CHAPTER 4

(Part 1)

GRAPH THEORY

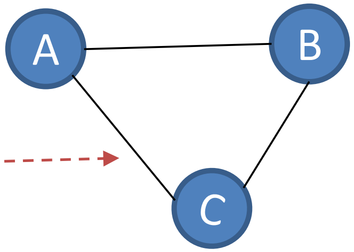
Definition of Graph

- A graph G consists of two finite sets:
  - A nonempty set $V(G)$ of vertices.
  - A set $E(G)$ of edges, where each edge is associated with a set consisting of either one or two vertices called its endpoints.
  - f is a function, called an incidence function, that assign to each edge, $e \in E$, a one element subset $\{v\}$ or two elements subset $\{v, w\}$, where v and w are vertices.
- We can write G as (V, E, f) or (V, E) or simply as G.

Definition of Graph (cont'd)

Pictorial representation of graph:

- **Vertex**: Dot
- **Edge**: line



Example

- Let,
  - $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$
  - $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$
- And *f* be defined by:
  - $f(e_1) = f(e_2) = \{v_1, v_2\}$
  - $f(e_3) = \{v_4, v_3\}$
  - $f(e_4) = f(e_6) = f(e_6) = \{v_6, v_3\}$
  - $f(e_5) = \{v_2, v_4\}$

Question: What is the pictorial representation of *G*?

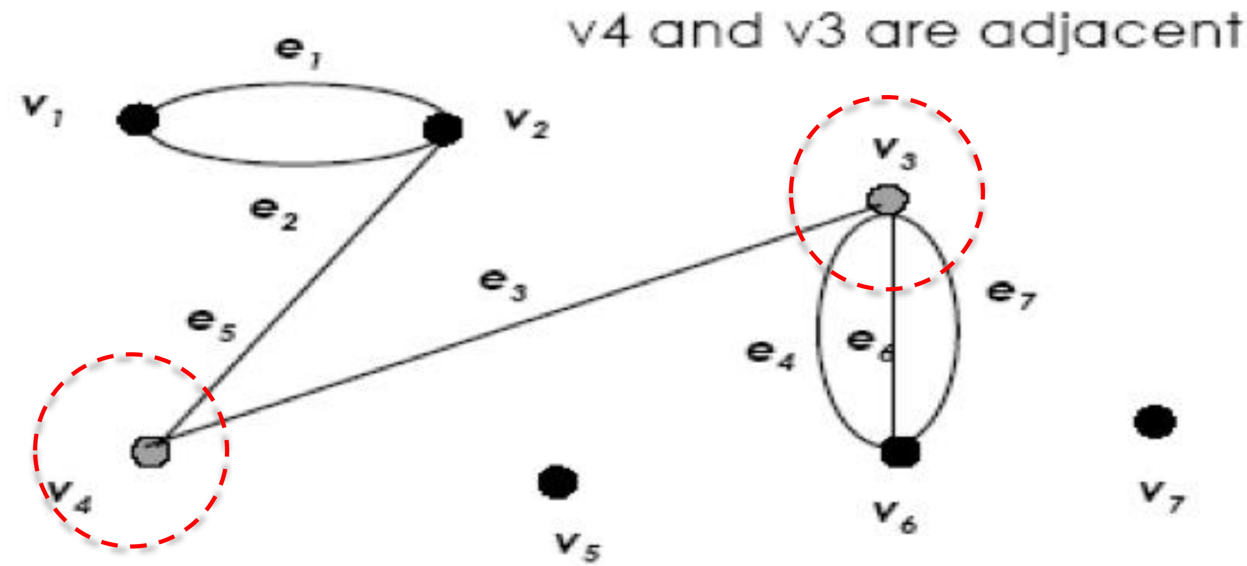
* Solution – refer module (Fig. 4.5), pg. 92

Characteristics of Graph

Adjacent Vertices

Two vertices that are connected by an edge are called adjacent; and a vertex that is an endpoint of a loop is said to be adjacent to itself.

Example

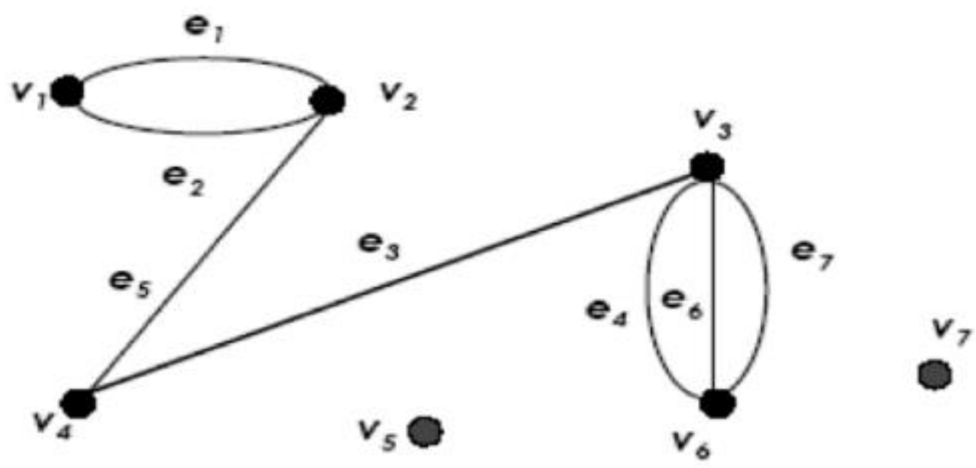


Incident Edge

An edge is said to be **incident** on each of its endpoints.

e_1 and e_2 are incident on v_1 and v_2

Example

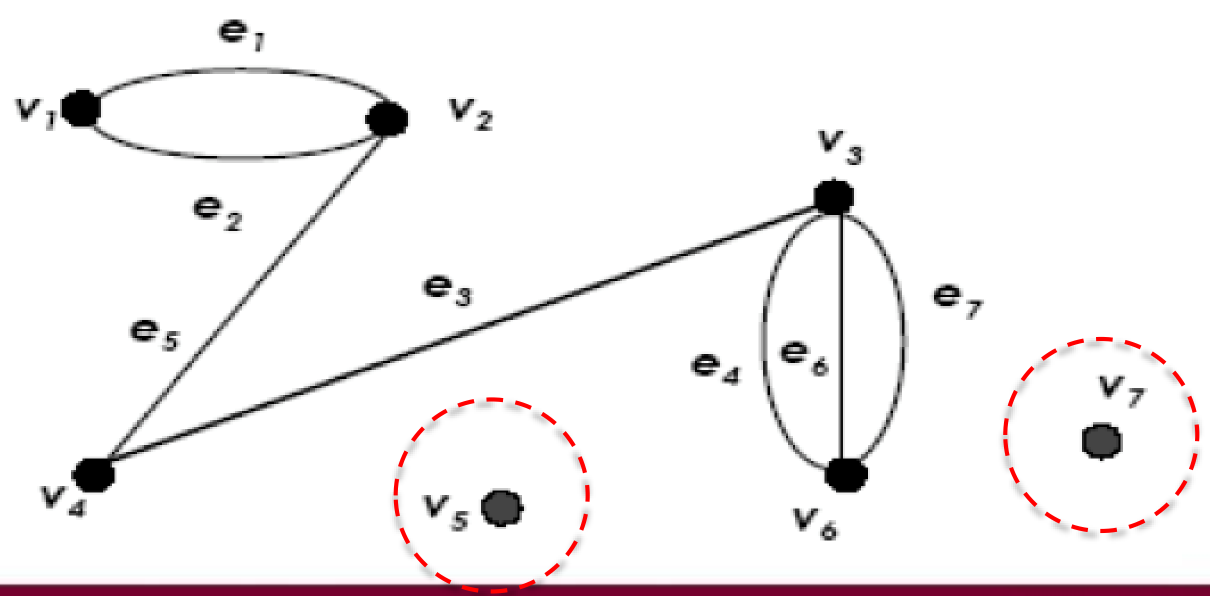


Isolated Vertex

Let G be a graph and *v* be a vertex in G. We say that *v* is an isolated vertex if it is not incident with any edge.

- v5 and v7 are isolated vertices.

Example

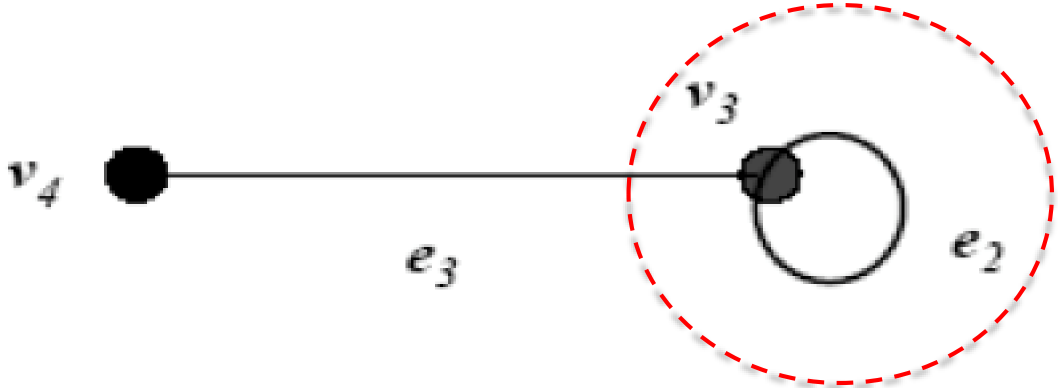


Characteristics of Graph (cont'd)

Loop

An edge with just one endpoint is called a loop.

Example



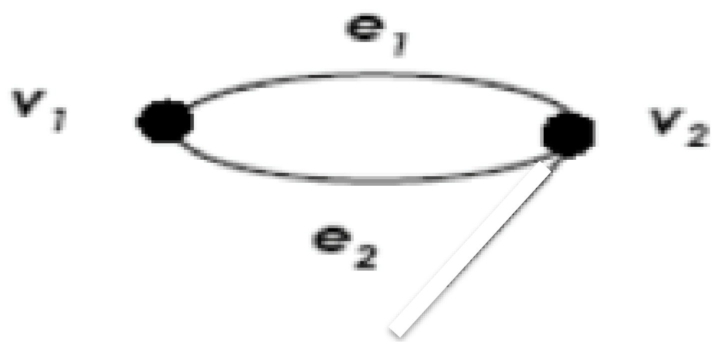
- e2 is a loop

Parallel Edges

Two or more distinct edges with the same set of endpoints are said to be parallel.

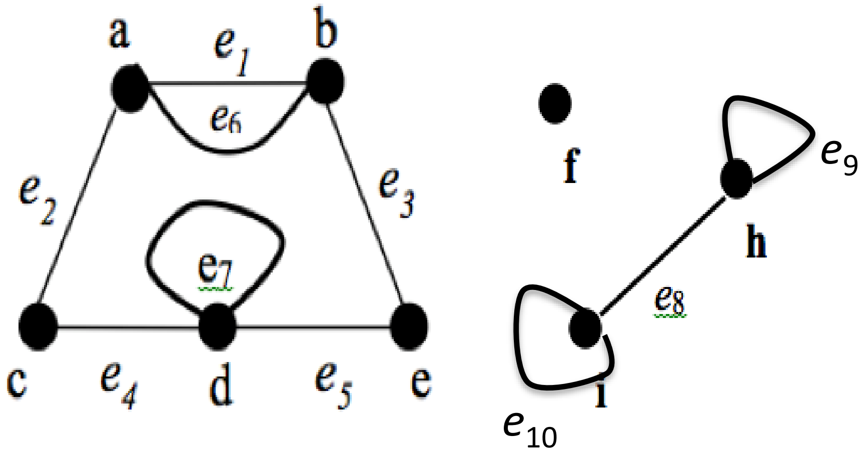
- e_1 and e_2 are parallel.

Example



Example

Given a graph as shown below,



a) Write a vertex set and the edge set, and give a table showing the edge-endpoint function.

a) Find all edges that are incident on **a**, all vertices that are adjacent to **a**, all edges that are adjacent to e_2, all loops, all parallel edges, all vertices that are adjacent to themselves and all isolated vertices.

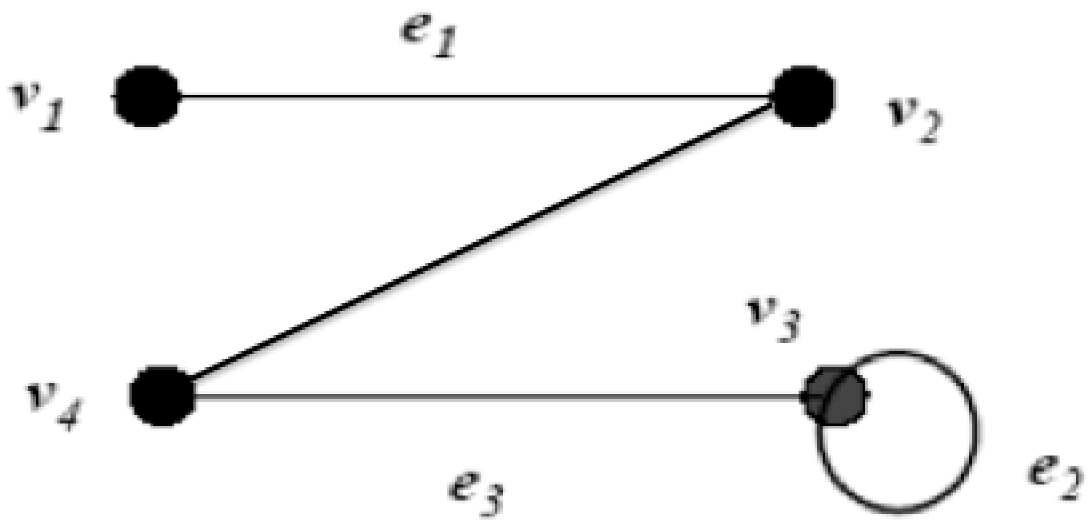
Note: Solution – Refer module, pg. 91-92

The Concept of Degree

- Let G be a graph and *v* be a vertex in G.
- The degree of *v*, written deg(*v*) or d(*v*) is the number of edges incident with *v*.
- Each **loop** on a vertex *v* contributes **2** to the degree of *v*.

Example

State the degree of each vertex for the following graph.

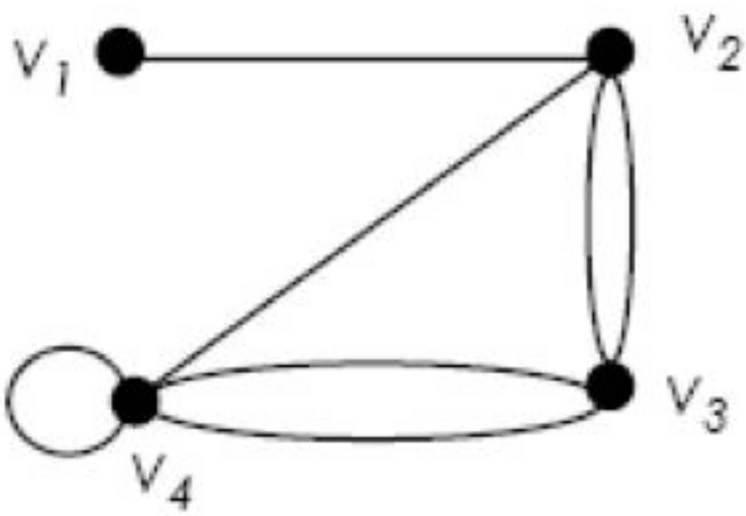


Answer:

$\deg(v_1) = 1$; $\deg(v_2) = 2$; $\deg(v_3) = 3$; $\deg(v_4) = 2$

Exercise#1

- Find the degree of each vertex in the graph.

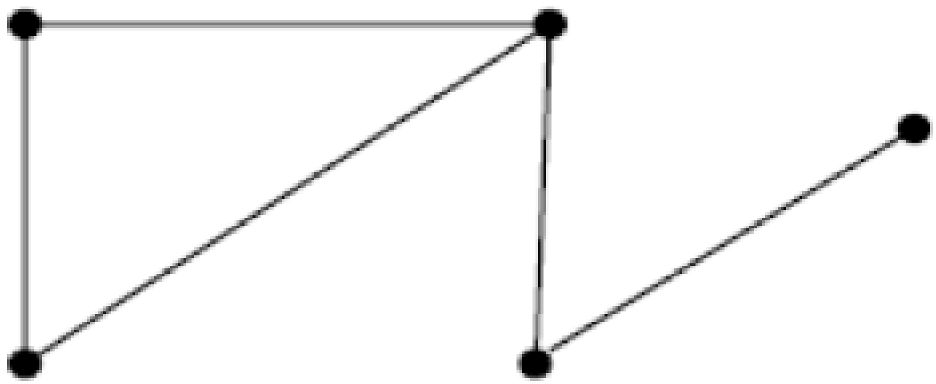


Types of Graphs

Simple Graph

A graph *G* is called a simple graph if *G* does not contain any parallel edges and any loops.

Example



Types of Graphs (cont'd)

Regular Graph

Let G be a graph and k be a nonnegative integer. G is called a k-regular graph if the degree of each vertex of G is k.

Example

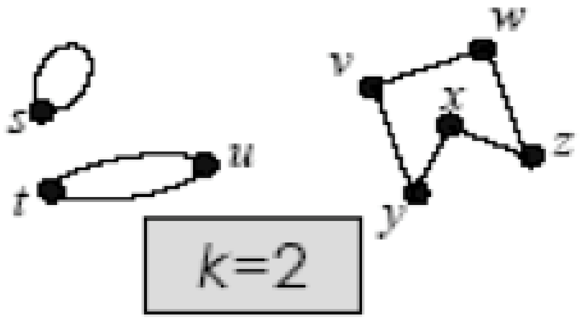


Fig.1: Graph A

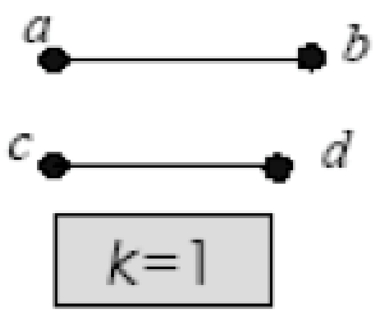


Fig.2: Graph B

Types of Graphs (cont'd)

Connected Graph

A graph G is connected if given any vertices *v* and *w* in G, there is a path from *v* to *w*.

Example

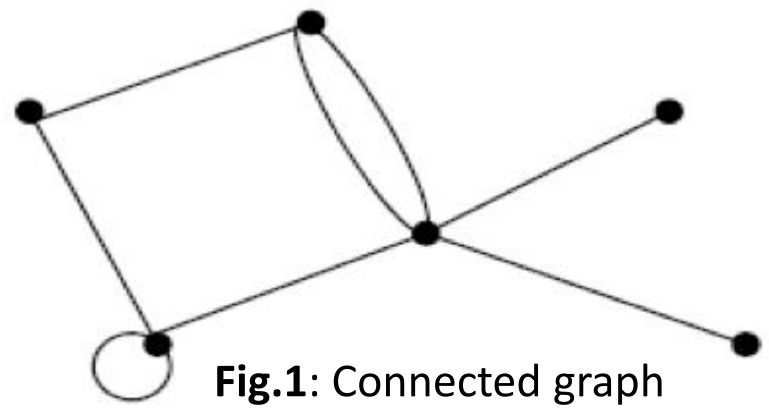


Fig.1: Connected graph

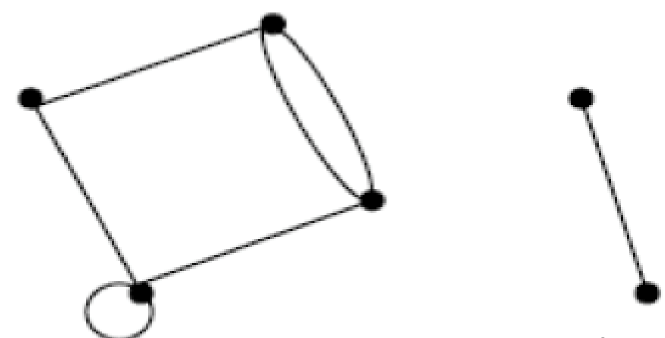


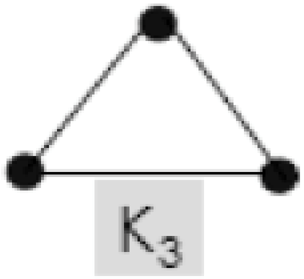
Fig.2: Not connected graph

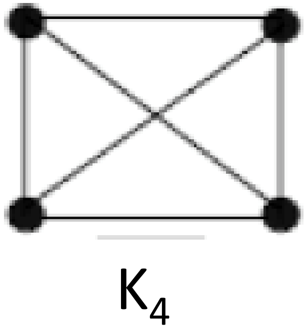
Types of Graphs (cont'd)

Complete Graph

A simple graph with n vertices in which there is an edge between every pair of distinct vertices is called a complete graph on n vertices. This is denoted by k_n .

Example

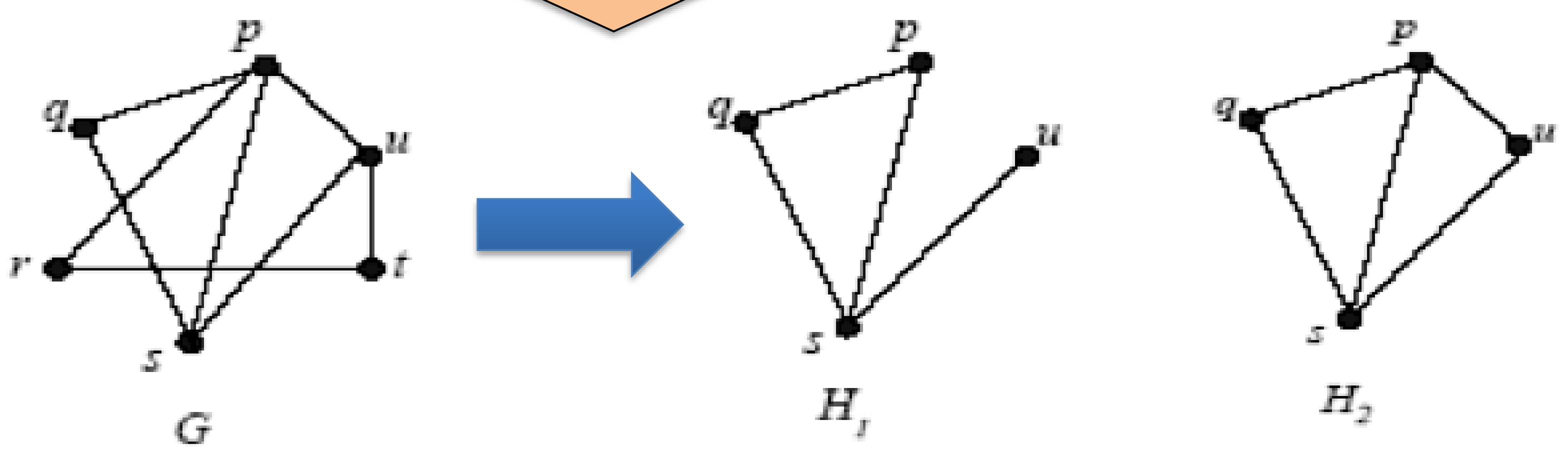




Subgraph

A graph H is said to be a subgraph of a graph G if, every vertex in H is also a vertex in G, every edge in H is also an edge in G, and every edge in H has the same endpoints as it has in G.

Example



Graph Representation

- To write programs that process and manipulate graphs, the graphs must be stored, that is, represented in computer memory.
- A graph can be represented (in computer memory) in several ways.
- 2-dimensional array: adjacency matrix and incidence matrix.

Graph Representation (cont'd)

Adjacency Matrix

- Let G be a graph with n vertices.

- The adjacency matrix, A_G is an $n \times n$ matrix $[a_{ij}]$ such that,

  a_{ij} = the number of edges from v_i to v_j, {undirected G}

  or,

  a_{ij} = the number of arrows from v_i to v_j, {directed G}

  for all $i, j = 1,2,\ldots\ldots, n$.

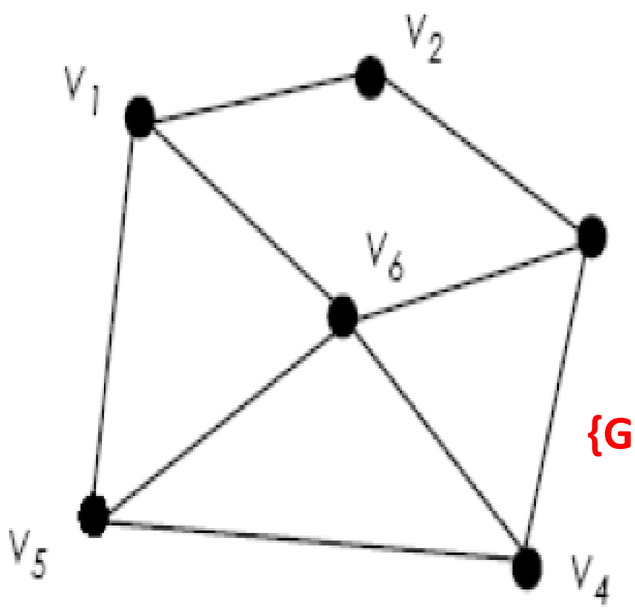
Graph Representation (cont'd)

- Adjacency matrix is a **symmetric matrix** if it is representing an undirected graph, where

$$a_{ij} = a_{ji}$$

- If the graph is directed graph, the presented matrix is not symmetrical.

Example



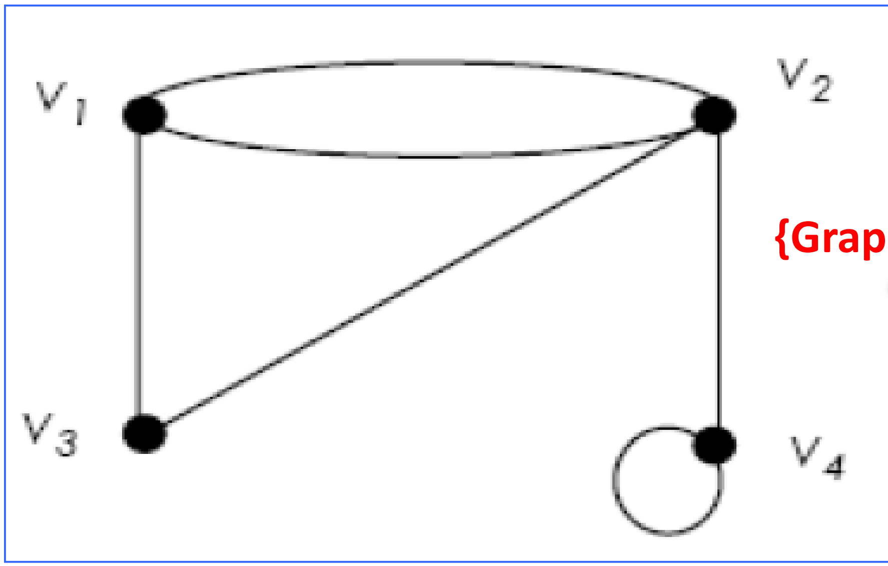
{Graph}

[Matrix]

$A_G =$

	V_1	V_2	V_3	V_4	V_5	V_6
V_1	0	1	0	0	1	1
V_2	1	0	1	0	0	0
V_3	0	1	0	1	0	1
V_4	0	0	1	0	1	1
V_5	1	0	0	1	0	1
V_6	1	0	1	1	1	0

Example



{Graph}

[Matrix]

$$A_G = \begin{array}{c} \\ \end{array}\begin{array}{cccc} V_1 & V_2 & V_3 & V_4 \end{array}$$

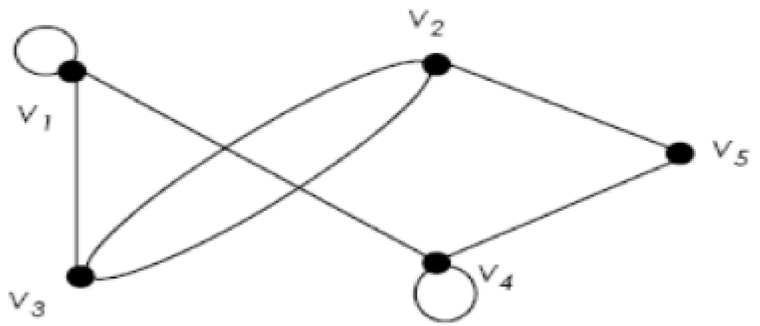
$$A_G = \begin{bmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$$

Example

Draw the graph based on the following matrix:

$$A_G = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Answer:



Graph Representation (cont'd)

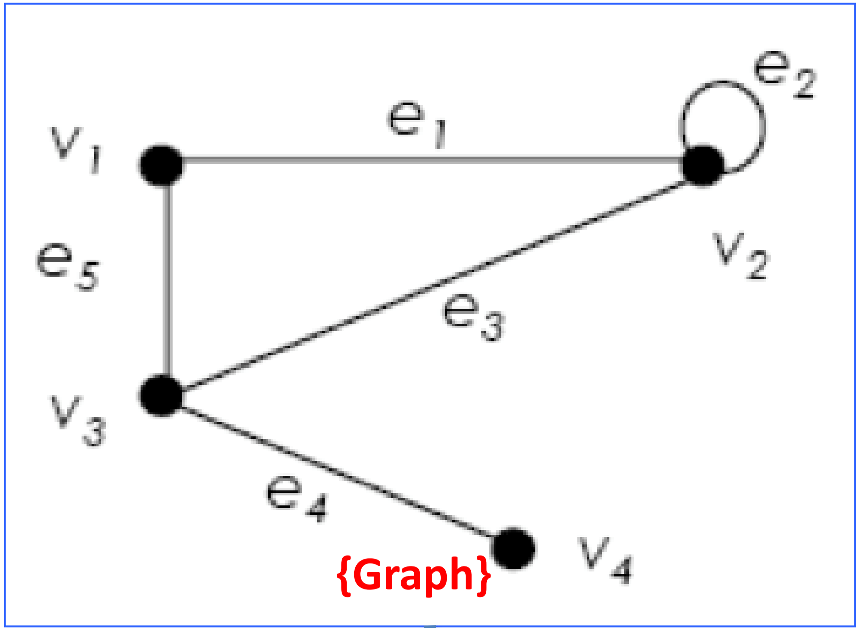
Incidence Matrix

- Let G be a graph with n vertices and m edges.

- The incidence matrix, I_G is an $n \times m$ matrix $[a_{ij}]$ such that,

$$a_{ij} = \begin{cases} 0 & \text{if } v_i \text{ is not an end vertex of } e_j, \\ 1 & \text{if } v_i \text{ is an end vertex of } e_j, \text{but } e_j \text{ is not a loop} \\ 2 & \text{if } e_j \text{ is a loop at } v_i \end{cases}$$

Example



[Matrix]

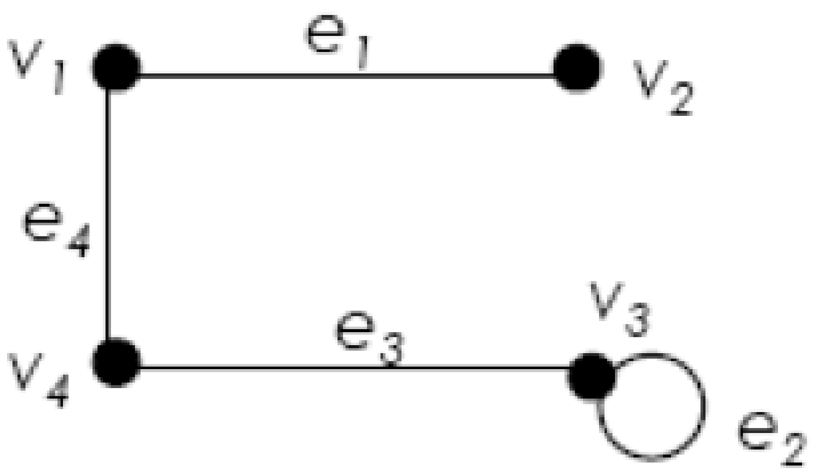
$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{ccccc} e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 1 & 2 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

deg(v_1)= 2;
deg(v_2)= 4;
deg(v_3)= 3;
deg(v_4)= 1

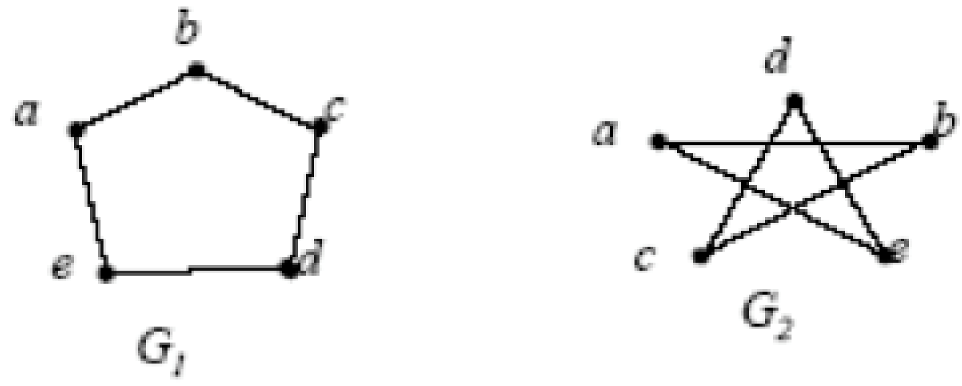
Notice that the sum of the *i*-th row is the degree of v_i.

Exercise #2

- Find the adjacency matrix and the incidence matrix of the graph.



Isomorphisms



- Are these two graphs (G_1 and G_2) are same?
- When we say that 2 graphs are the same mean they are isomorphic to each other.

Isomorphisms (cont'd)

Definition

Let $G = \{V, E\}$ and $G' = \{V', E'\}$ be graphs. G and G' are said to be isomorphic if there exist a pair of functions $f : V \to V'$ and $g : E \to E'$ such that f associates each element in V with exactly one element in V' and vice versa; g associates each element in E with exactly one element in E' and vice versa, and for each $v \in V$, and each $e \in E$, if v is an endpoint of the edge e, then $f(v)$ is an endpoint of the edge $g(e)$.

Isomorphisms (cont'd)

- If two graphs is isomorphic, they must have:
  - the same number of vertices and edges,
  - the same degrees for corresponding vertices,
  - the same number of connected components,
  - the same number of loops and parallel edges,
  - both graphs are connected or both graph are not connected,
  - pairs of connected vertices must have the corresponding pair of vertices connected.

- In general, it is easier to prove two graphs are not isomorphic by proving that one of the above properties fails.

Example

- Determine whether G is isomorphic to *H*.

Example 1 - Solution

- Both graphs are simple and have the same number of vertices and the same number of edges.

- All the vertices of both graphs have degree 2.

- Define $f : U \rightarrow V$, where $U = \{u_1, u_2, u_3, u_4\}$ and $V = \{v_1, v_2, v_3, v_4\}$,
  $f(u_1) = v_1$; $f(u_2) = v_4$; $f(u_3) = v_3$; $f(u_4) = v_2$.

Example 1 – Solution (cont'd)

- To verify whether G and H are isomorphic, we examine the adjacency matrix A_G with rows and columns labeled in the order u_1, u_2 , u_3, u_4 , and the adjacency matrix A_H with rows and columns labeled in the order v_1, v_2 , v_3, v_4 .

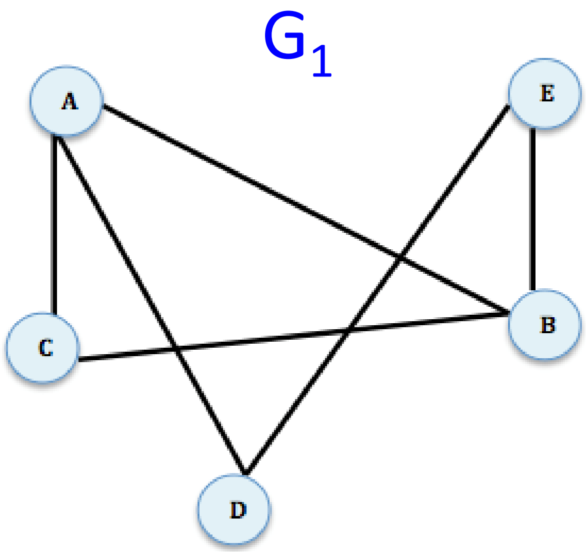
Example 1 – Solution (cont'd)

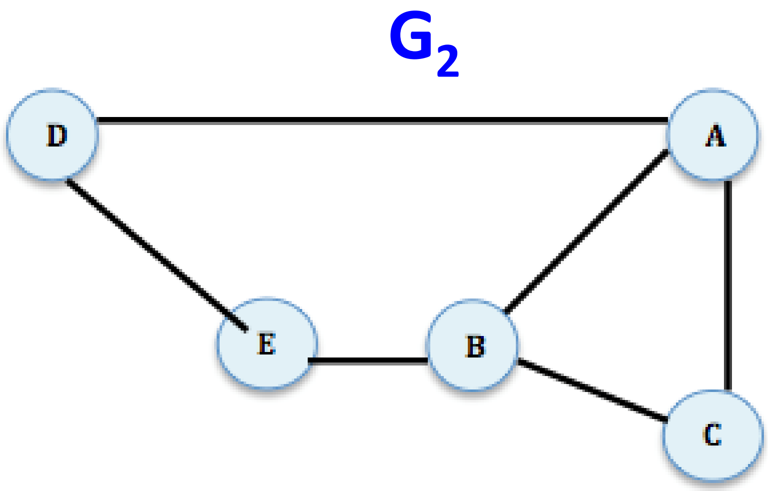
- A_G and A_H are the same, G and *H* are isomorphic.

$$A_G = \begin{array}{c} \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{array} \begin{array}{c} \begin{array}{cccc} u_1 & u_2 & u_3 & u_4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{array} \qquad A_H = \begin{array}{c} \\ v_1 \\ v_4 \\ v_3 \\ v_2 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_4 & v_3 & v_2 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Exercise # 3

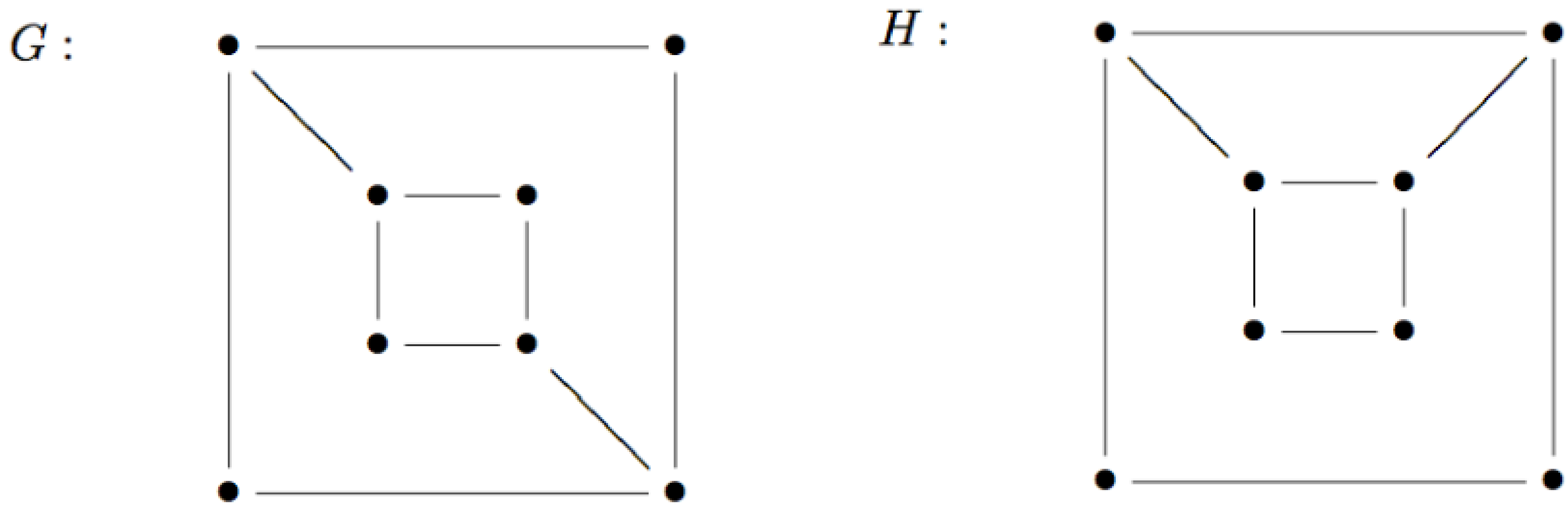
Show that the following two graphs are isomorphic.





Exercise # 4

Is these two graphs are isomorphic?



Trails, Paths & Circuits

Term and Description

- A walk from v to w is a finite alternating sequence of adjacent vertices and edges of G. Thus a walk has the form

$$(v_0 , e_1 , v_1 , e_2 , v_2 , \ldots\ldots , v_{n-1} , e_n , v_n)$$

where the v's represent vertices, the e's represent edges, $v = v_0$, $w = v_n$, and for i = 1,2,...,n. v_{i-1} and v_i are the endpoints of e_i .

- A trivial walk from v to w consist of the single vertex v.

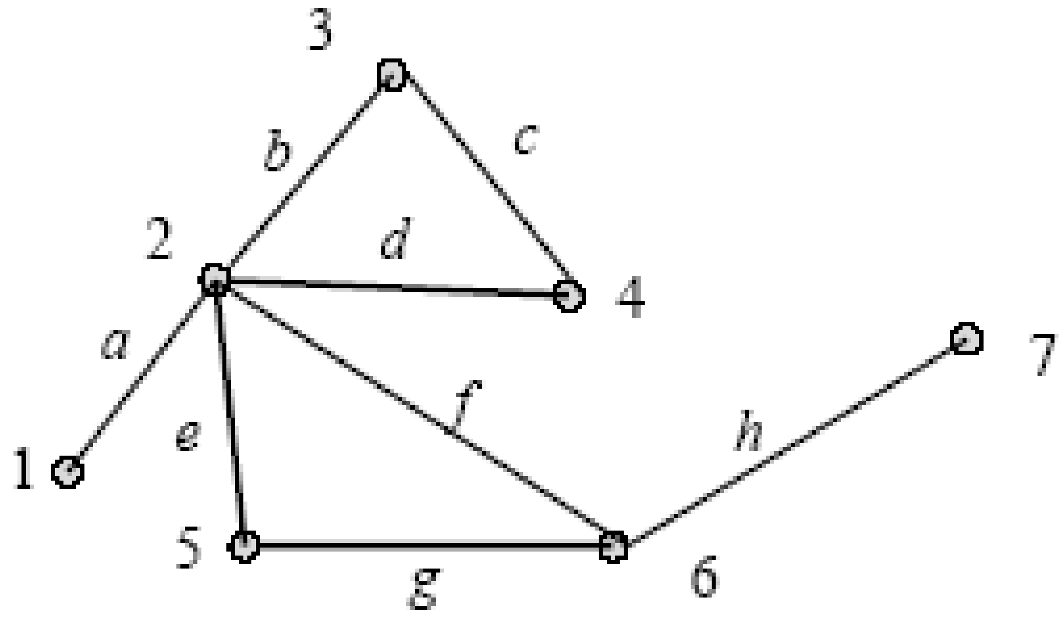
- The length of a walk is the number of edges it has.

Term and Description (cont'd)

- A trail from v to w is a walk from v to w that does not contain a repeated edge.
- A path from v to w is a trail from v to w that does not contain a repeated vertex.
- A closed walk is a walk that start and ends at the same vertex.
- A circuit/cycle is a closed walk that contains at least one edge and does not contain a repeated edge.
- A simple circuit is a circuit that does not have any other repeated vertex except the first and the last.

Example – Trail & Path

- (1, a, 2, b, 3, c, 4, d, 2, e, 5) is a trail.
- (6, g, 5, e, 2, d, 4) is a path.



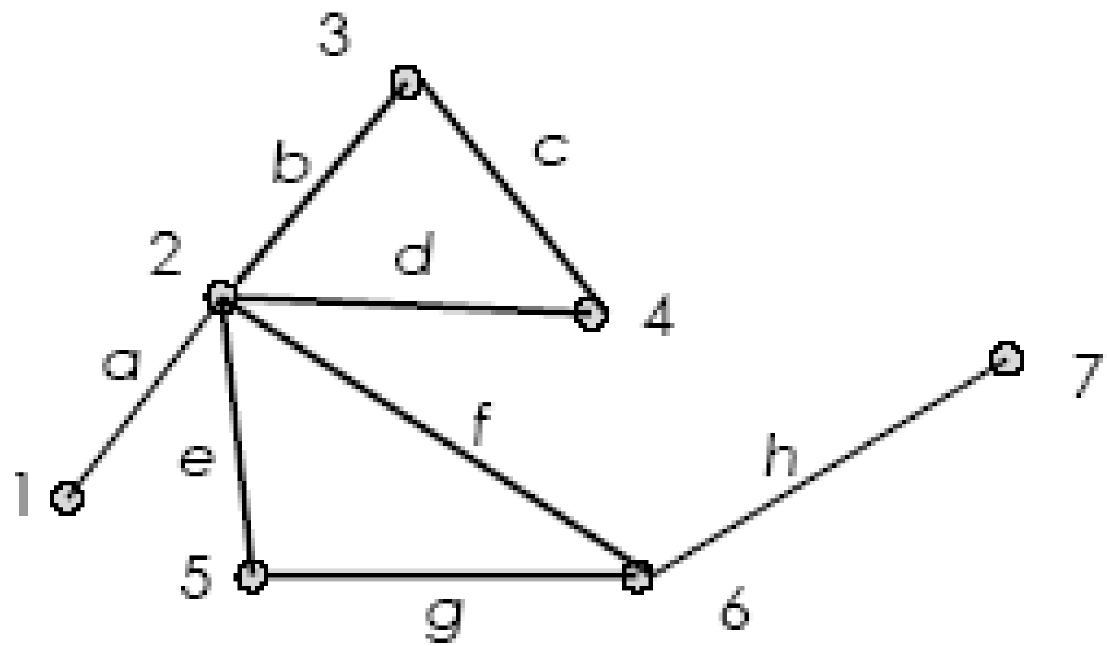
Note:

Trail: No repeated edge (can repeat vertex).

Path: No repeated vertex and edge.

Example – Cycle/circuit

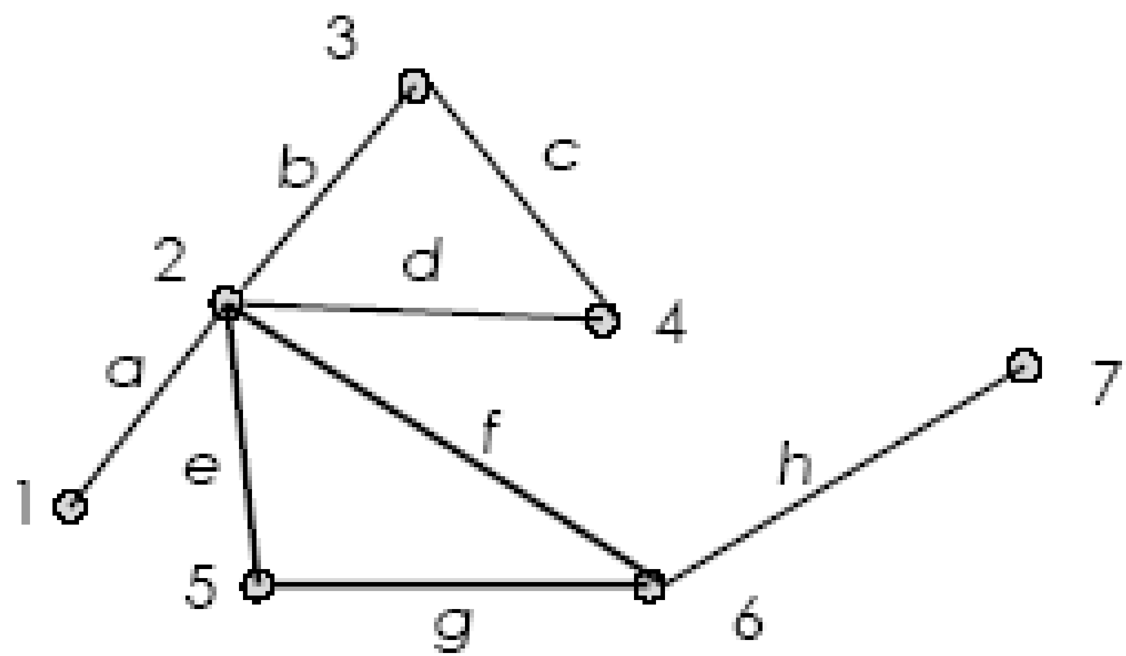
- (2, f, 6, g, 5, e, 2, d, 4, c, 3, b, 2) is a cycle.



Note: cycle –> start and end at same vertex, no repeated edge.

Example – Simple Cycle

- (5, g, 6, f, 2, e, 5) is a simple cycle.

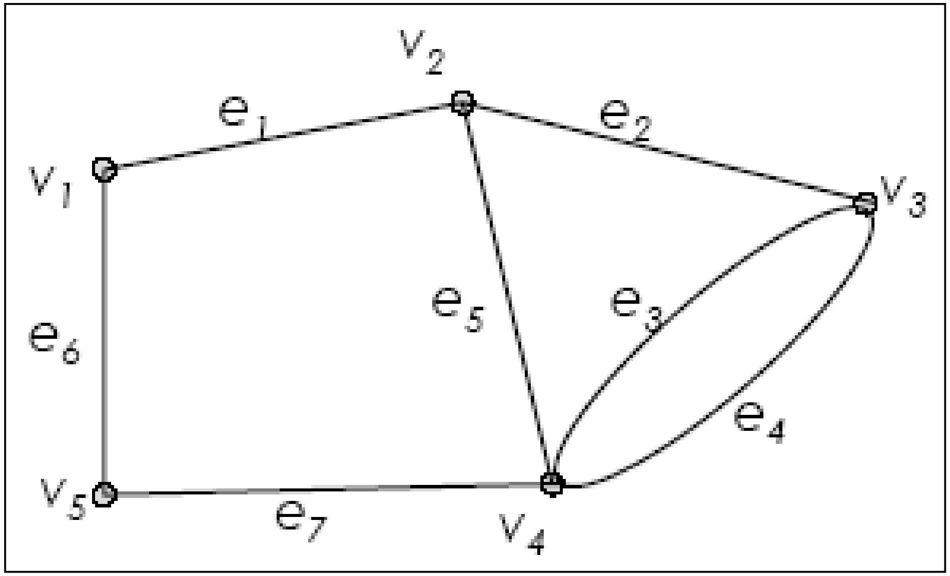


Note: Simple cycle –> start and end at same vertex, no repeated edge or vertex except for the start and end vertex.

Exercise # 5

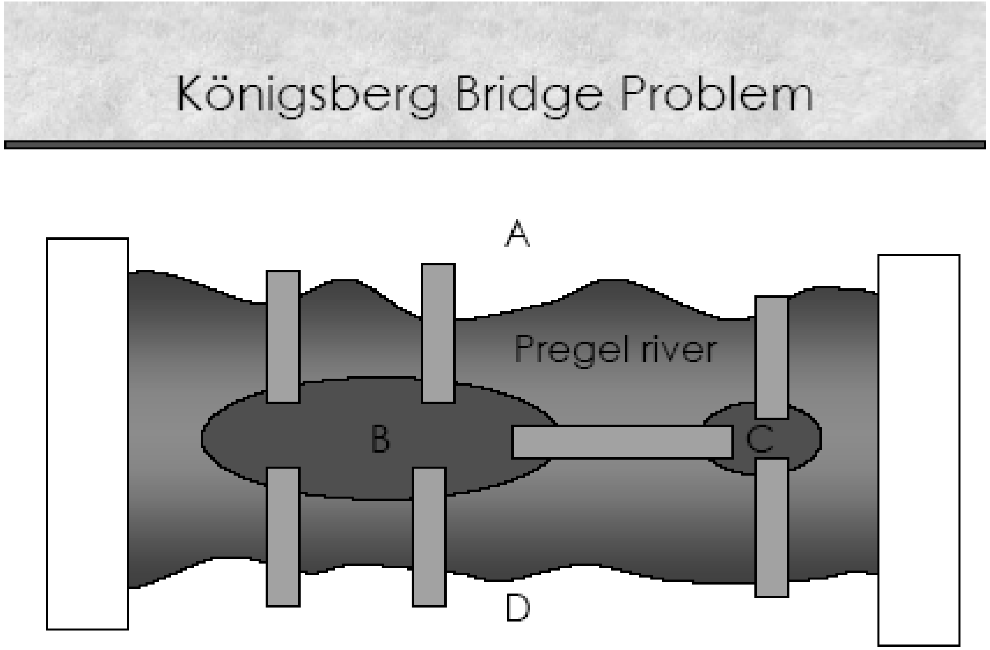
Tell whether the following is either a walk, trail, path, cycle, simple cycle, closed walk or none of these.

- (v_1, e_1, v_2)
- $(v_2, e_2, v_3, e_3, v_4, e_4, v_3)$
- $(v_4, e_7, v_5, e_6, v_1, e_1, v_2, e_2, v_3, e_3, v_4)$
- $(v_4, e_4, v_3, e_3, v_4, e_5, v_2, e_1, v_1, e_6, v_5, e_7, v_4)$



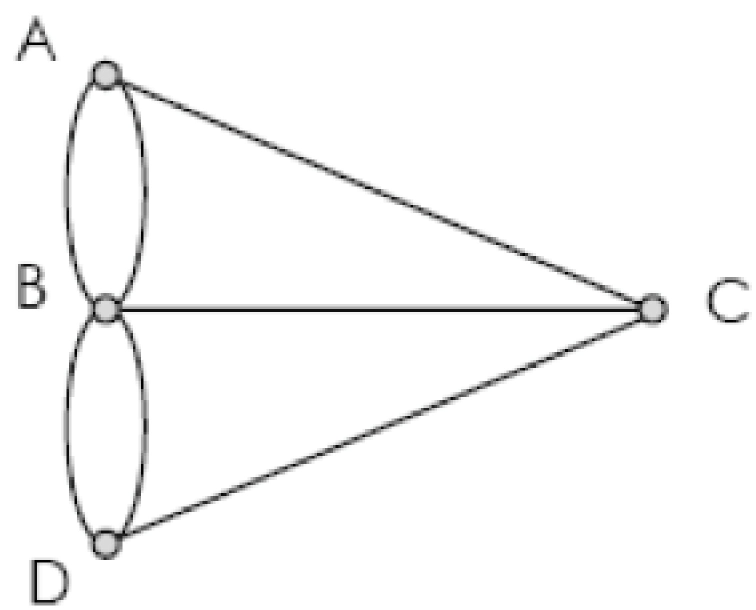
Euler Path & Circuit

The town of Konigsberg in Prussia (now Kaliningrad in Russia) was built at a point where two branches of the Pregel River came together. It consisted of an island and some land along the river banks. These were connected by seven bridges as shown in figure below:



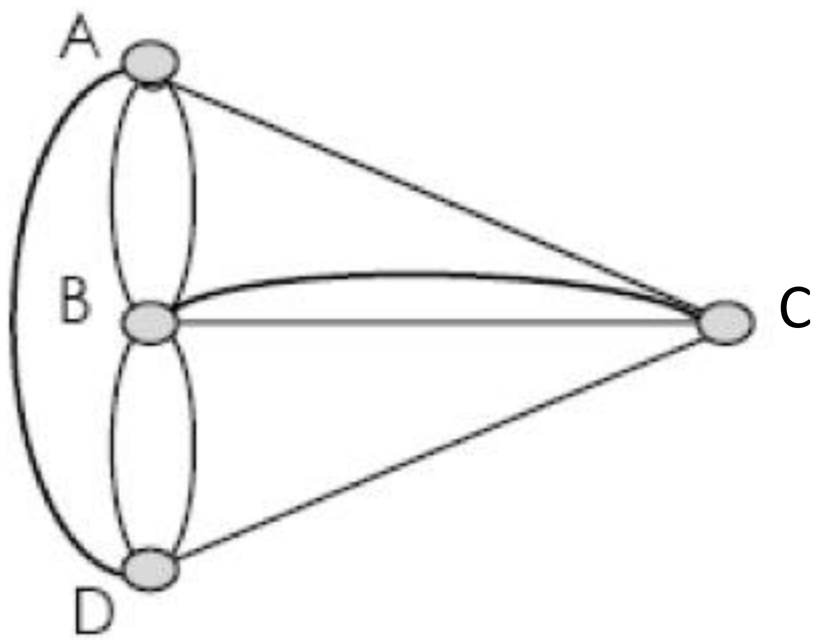
Problem: Starting at one land area, is it possible to walk across all of the bridges exactly once and return to the starting land area?

- Graph of the Königsberg Bridge Problem



no Euler cycle

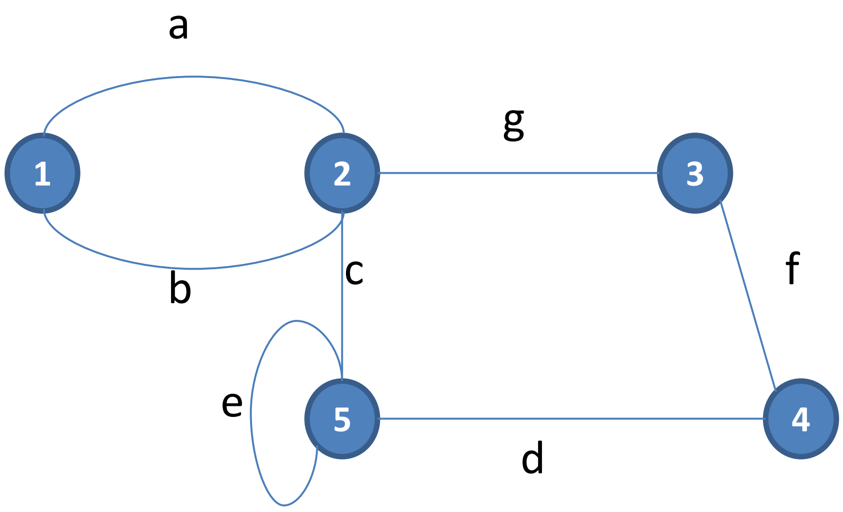
Solution: It is not possible to walk across all of the bridges exactly once and return to the starting land area. Therefore, two additional bridges have been constructed on the Pregel river.



Euler Circuit

Let *G* be a graph. An Euler circuit for *G* is a circuit that contains every vertex and every edges of *G*. That is, an Euler circuit for *G* is a sequence of adjacent vertices and edges in *G* that has at least one edges, starts and ends at the same vertex, uses every vertex of G at least once, and uses every edge of *G* exactly once.

Example – Euler Cycle

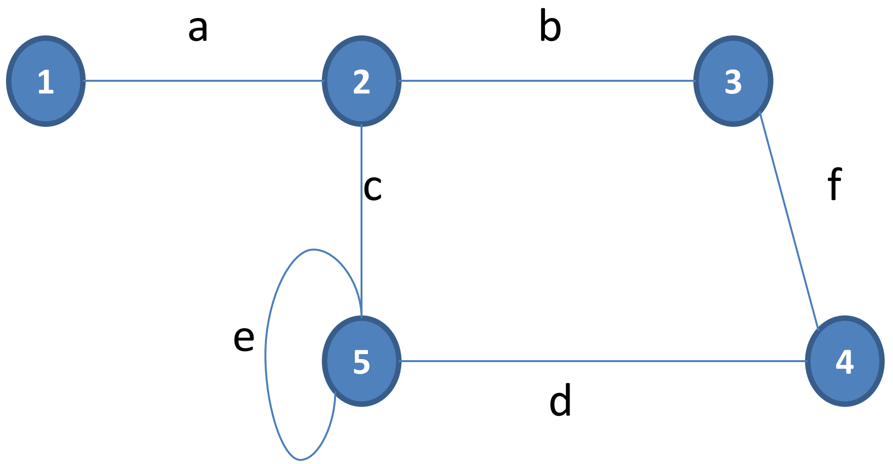


(**1**, a, **2**, c, **5**, e, **5**, d, **4**, f, **3**, g, **2**, b, **1**) is an Euler cycle.

Euler Trail

Let *G* be a graph, and let *v* and *w* be two distinct vertices of *G*. An Euler trail from *v* to *w* is a sequence of adjacent vertices and edges that starts at *v* and ends at *w*, passes through every **vertex** of *G* at least once, and traverses every **edge** of *G* exactly once.

Example – Euler Trail

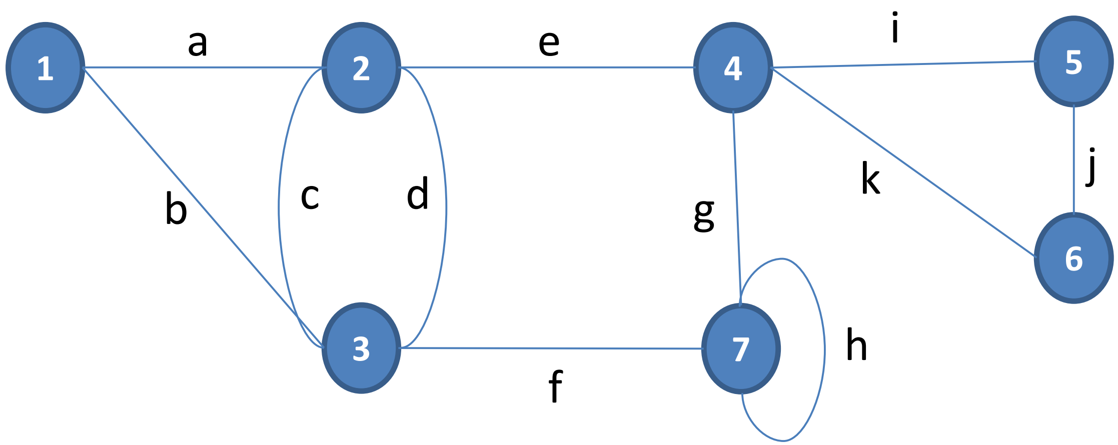


(**1**, a, **2**, c, **5**, e, **5**, d, **4**, f, **3**, b, **2**) is an Euler trail.

Theorem - Euler

- If G is a connected graph and every vertex has even degree, then G has an Euler circuit.
- A graph has an Euler trail from v to w $(v \neq w)$ if and only if it is connected and v and w are the only vertices having odd degree.

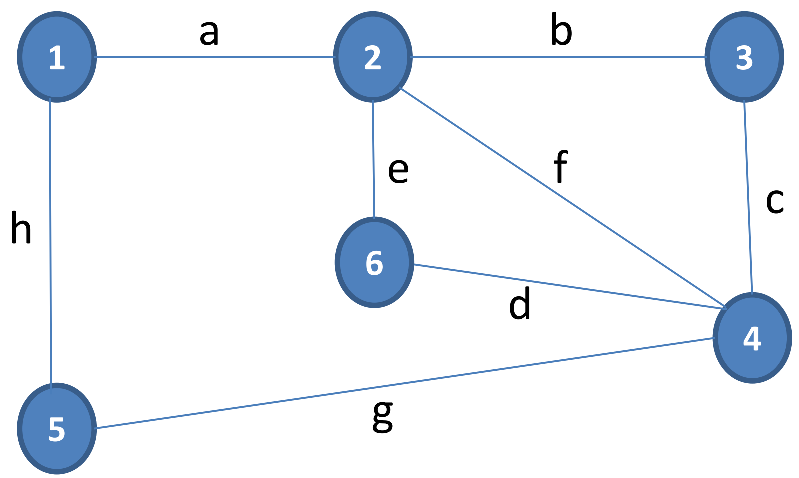
Example



This graph has an Euler cycle.

Vertex	1	2	3	4	5	6	7
Degree	2	4	4	4	2	2	4

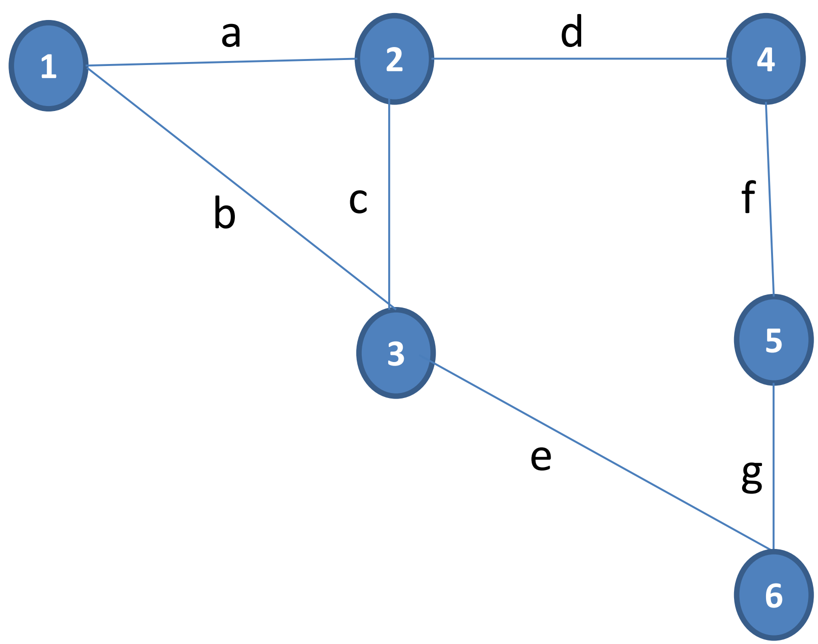
Example



This graph has an Euler cycle.

Vertex	1	2	3	4	5	6
Degree	2	4	2	4	2	2

Example

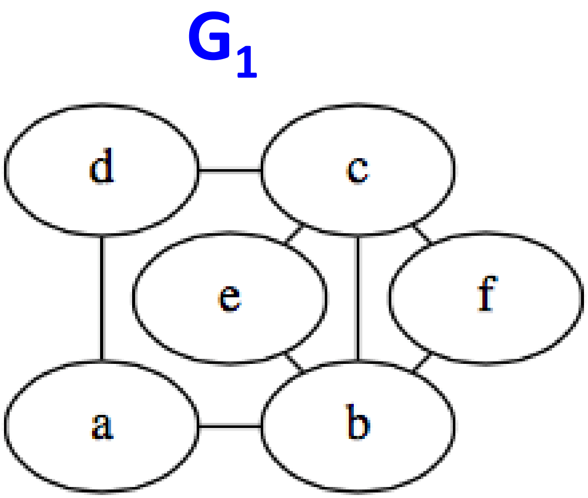


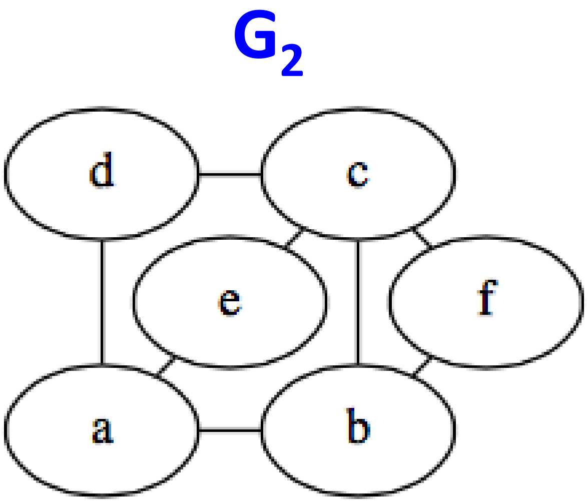
This graph has an Euler trail.

Vertex	1	2	3	4	5	6
Degree	2	3	3	2	2	2

Exercise # 6

Which of the following graphs has Euler circuit? Justify your answer.

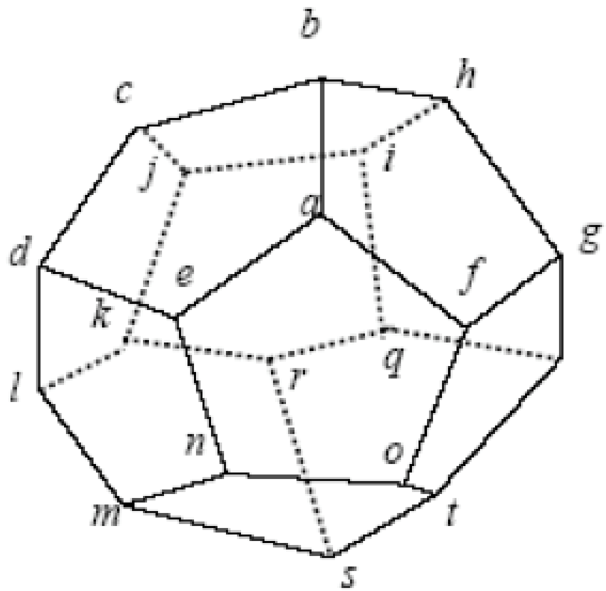


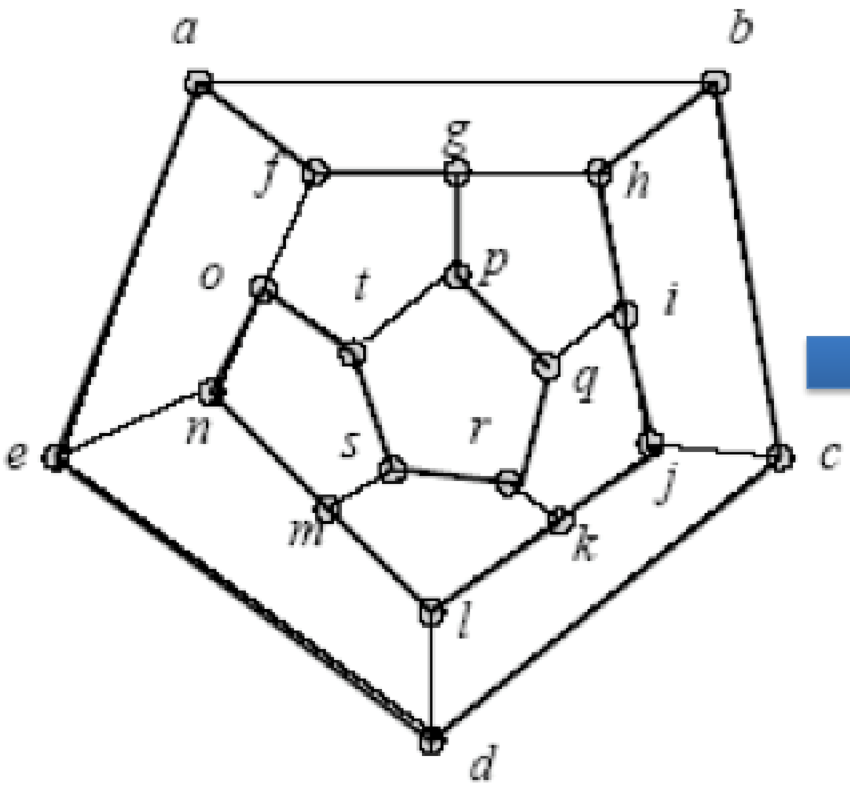


Hamilton Circuits

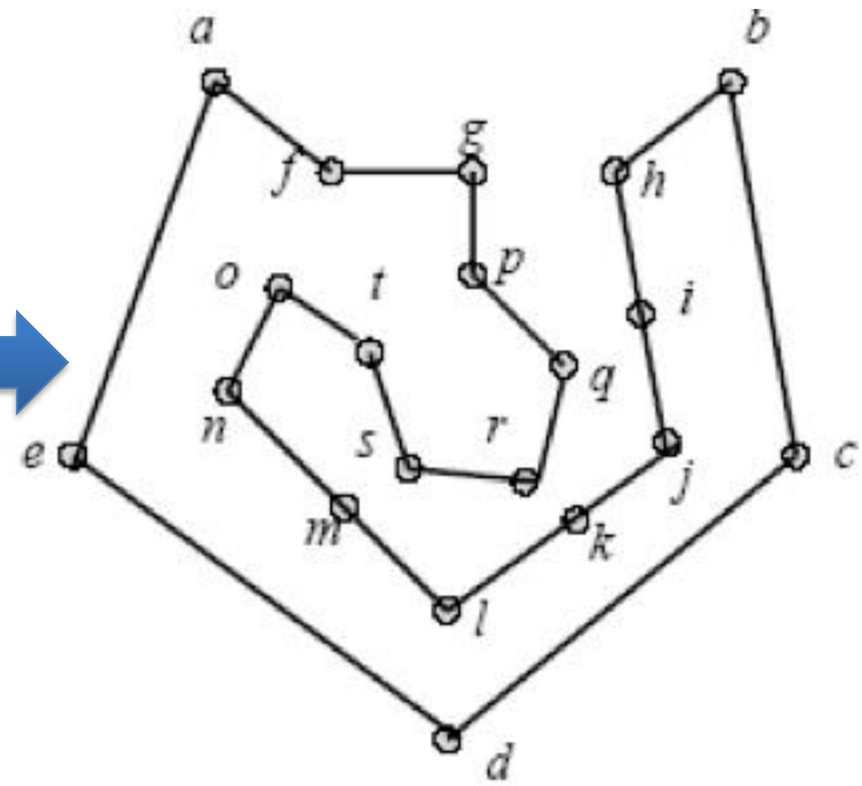
Given a graph *G*, a Hamiltonian circuit for *G* is a simple circuit that includes every vertex of *G* (but doesn't need to include all edges). That is, a Hamiltonian circuit for *G* is a sequence of adjacent vertices and distinct edges in which every vertex of *G* appears exactly once, except for the first and the last, which are the same.

- Sir William Rowan Hamilton marketed a puzzle in the mid-1800s in the form of dodecahedron.
- Each corner bore the name of a city.
- The problem was to start at any city, travel along the edges, visit each city exactly one time and return to the initial city.



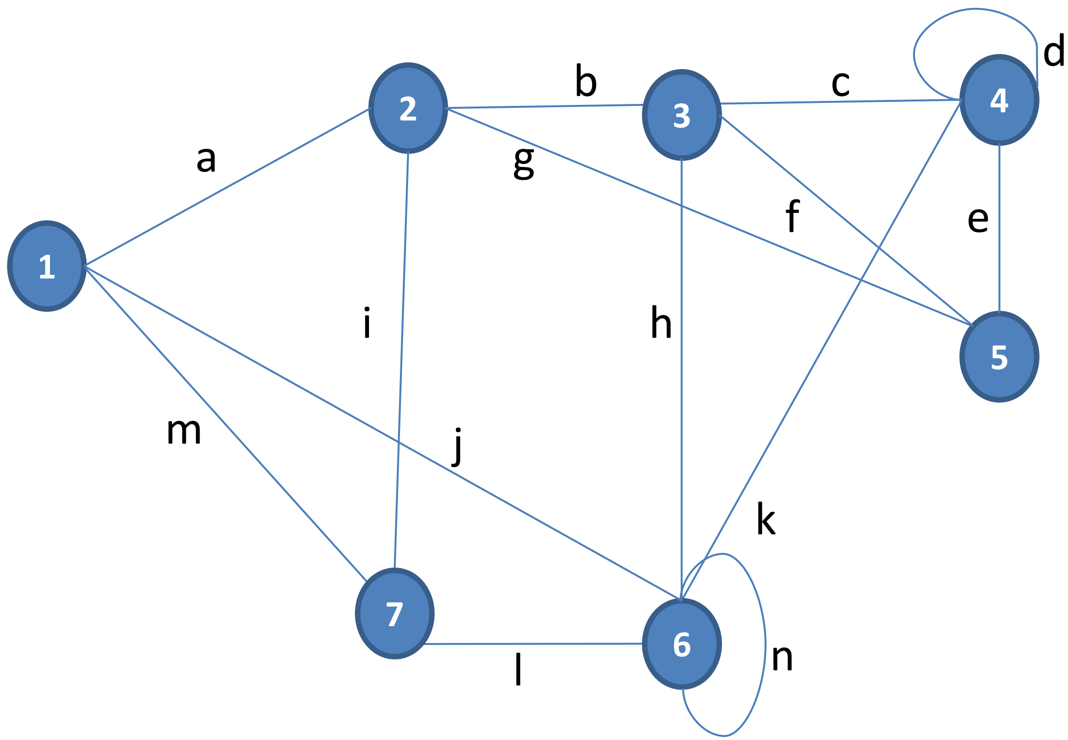


(a): The graph



(b): Hamilton circuit

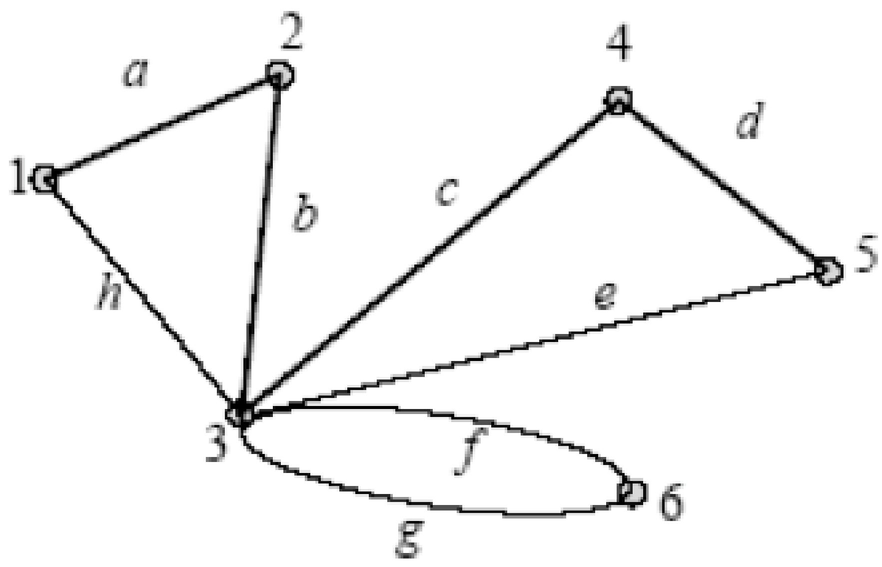
Example



This graph has Hamilton circuit.
The circuit is (1, a, 2, b, 3, f, 5, e, 4, k, 6, l, 7, m, 1)
Note: Visit each vertex just once.

Example

This graph does not contain Hamilton circuit.

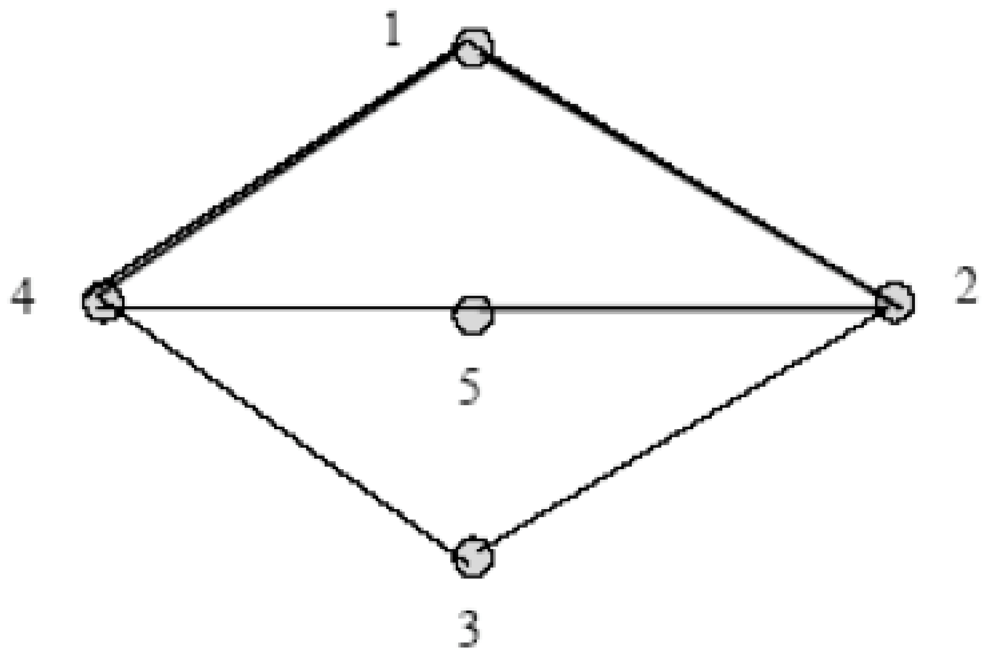


(1, a, 2, b, 3, g, 6, f, 3, e, 5, d, 4, c, 3, h, 1)

- Vertex (3) has to be visited more than once.

Example

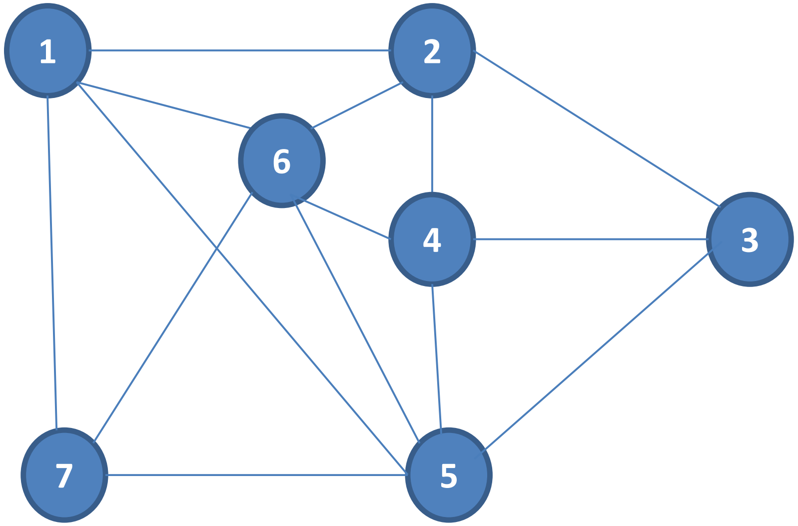
Is this graph has Hamiltonian cycle?



Solution: No, because vertex (4) has to be visited more than once. That is (1, 4, 3, 2, 5, 4, 1)

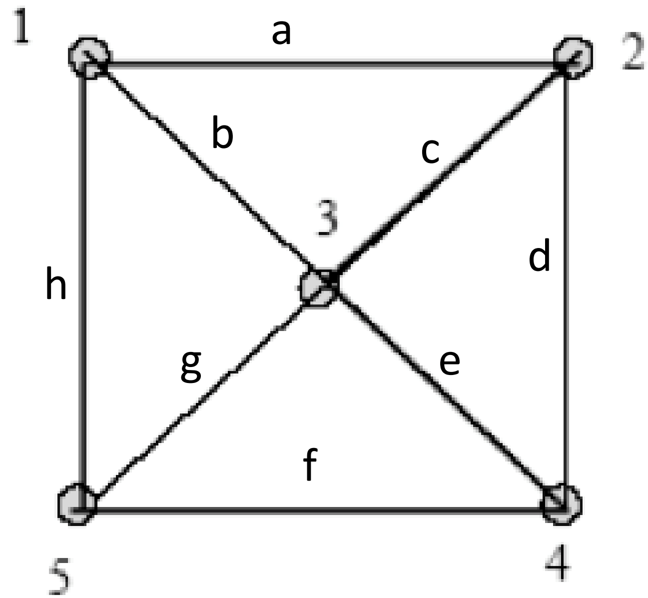
Exercise # 7

Is this graph has Hamiltonian cycle? If yes, exhibit one.



Exercise # 8

Prove that this graph has Hamiltonian cycle.



Exercise # 9

- Find a Hamiltonian cycle in this graph.