Graph Theory

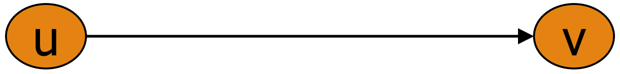
Definitions - Graph

A generalization of the simple concept of a set of dots, links, edges or arcs.

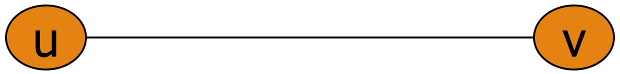
Representation: Graph G =(V, E) consists set of vertices denoted by V, or by V(G) and set of edges E, or E(G)

Definitions – Edge Type

Directed: Ordered pair of vertices. Represented as (u, v) directed from vertex u to v.

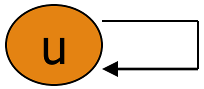


Undirected: Unordered pair of vertices. Represented as {u, v}. Disregards any sense of direction and treats both end vertices interchangeably.

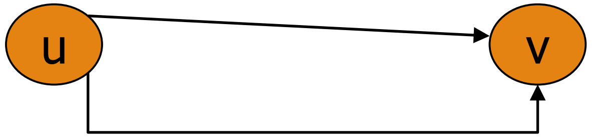


Definitions – Edge Type

Loop: A loop is an edge whose endpoints are equal i.e., an edge joining a vertex to it self is called a loop. Represented as {u, u} or {u}

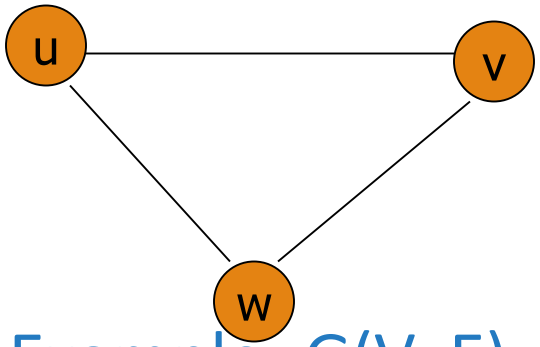


Multiple Edges: Two or more edges joining the same pair of vertices.



Definitions – Graph Type

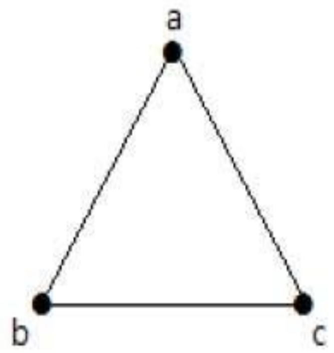
Simple (Undirected) Graph: A graph in which each edge connects two different vertices and where no two edges connect the same pair of vertices is called a **simple graph**.



Representation Example: G(V, E), V = {u, v, w}, E = {{u, v}, {v, w}, {u, w}}

The maximum number of edges possible in a simple graph with 'n' vertices is ${}^nC_2$ where ${}^nC_2 = n(n-1)/2$.

The number of simple graphs possible with 'n' vertices = $2^{{}^nc_2} = 2^{n(n-1)/2}$.

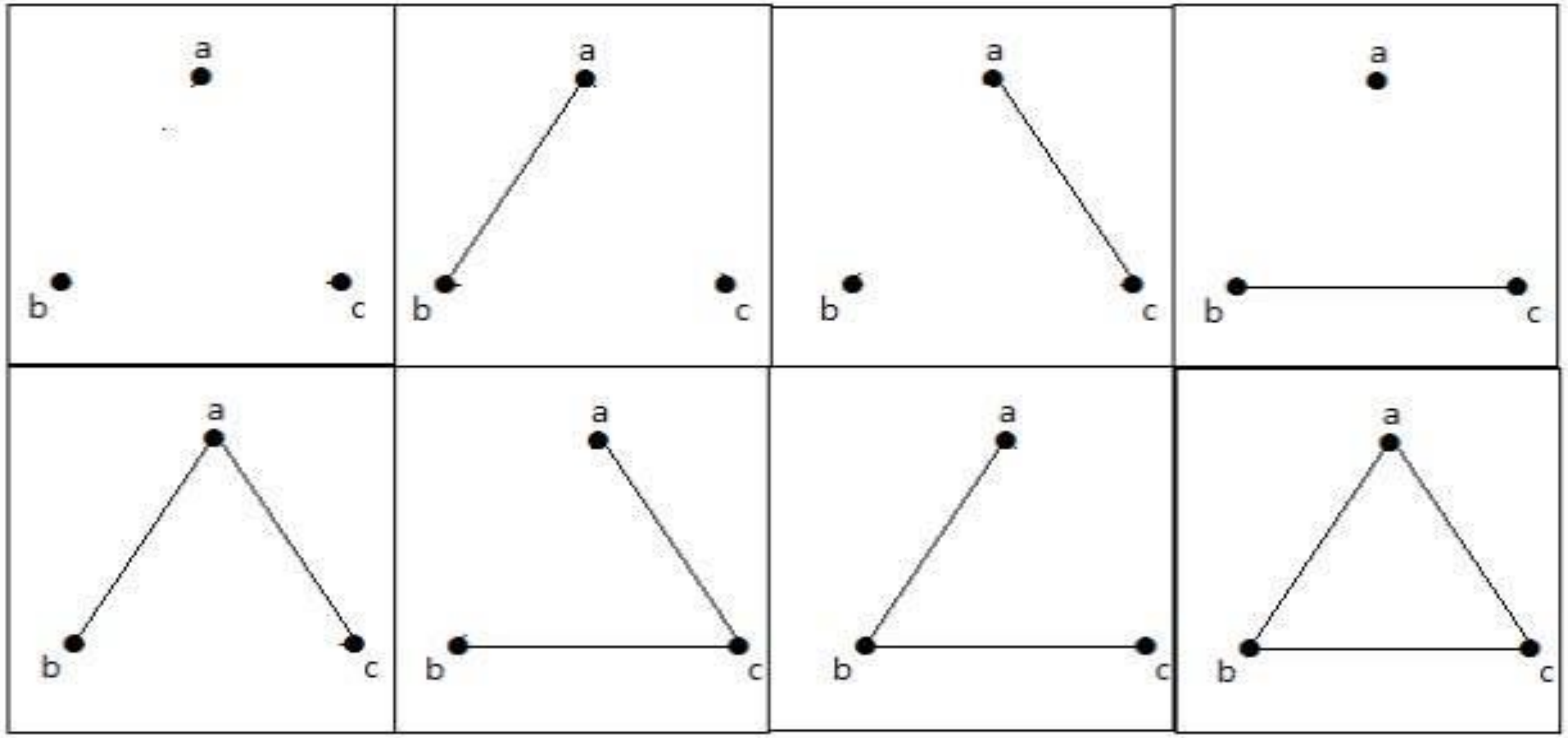


The maximum number of edges with n=3 vertices –
${}^nC_2 = n(n-1)/2 \quad = 3(3-1)/2 \quad = 6/2 \quad = 3$ edges

The maximum number of simple graphs with n=3 vertices −

$2^{{}^nC_2} = 2^{n(n-1)/2} = 2^{3(3-1)/2} = 2^3 = 8$

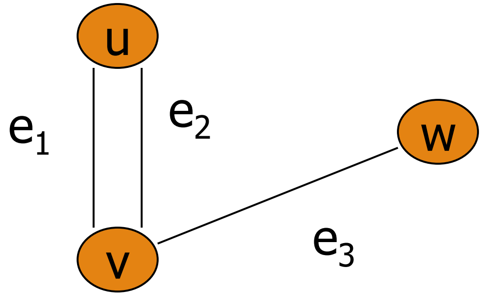
These 8 graphs are as shown below −



Definitions – Graph Type

Multigraph: Graphs that may have **multiple edges** connecting the same vertices are called **Multigraphs.**

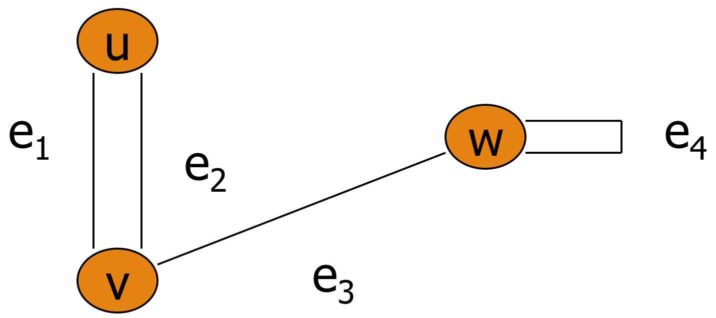
Representation Example: $V = \{u, v, w\}$, $E = \{e_1, e_2, e_3\}$



Definitions – Graph Type

Pseudograph: Graphs that may include loops, and possibly multiple edges connecting the same pair of vertices or a vertex to itself, are called **pseudographs**.

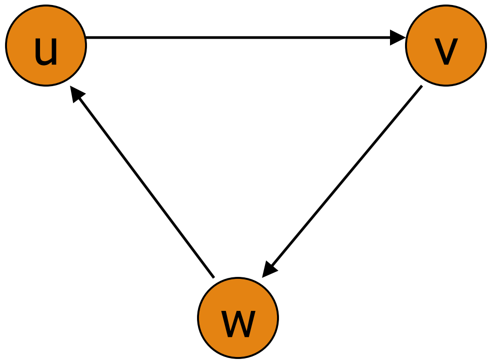
Representation Example: $V = \{u, v, w\}$, $E = \{e_1, e_2, e_3, e_4\}$



Definitions – Graph Type

Directed Graph: G(V, E), set of vertices V, and set of Edges E, that are ordered pair of elements of V (directed edges)

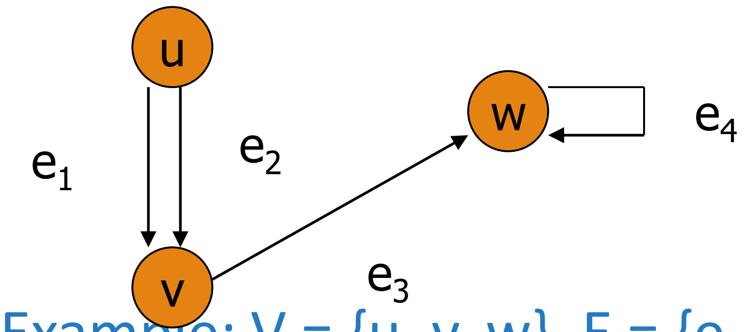
Representation Example: G(V, E), V = {u, v, w}, E = {(u, v), (v, w), (w, u)}



The directed edge associated with the ordered pair (u, v) is said to *start* at u and *end* at v.

Definitions – Graph Type

Directed Multigraph: Directed graphs that have **multiple directed edges** from a vertex to a second (possibly the same) vertex are called **directed multigraphs**. When there are *m* directed edges, each associated to an ordered pair of vertices *(u, v)*, we say that *(u, v)* is an edge of **multiplicity** *m*.



Representation Example: $V = \{u, v, w\}$, $E = \{e_1, e_2, e_3, e_4\}$

Definitions – Graph Type

Mixed graph: Sometimes we may need a graph where some edges are undirected, while others are directed.

A graph with both directed and undirected edges is called a **mixed graph**.

For example, a mixed graph might be used to model a computer network containing links that operate in both directions and other links that operate only in one direction.

Definitions – Graph Type

Type	Edges	Multiple Edges Allowed ?	Loops Allowed ?
Simple Graph	undirected	No	No
Multigraph	undirected	Yes	No
Pseudograph	undirected	Yes	Yes
Directed Graph	directed	No	Yes
Directed Multigraph	directed	Yes	Yes

Terminology — Undirected graphs

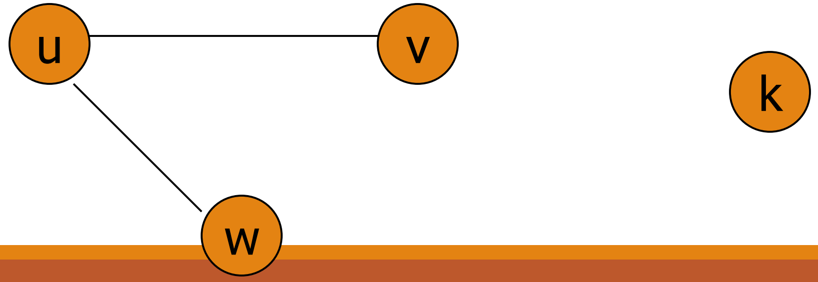
u and v are **adjacent** if {u, v} is an edge, e is called **incident** with u and v. u and v are called **endpoints** of {u, v}

Degree of Vertex (deg (v)): the number of edges incident on a vertex. A loop contributes twice to the degree.

Pendant Vertex: deg (v) =1

Isolated Vertex: deg (k) = 0

Representation Example: For V = {u, v, w} , E = { {u, w}, {u, v} }, deg (u) = 2, deg (v) = 1, deg (w) = 1, deg (k) = 0, w and v are pendant , k is isolated



Terminology – Directed graphs

For the edge (u, v), u is **adjacent to** v OR v is **adjacent from** u, u – **Initial vertex**, v – **Terminal vertex**

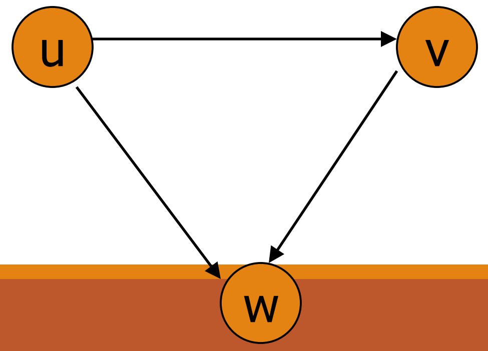
In-degree (deg^- (u)): number of edges for which u is terminal vertex

Out-degree (deg^+ (u)): number of edges for which u is initial vertex

Note: A loop contributes 1 to both in-degree and out-degree

Representation Example: For V = {u, v, w} , E = { (u, w), (v, w), (u, v) }, deg^- (u) = 0, deg^+ (u) = 2, deg^- (v) = 1,

deg^+ (v) = 1, and deg^- (w) = 2, deg^+ (u) = 0



Theorems: Undirected Graphs

<u>Theorem 1</u>

The Handshaking theorem:

$$2e = \sum_{v \in V} \deg(v)$$

Every edge connects 2 vertices

Theorems: Undirected Graphs

Theorem 2:

An undirected graph has even number of vertices with odd degree

Pr*oof* *V*1 is the set of even degree vertices and V2 refers to odd degree vertices

$$2e = \sum_{v \in V} \deg(v) = \sum_{u \in V_1} \deg(u) + \sum_{v \in V_2} \deg(v)$$

$\Rightarrow$ deg (v) is even for $v \in V_1$,

$\Rightarrow$ The first term in the right hand side of the last inequality is even.

$\Rightarrow$ The sum of the last two terms on the right hand side of the last inequality is even since sum is 2e.

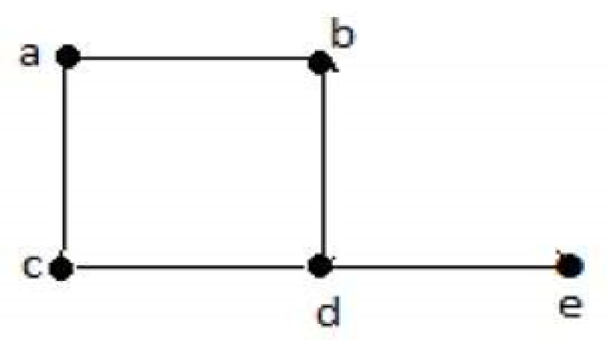
Hence second term is also even

$$\Rightarrow \text{second term } \sum_{v \in V_2} \deg(u) = even$$

Degree Sequence of a Graph

If the degrees of all vertices in a graph are arranged in descending or ascending order, then the sequence obtained is known as the degree sequence of the graph.

Example 1



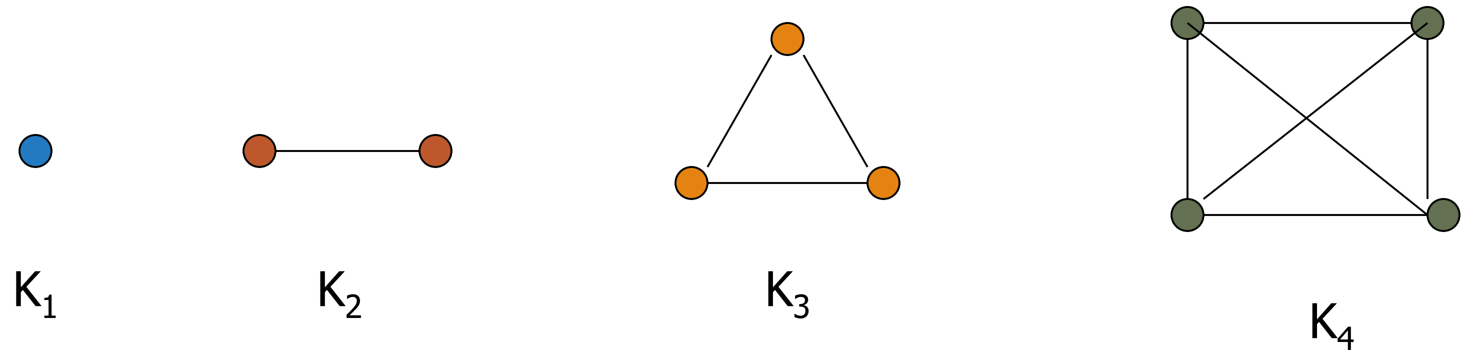
Vertex	a	b	c	d	e
Connecting to	b,c	a,d	a,d	c,b,e	d
Degree	2	2	2	3	1

In the above graph, for the vertices {d, a, b, c, e}, the degree sequence is {3, 2, 2, 2, 1}.

Simple graphs – special cases

Complete graph: K_n, is the simple graph that contains exactly one edge between each pair of distinct vertices.

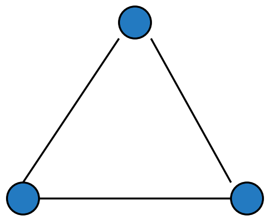
Representation Example: K_1, K_2, K_3, K_4



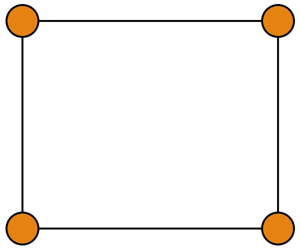
Simple graphs – special cases

Cycle: C_n, $n \geq 3$ consists of n vertices v_1, v_2, v_3 … v_n and edges $\{v_1, v_2\}$, $\{v_2, v_3\}$, $\{v_3, v_4\}$ … $\{v_{n-1}, v_n\}$, $\{v_n, v_1\}$

Representation Example: C_3, C_4



C_3

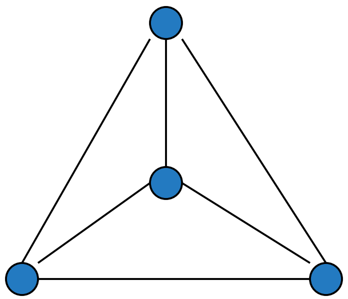


C_4

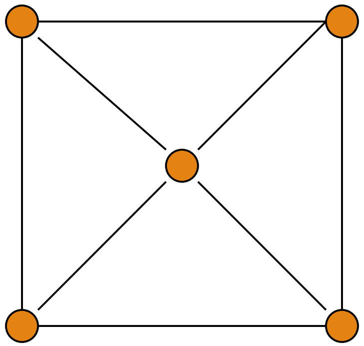
Simple graphs – special cases

Wheels: W_n, obtained by adding additional vertex to Cn and connecting all vertices to this new vertex by new edges.

Representation Example: W_3, W_4



W_3

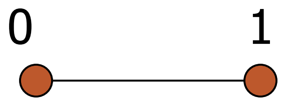


W_4

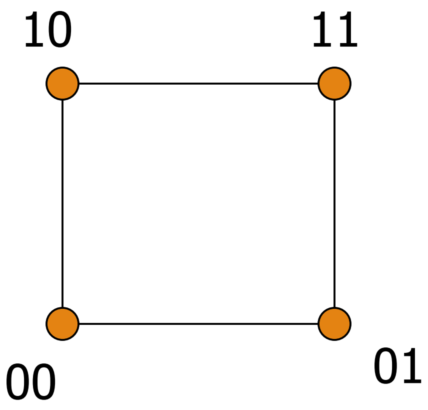
Simple graphs – special cases

N-cubes: Q_n, vertices represented by 2^n bit strings of length n. Two vertices are adjacent if and only if the bit strings that they represent differ by exactly one bit positions

Representation Example: Q_1, Q_2



Q_1



Q_2

Representation

Incidence (Matrix): Most useful when information about edges is more desirable than information about vertices.

Adjacency (Matrix/List): Most useful when information about the vertices is more desirable than information about the edges. These two representations are also most popular since information about the vertices is often more desirable than edges in most applications

Representation- Incidence Matrix

- G = (V, E) be an undirected graph. Suppose that $v_1, v_2, v_3, \ldots, v_n$ 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 nx m matrix M = $[m_{ij}]$, where

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

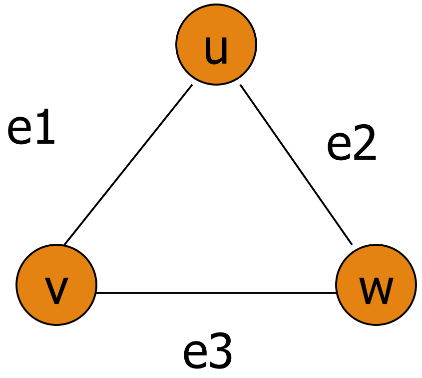
Can also be used to represent :

Multiple edges: by using columns with identical entries, since these edges are incident with the same pair of vertices

Loops: by using a column with exactly one entry equal to 1, corresponding to the vertex that is incident with the loop

Representation- Incidence Matrix

Representation Example: G = (V, E)



	e_1	e_2	e_3
v	1	0	1
u	1	1	0
w	0	1	1

Representation- Adjacency Matrix

- There is an N x N matrix, where |V| = N , the Adjacenct Matrix (NxN) A = $[a_{ij}]$

For undirected graph

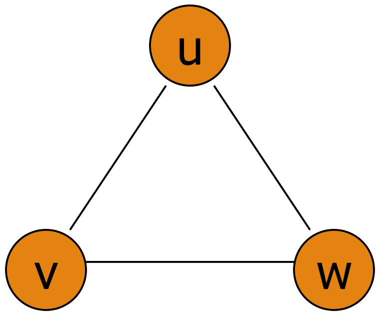
$$a_{ij} = \begin{cases} 1 \text{ if } v_i \text{ and } v_j \text{ are adjacent} \\ 0 \text{ otherwise} \end{cases}$$

- **For directed graph**

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

Representation- Adjacency Matrix

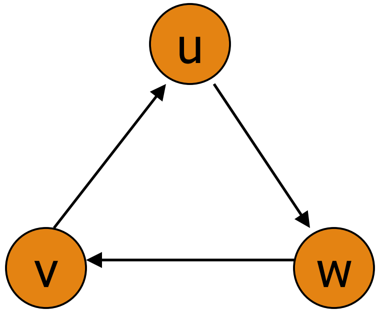
Example: Undirected Graph G (V, E)



	v	u	w
v	0	1	1
u	1	0	1
w	1	1	0

Representation- Adjacency Matrix

Example: directed Graph G (V, E)

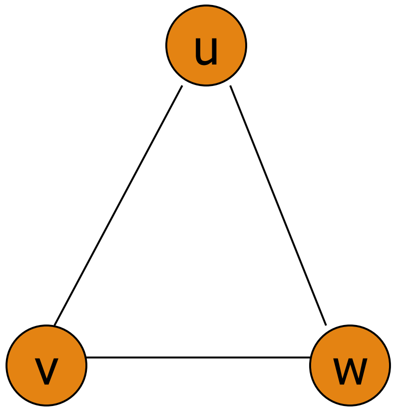


	v	u	w
v	0	1	0
u	0	0	1
w	1	0	0

Representation- Adjacency List

Each node (vertex) has a list of which nodes (vertex) it is adjacent

Example: undirectd graph G (V, E)

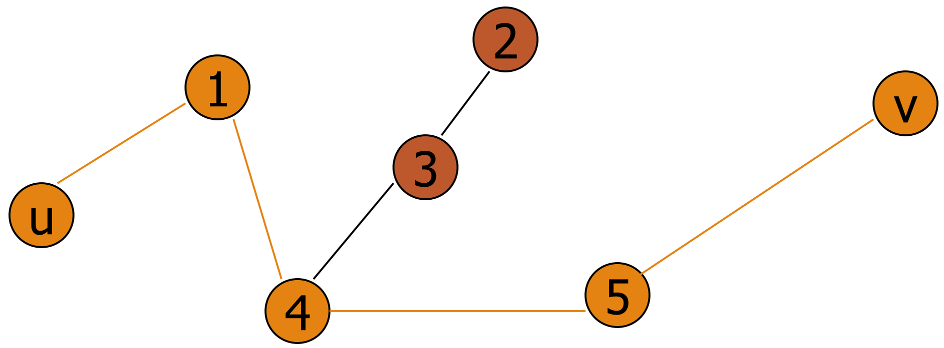


node	Adjacency List
u	v , w
v	w, u
w	u , v

Connectivity – Path

A **Path** is a sequence of edges that begins at a vertex of a graph and travels along edges of the graph, always connecting pairs of adjacent vertices.

Representation example: G = (V, E), Path P represented, from u to v is {{u, 1}, {1, 4}, {4, 5}, {5, v}}



Connectivity – Path

Path for Directed Graphs

A sequence of connected ordered pairs.

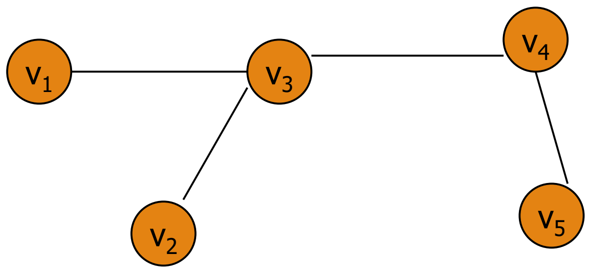
Circuit/Cycle: $u = v$, length of path > 0

Simple Path: does not contain an edge more than once

Connectivity – Connectedness

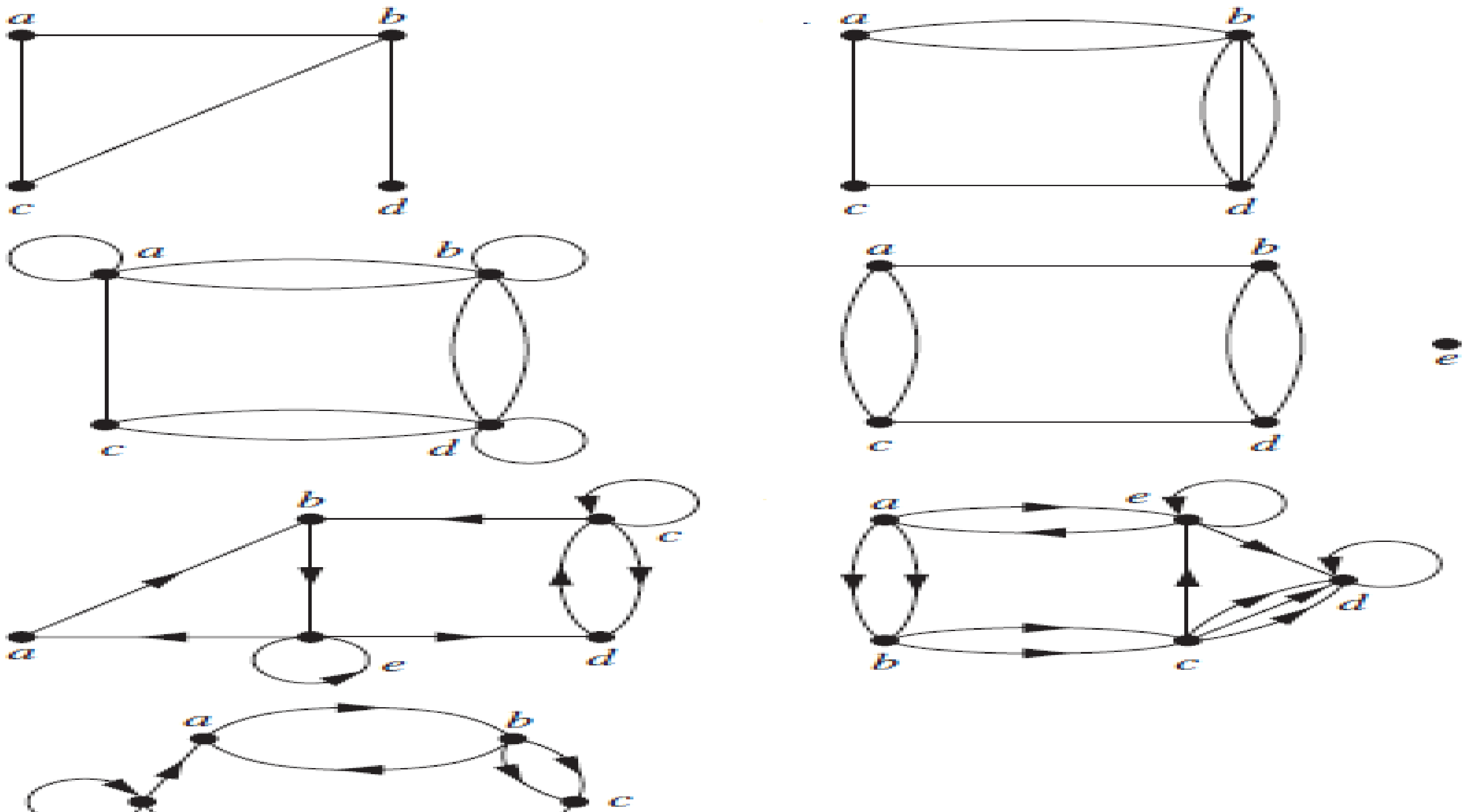
An graph is connected if there exists is a simple path between every pair of vertices

Representation Example: G (V, E) is connected since for V = $\{v_1, v_2, v_3, v_4, v_5\}$, there exists a path between $\{v_i, v_j\}$, $1 \leq i, j \leq 5$



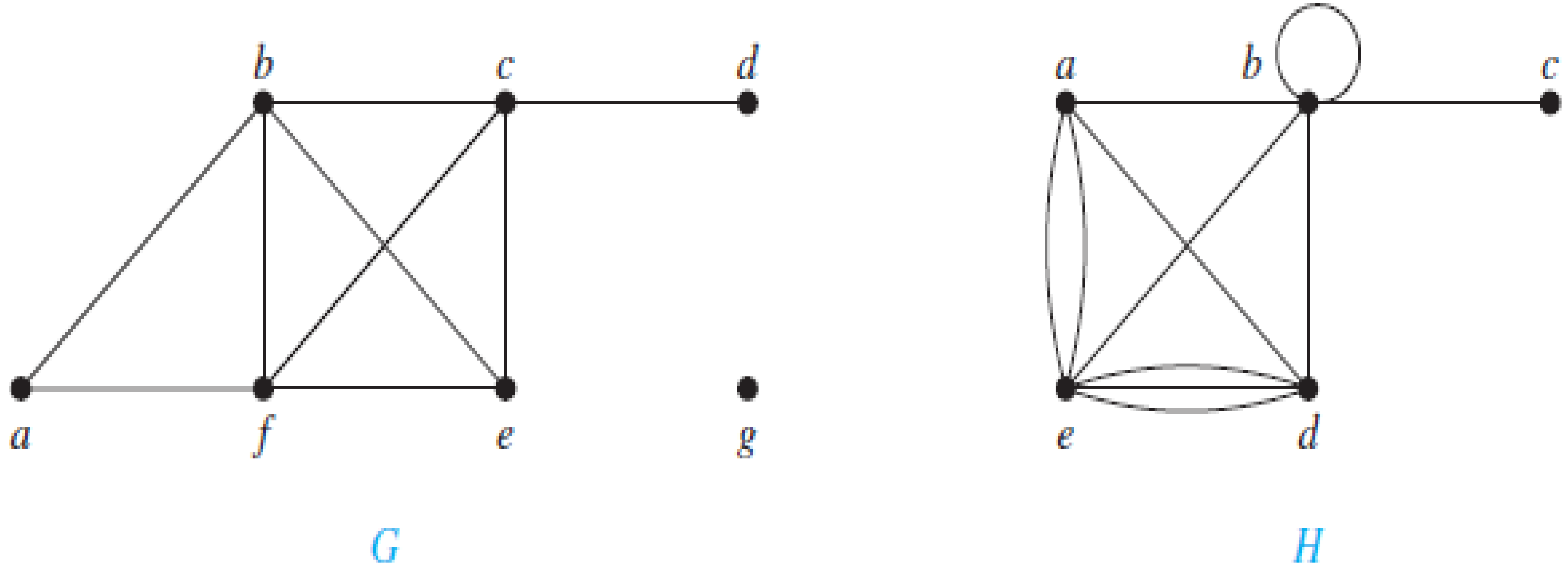
Class Exercise 1

Determine whether the graph shown has directed or undirected edges, whether it has multiple edges, and whether it has one or more loops. Use your answers to determine the type of graph is.



Class Exercise 2

What are the degrees and what are the neighborhoods of the vertices in the graphs *G* and *H* displayed in Figure?

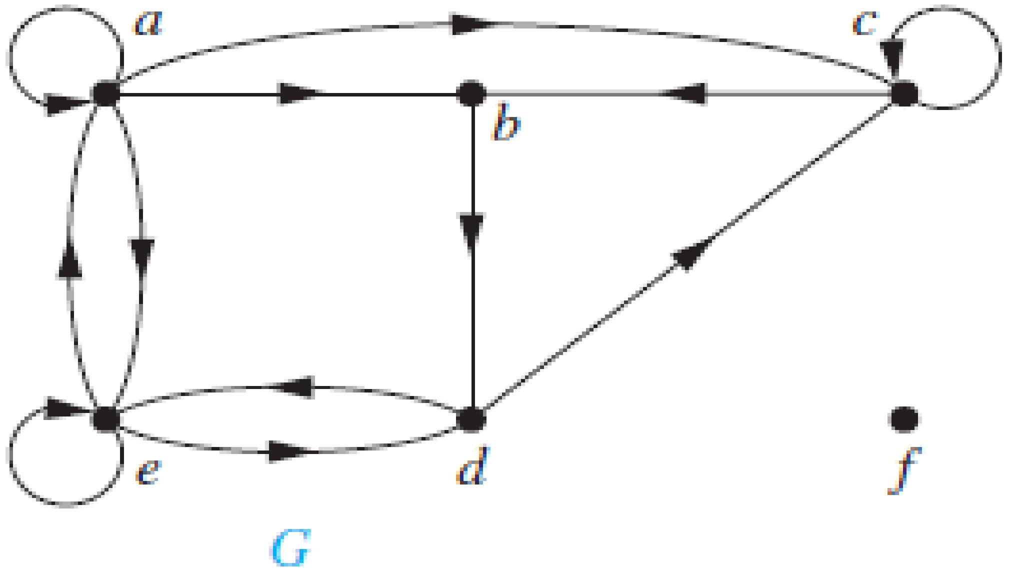


Class Exercise 3

How many edges are there in a graph with 10 vertices each of degree six?

Class Exercise 4

Find the in-degree and out-degree of each vertex in the graph *G* with directed edges shown in Figure.



Class Exercise 5

How many edges does a graph have if its degree sequence is 4, 3, 3, 2, 2? Draw such a graph.

Class Exercise 6

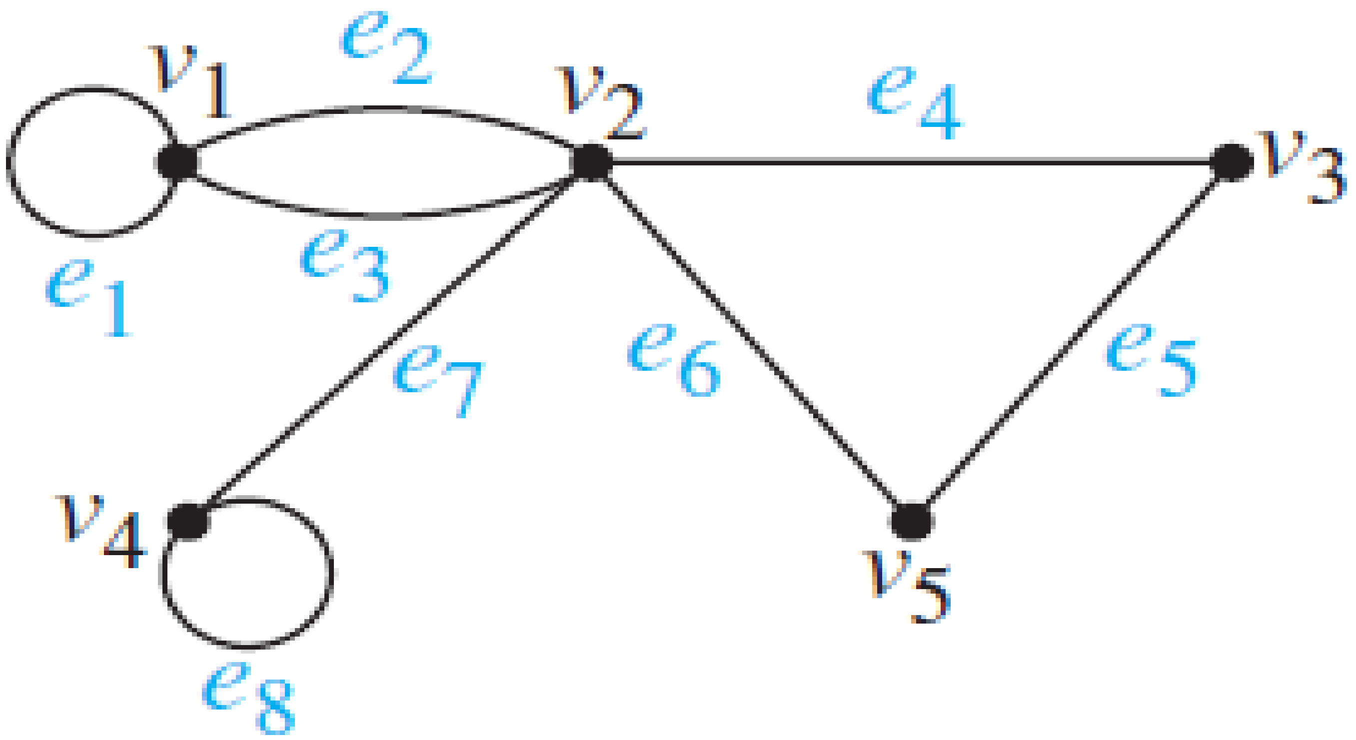
Draw a graph with the adjacency matrix

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

with respect to the ordering of vertices a, b, c, d.

Class Exercise 7

Represent the pseudograph shown in Figure using an incidence matrix.

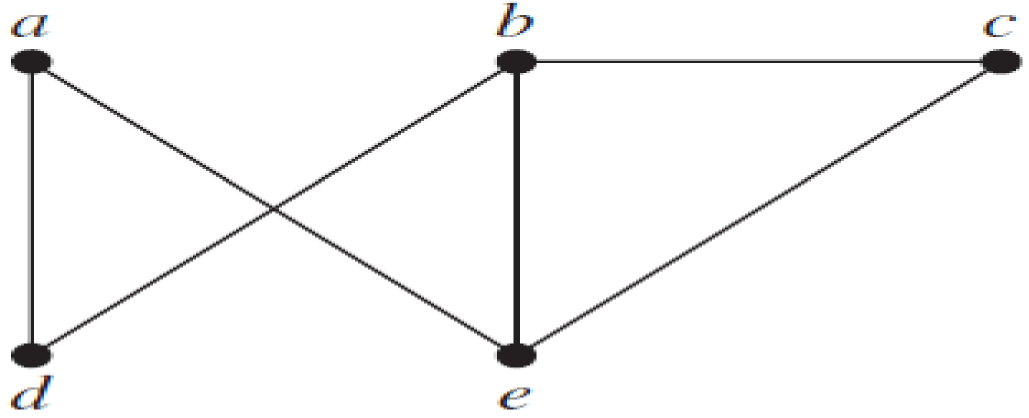


Class Exercise 8

Does each of these lists of vertices form a path in the following graph? Which paths are simple? Which are circuits? What are the lengths of those that are paths?

a) *a, e, b, c, b* **b)** *a, e, a, d, b, c, a*

c) *e, b, a, d, b, e* **d)** *c, b, d, a, e, c*

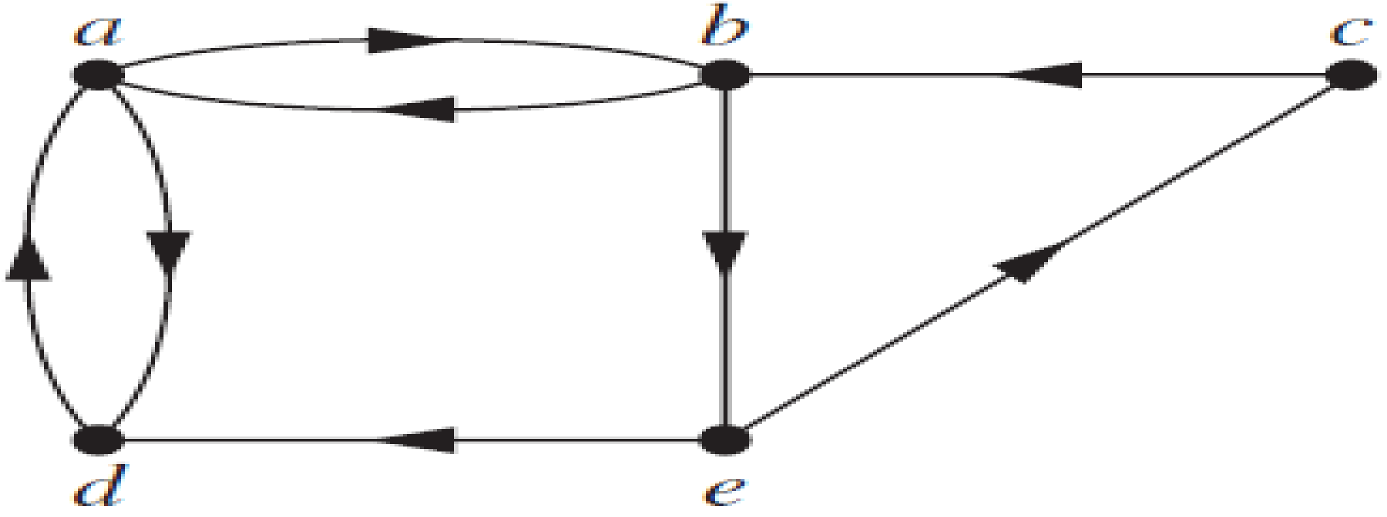


Class Exercise 9

Does each of these lists of vertices form a path in the following graph? Which paths are simple? Which are circuits? What are the lengths of those that are paths?

a) *a, b, e, c, b* **b)** *a, d, a, d, a*

c) *a, d, b, e, a* **d)** *a, b, e, c, b, d, a*



Class Exercise 10

Find the degree sequence of each of the following graphs.

a) *K4*

b) *C4*

c) *W4*

d) *Q3*

Class Exercise 11

Let G be a simple graph with n vertices. Show that the number of edges in the graph is

$n(n-1)/2$.

Class Exercise 12

A Graph has 12 edges, two vertices of degree 3, two vertices of degree 4and other vertices of degree 5. Find the number of vertices in the graph.

Class Exercise 13

Prove that in a full binary tree with n vertices, the number of pendant vertices is $(n+1)/2$.