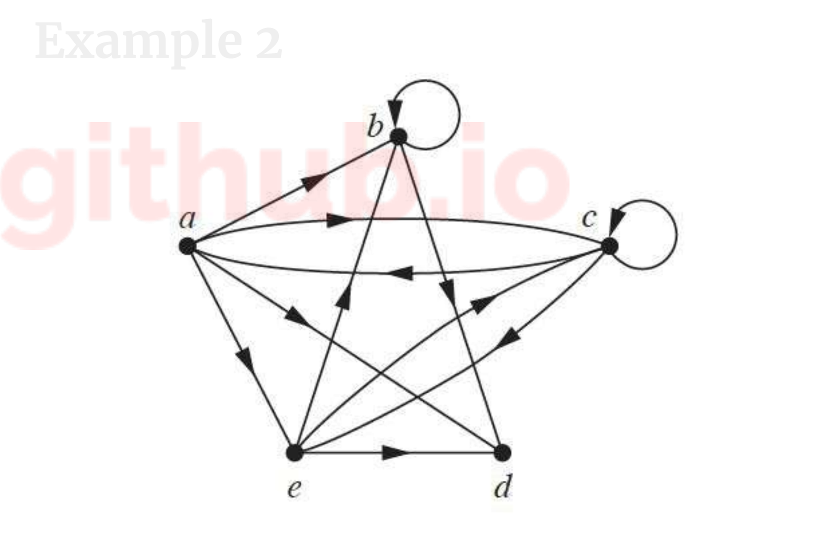


REPRESENTING GRAPHS

Adjacency lists

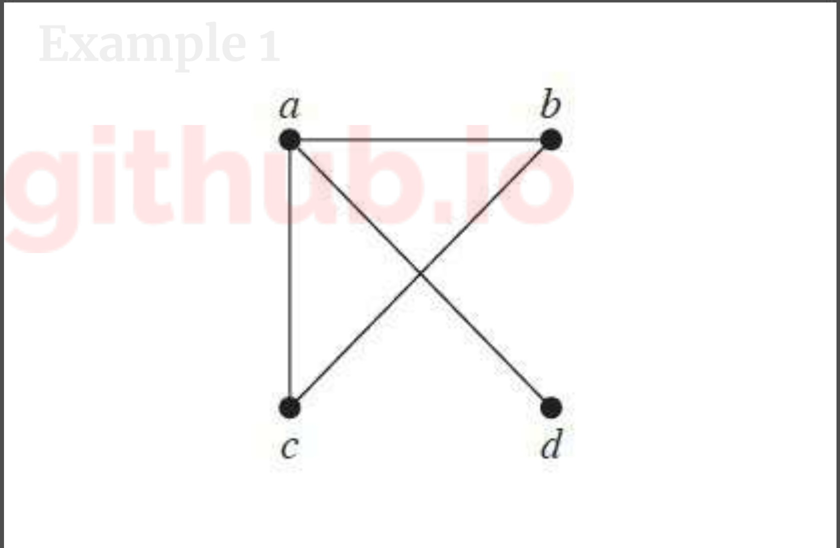
Adjacency matrices

Incidence matrices



REPRESENTING GRAPHS

- Adjacency lists
- **Adjacency matrices**
- Incidence matrices



REPRESENTING GRAPHS

Adjacency lists

Adjacency matrices

Incidence matrices

Example 2

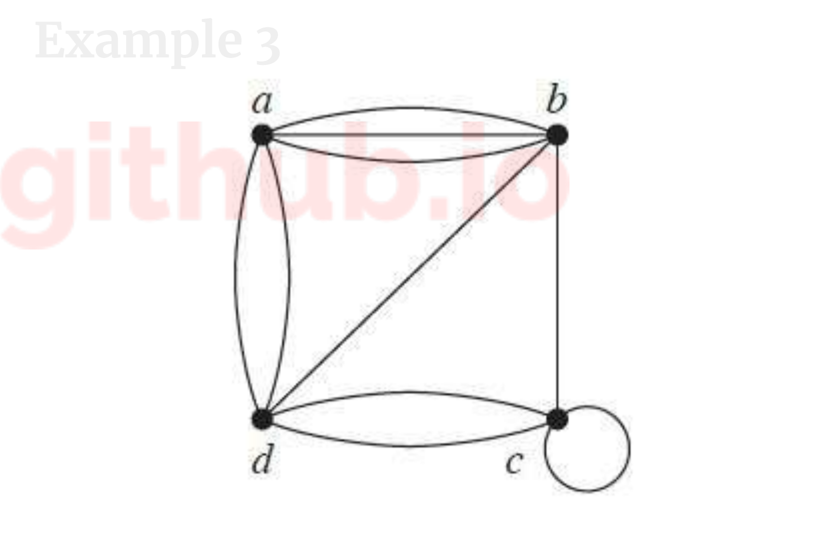
$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

REPRESENTING GRAPHS

Adjacency lists

Adjacency matrices

Incidence matrices

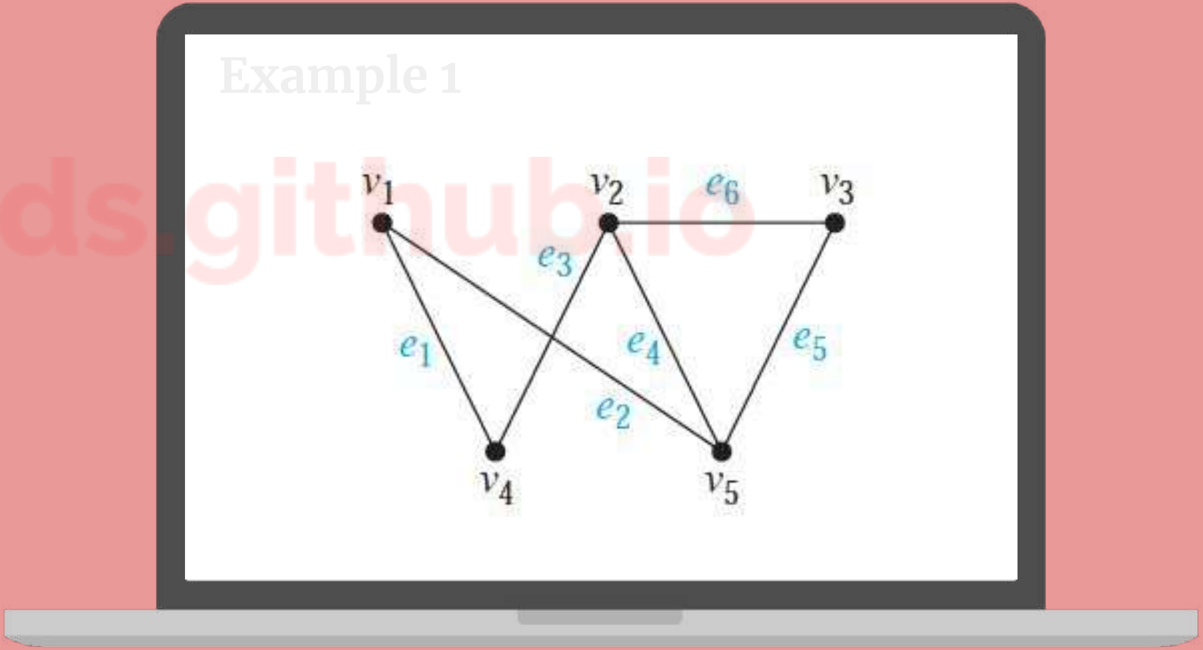


REPRESENTING GRAPHS

Adjacency lists

Adjacency matrices

Incidence matrices

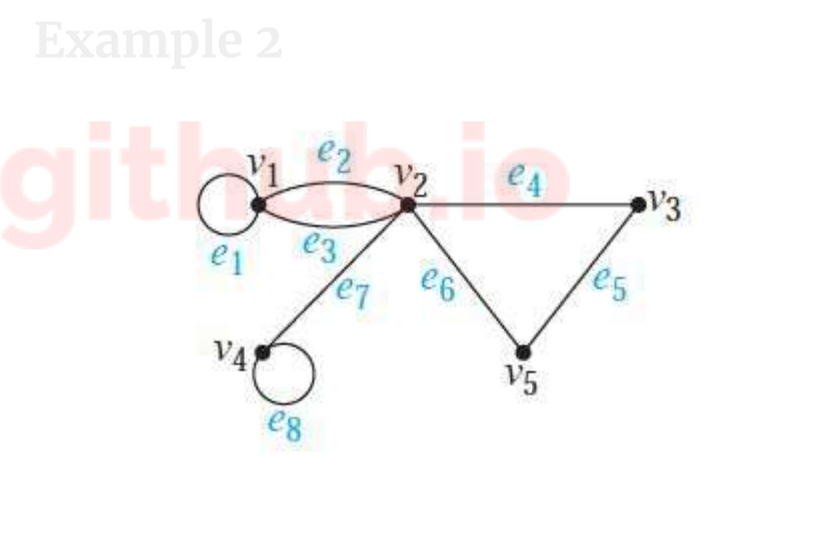


REPRESENTING GRAPHS

Adjacency lists

Adjacency matrices

Incidence matrices



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

Example

Show that the graphs $G = (V, E)$ and $H = (W, F)$, displayed in Figure 8, are isomorphic.

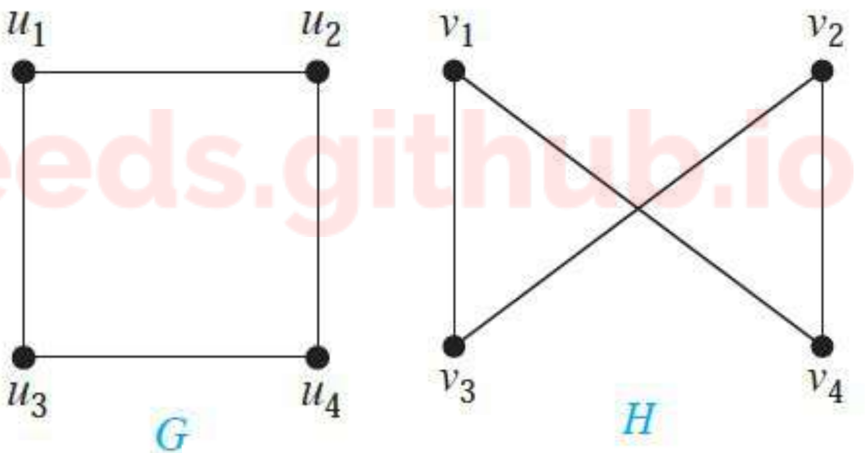


FIGURE 8 The Graphs G and H.

Example

Show that the graphs displayed in Figure 9 are not isomorphic.

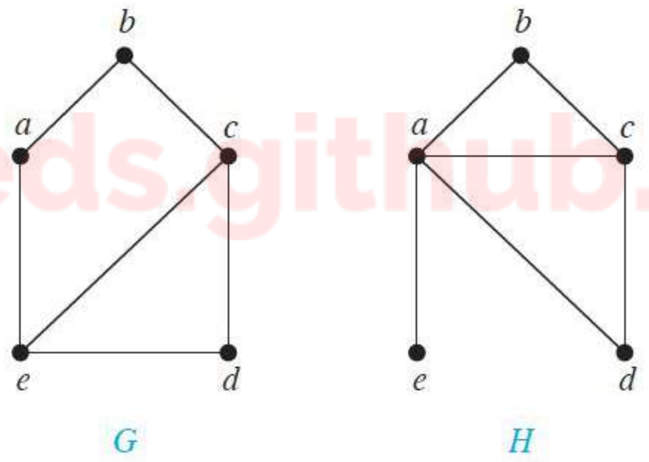


FIGURE 9 **The Graphs G and H.**

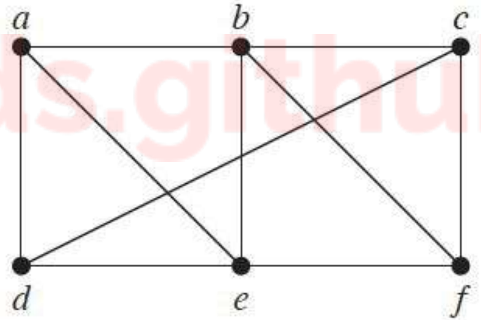
CONNECTIVITY

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 $e_1, \ldots, e_n$ 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.

Example

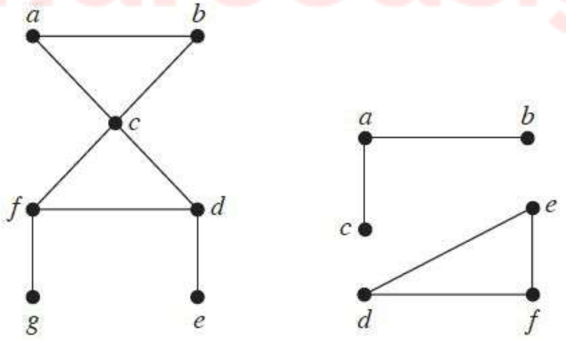
Is **a, d, c, f , e** a simple path? Or **d, e, c, a**? Or **b, c, f , e, b**? Or **a, b, e, d, a, b**?



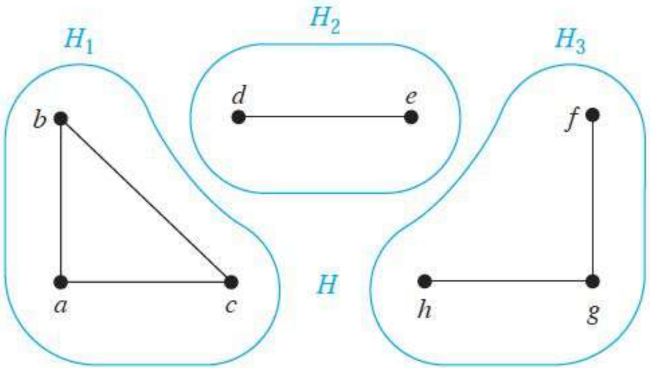
Definition

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

Examples

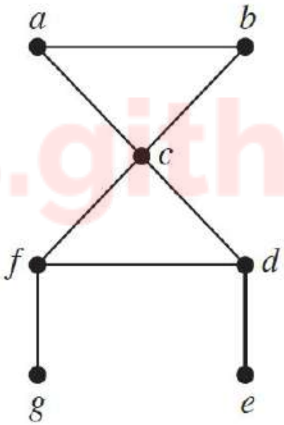


G_1 G_2

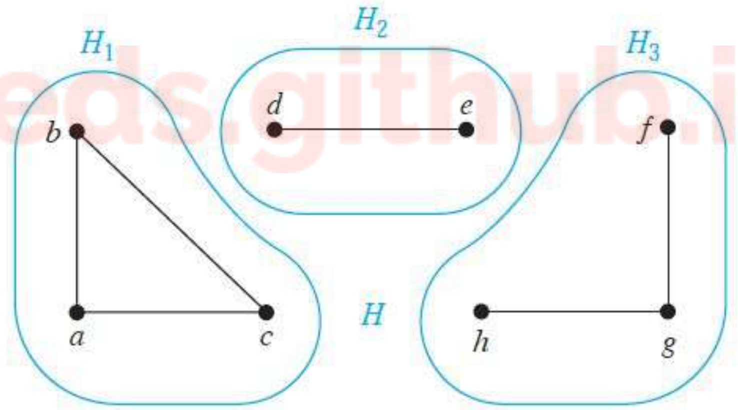


Theorem

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



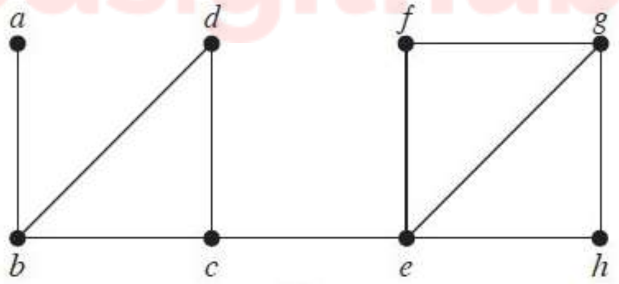
A **connected component** of a graph G is a connected subgraph of G that is not a proper subgraph of another connected subgraph of G. That is, a connected component of a graph G is a maximal connected subgraph of G.



A vertex whose removal produces a graph with more connected components than in the original graph is called a **cut vertex** (or **articulation point**).

Example

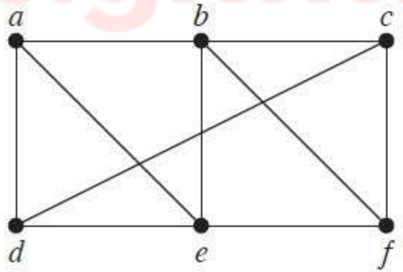
Find the cut vertices



A subset V' of the vertex set V of a graph $G = (V, E)$ is a **vertex cut**, or **separating set**, if $G - V'$ is disconnected.

Examples

Show that the subset $V'_1 = \{b, c, e\}$ is a vertex cut.



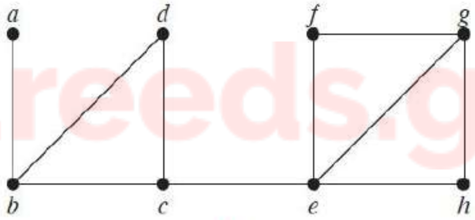
VERTEX CONNECTIVITY

The **vertex connectivity** of a noncomplete graph G, denoted by $k(G)$, is the *minimum number of vertices in a vertex cut.*

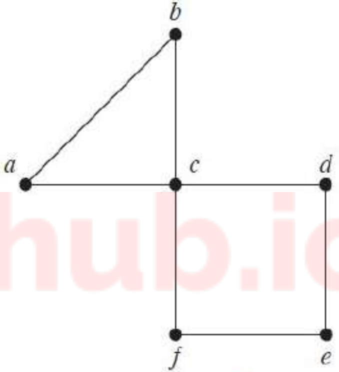
For a complete graph K_n, we denote $k(K_n) = n - 1$.

Example

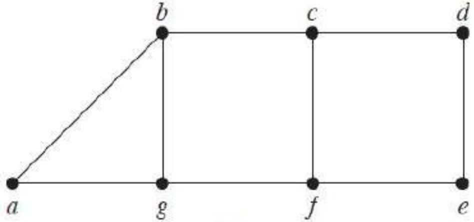
Find the **vertex** connectivity for each of the following graphs



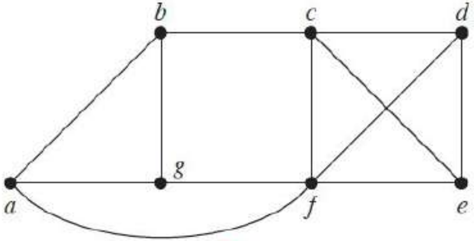
G_1



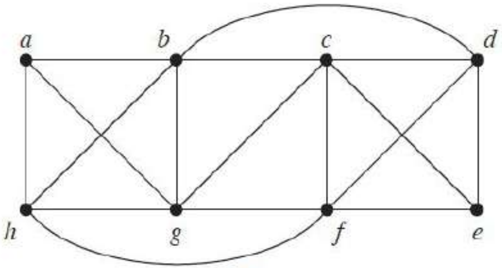
G_2



G_3



G_4



G_5

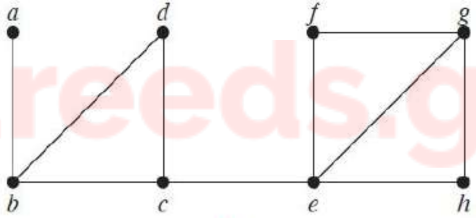
VERTEX CONNECTIVITY

A graph is ***k*-connected** (or ***k*-vertex-connected**), if $k(G) \geq k$.

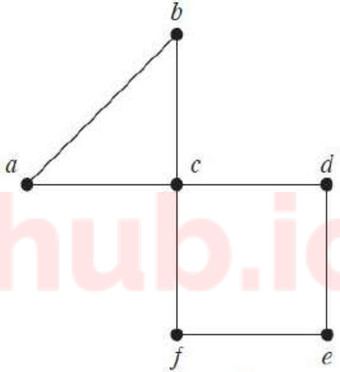
For any graph G with n vertices, we have $0 \leq k(G) \leq \mathrm{n} - 1$.

Example

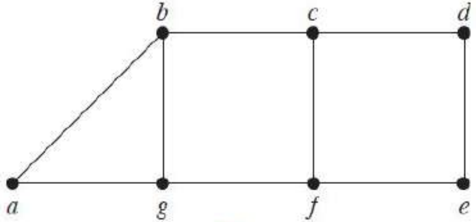
Describe which type of k-connected the following graphs are



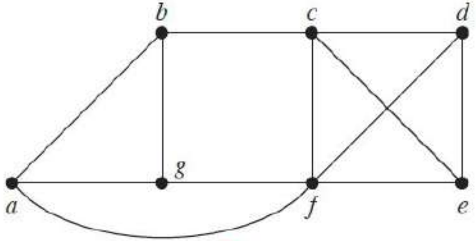
G_1



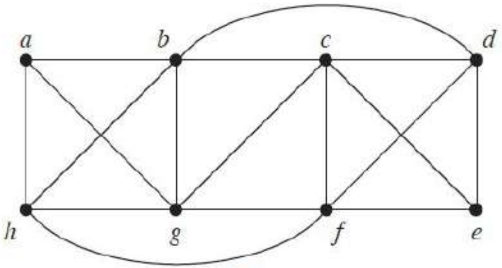
G_2



G_3



G_4



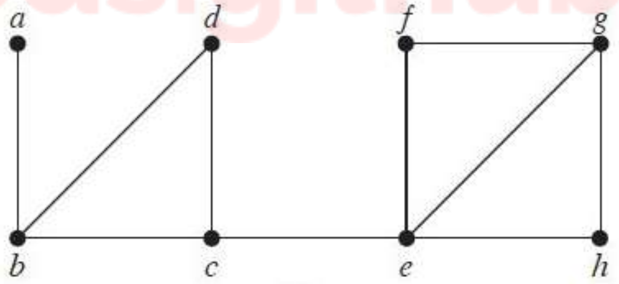
G_5

EDGE CONNECTIVITY

An edge whose removal produces a graph with more connected components than in the original graph is called a **cut edge** (or **bridge**).

Example

Find the cut edges

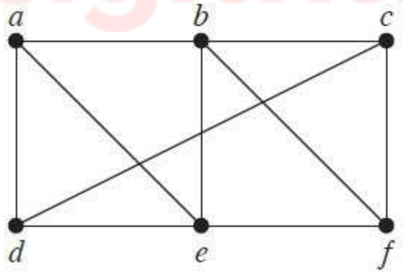


EDGE CONNECTIVITY

A subset E' of the vertex set E of a graph $G = (V, E)$ is a **edge cut** if G - E' is disconnected.

Example

Show that the subset $E'_1 = \{(b, c), (c, d), (b, f), (e, f)\}$ is a edge cut.



EDGE CONNECTIVITY

The **edge connectivity** of a graph G, denoted by $\lambda(G)$, is the *minimum number of edges in an edge cut of G.*

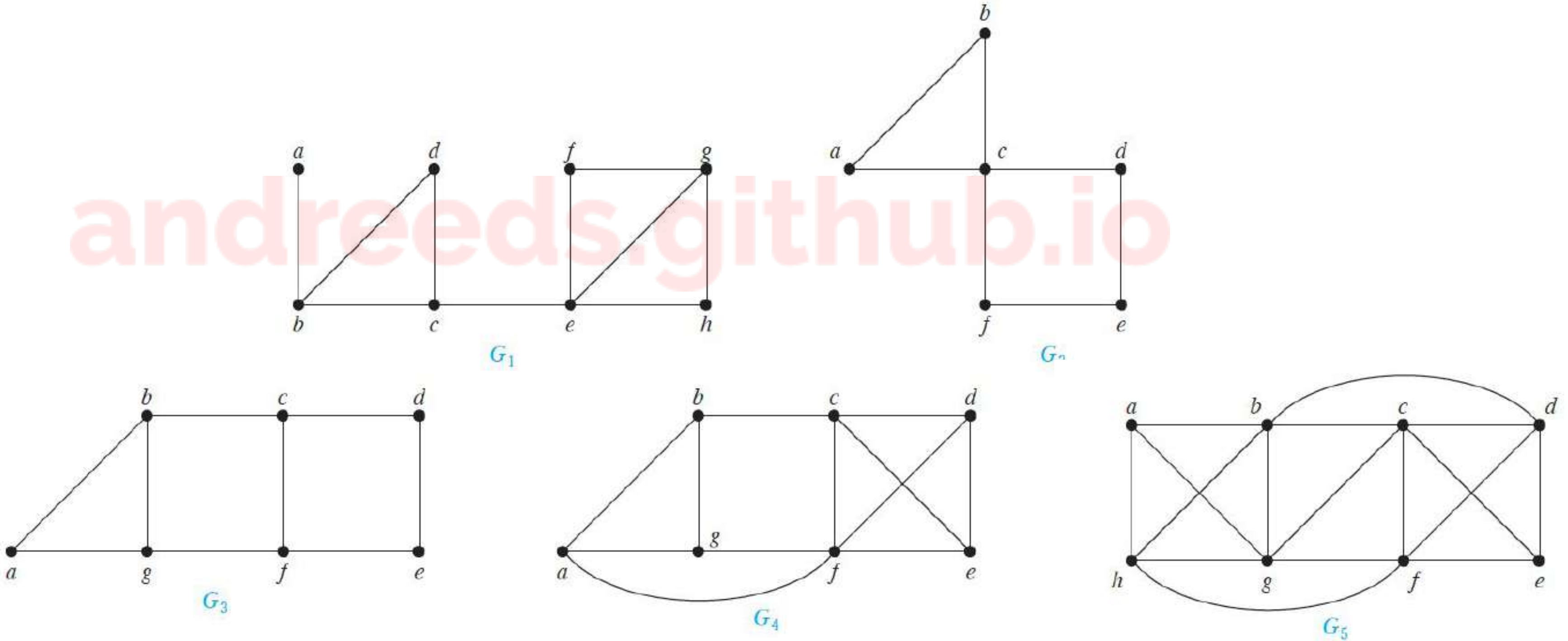
For any graph G with n vertices, $0 \leq \lambda(G) \leq n - 1$.

An Inequality for Vertex Connectivity and Edge Connectivity

$$k(G) \leq \lambda(G) \leq \min_{v \in V} \deg(v)$$

Example

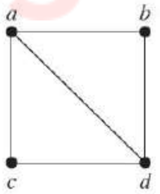
Find the **edge** connectivity for each of the following graphs



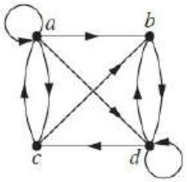
REVIEW QUESTIONS Pt. 3
GRAPHS

Use adjacency lists and adjacency matrices to represent the given graphs.

1.



3.

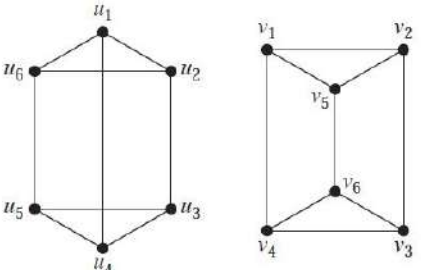


REVIEW QUESTIONS Pt. 3
GRAPHS

- Draw the graph represented by the given adjacency matrix.

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

- Determine whether the given pair of graphs is isomorphic. Exhibit an isomorphism or provide a rigorous argument that none exists.



- Construct a graph G with $\kappa(G) = 1$, $\lambda(G) = 2$, and $\min_{v \in V} \deg(v) = 3$.