Graph Theory and Tree

By

Dr. Yogesh M. Rajput

Assistant Professor

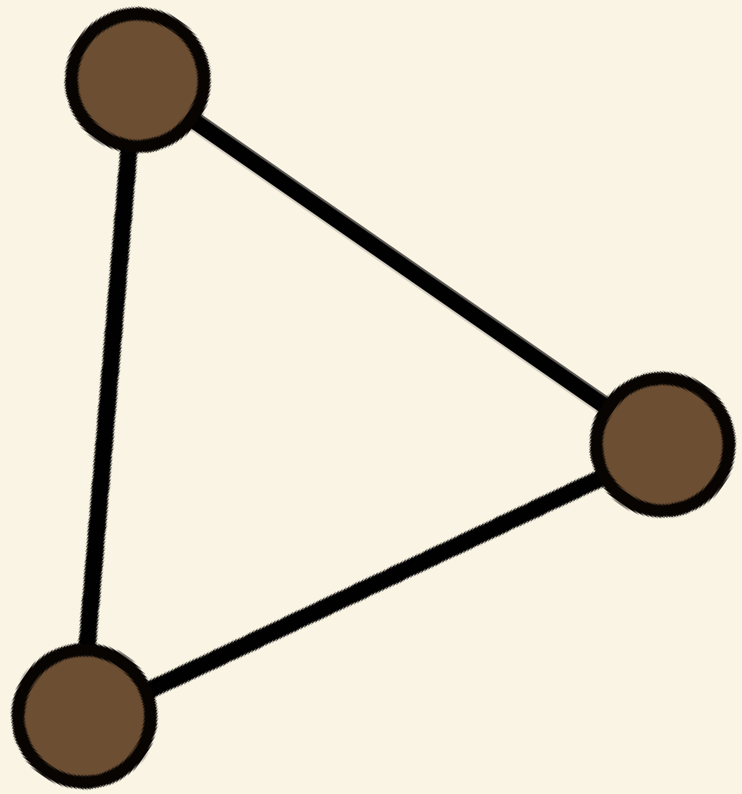
Dr. G. Y. Pathrikar College of Computer Science and Information Technology,

MGM University, Aurangabad.

Introduction to Graph Theory

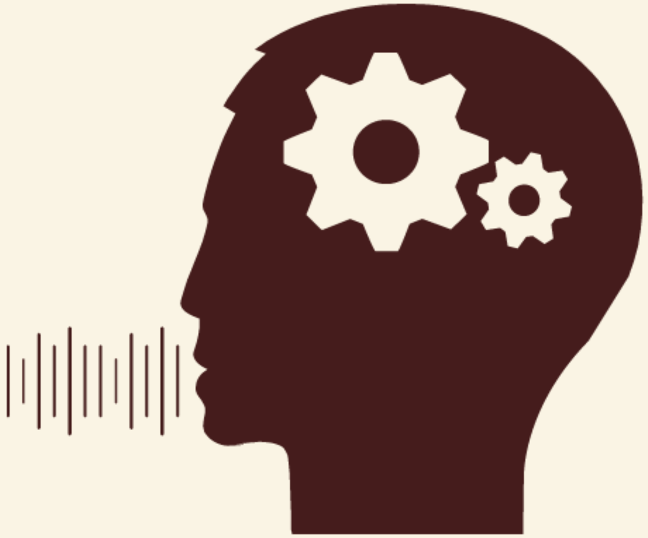
Introduction to Graph

- In the domain of mathematics and computer science, graph theory is the study of graphs that concerns with the relationship among **edges** and **vertices**.



Introduction to Graph

- It is a popular subject having its **applications** in computer science, information technology, biosciences, mathematics, and linguistics to name a few.



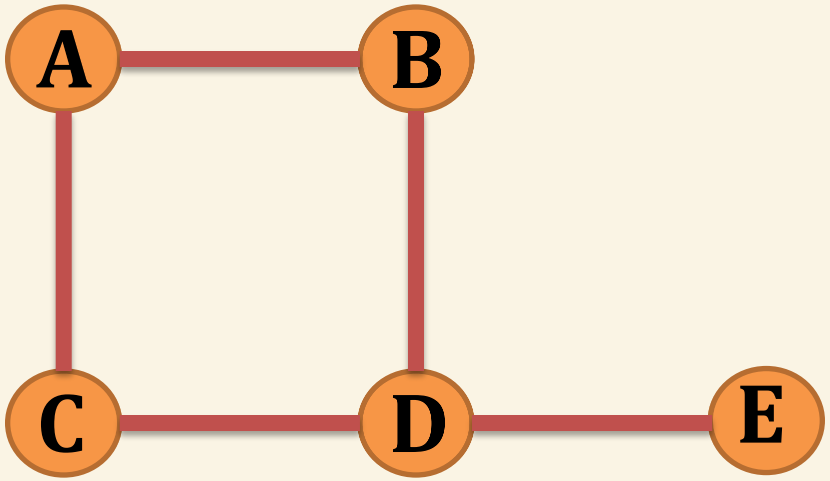
What is Graph?

Introduction to Graph

- A graph is a **pictorial** representation of a set of **objects** where some pairs of objects are connected by **links**.
- The interconnected objects are represented by points termed as **vertices**, and the links that connect the vertices are called **edges**.

Introduction to Graph

- Formally, a graph is a pair of sets **(V, E)**, where **V** is the set of **vertices** and **E** is the set of **edges**, connecting the pairs of vertices.

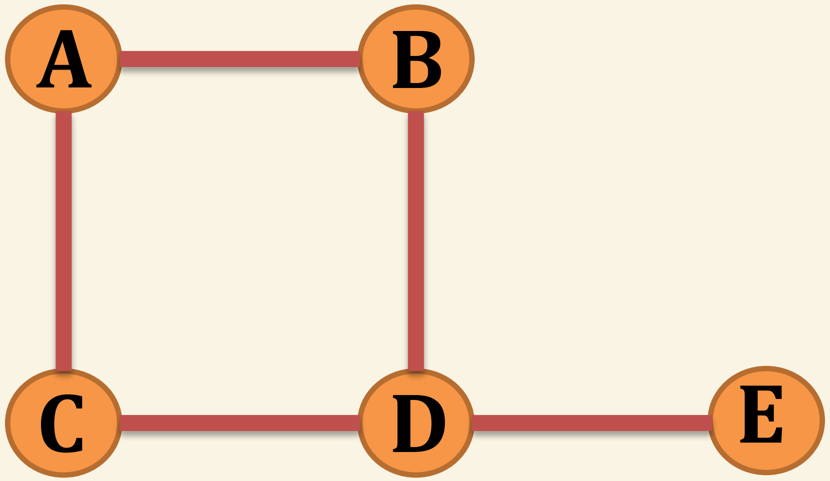


Introduction to Graph

- In the graph,

  V = {A, B, C, D, E}

  E = {AB, AC, BD, CD, DE}



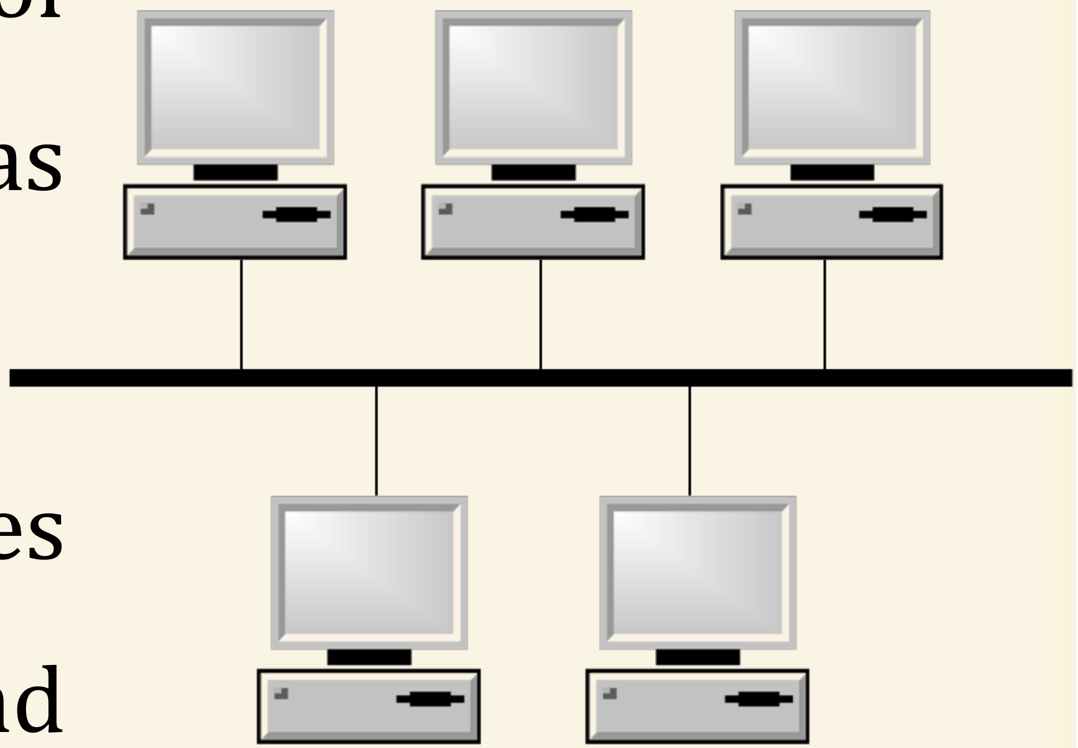
Applications of Graph Theory

1. Electrical Engineering

- The concepts of graph theory is used extensively in designing circuit connections.

1. Electrical Engineering

- The types or organization of connections are named as topologies.
- Some examples for topologies are star, bridge, series, and parallel topologies.



2. Computer Science

- Graph theory is used for the study of algorithms. For example,
  - Kruskal's Algorithm
  - Prim's Algorithm
  - Dijkstra's Algorithm

3. Computer Network

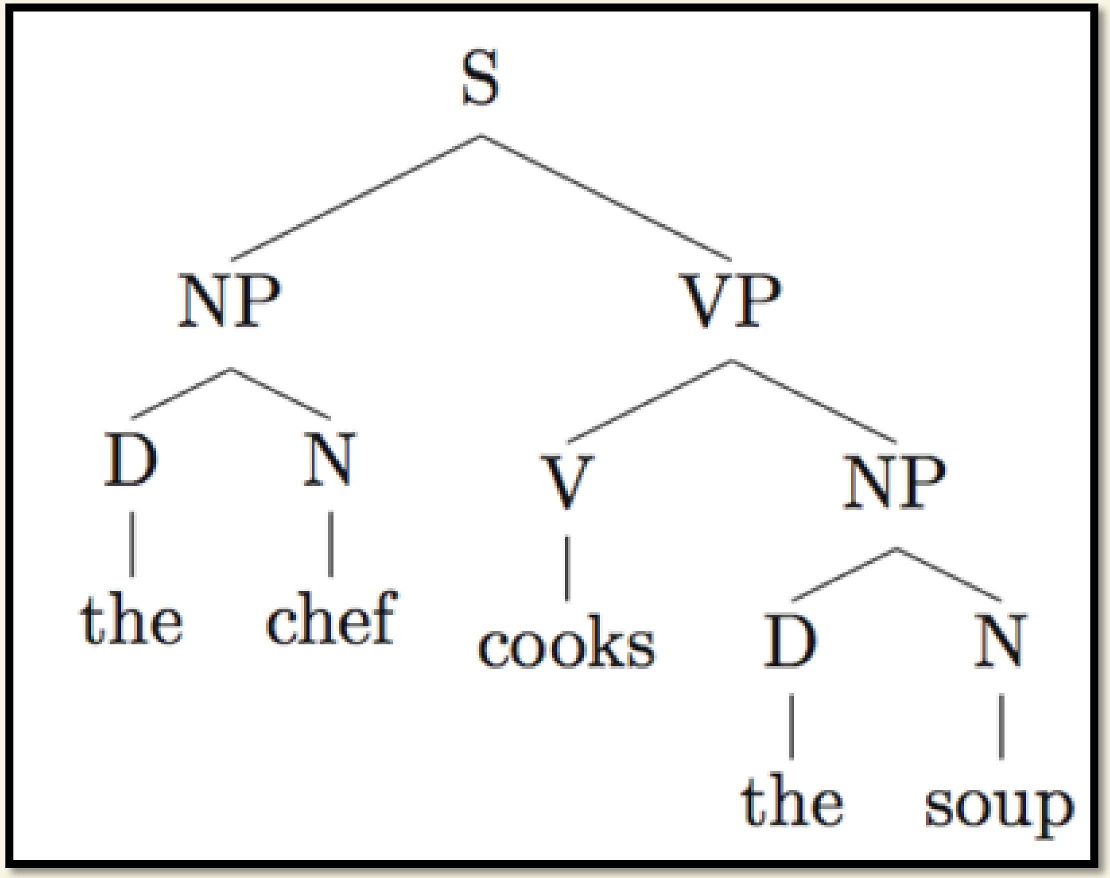
- The relationships among interconnected computers in the network follows the principles of graph theory.

4. Science

- The molecular structure and chemical structure of a substance, the DNA structure of an organism, etc., are represented by graphs.

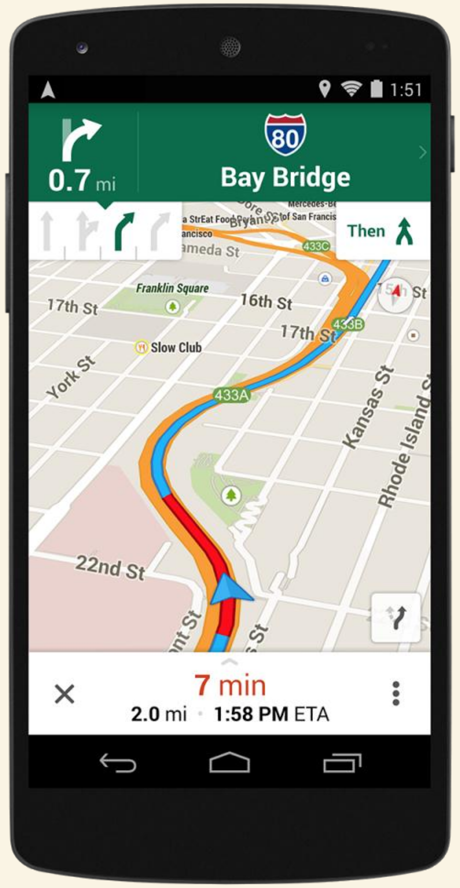
5. Linguistics

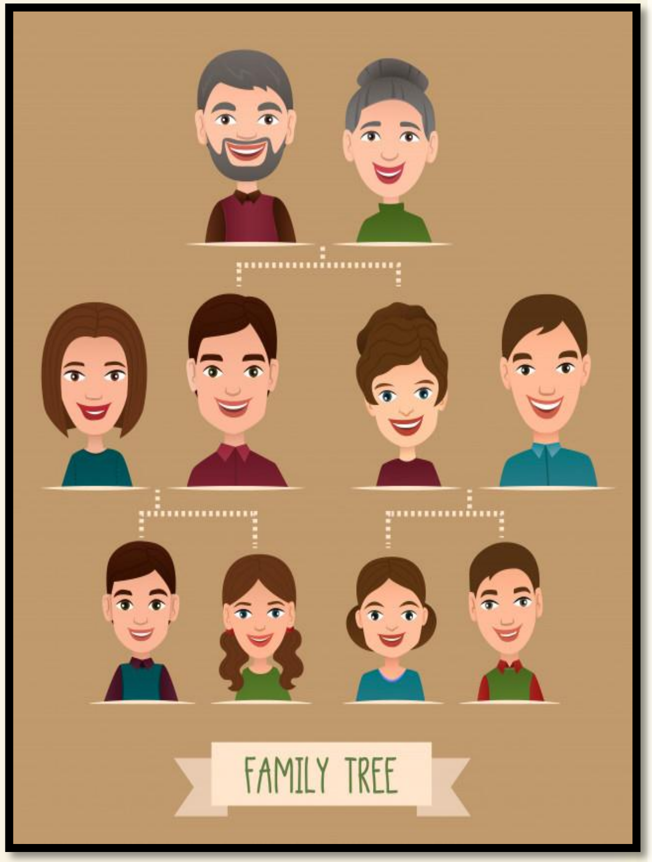
- The parsing tree of a language and grammar of a language uses graphs



6. General

- Routes between the cities can be represented using graphs.
- Depicting hierarchical ordered information such as family tree can be used as a special type of graph called tree.

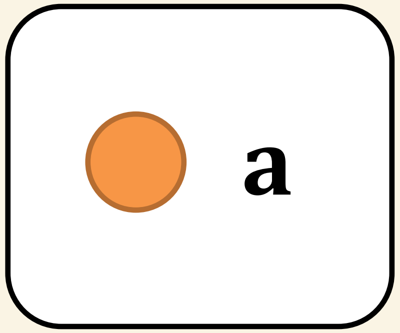




Fundamentals of Graph Theory

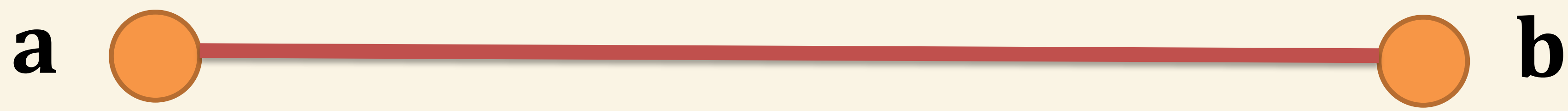
1. Point

- A **point** is a particular position in a one-dimensional, two-dimensional, or three-dimensional space.
- For better understanding, a point can be denoted by an **alphabet**.
- It can be represented with a **dot**.



2. Line

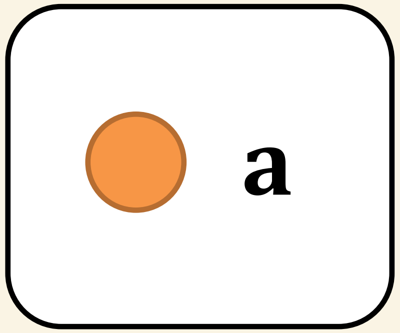
- A **Line** is a connection between **two points**.
- It can be represented with **a solid line**.



- Here, ‘**a**’ and ‘**b**’ are the points.
- The **link** between these **two** points is called a **line**.

3. Vertex

- A **vertex** is a point where **multiple lines** meet.
- It is also called a **node**.
- Similar to points, a **vertex** is also denoted by an **alphabet**.
- Here, the vertex is named with an alphabet '**a**'.



4. Edge

- An **edge** is the mathematical term for a **line** that connects **two vertices**.
- Many edges can be formed from a single vertex.
- Without a vertex, an **edge** cannot be formed.
- There must be a **starting** vertex and an **ending** vertex for an edge.

4. Edge

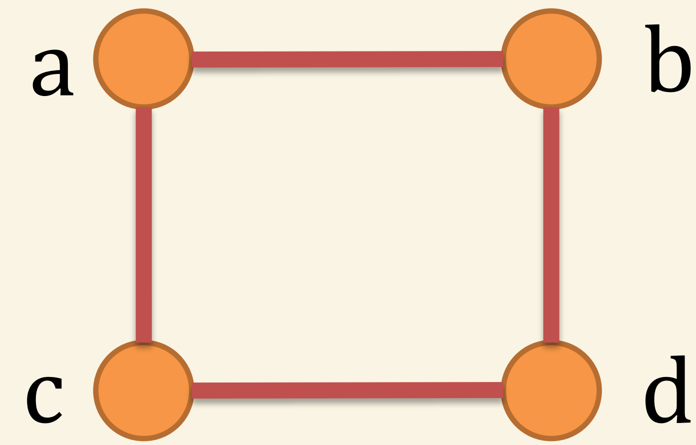
- Here, ‘**a**’ and ‘b’ are the points.
- The link between these two points is called a line.

5. Graph

- A graph 'G' is defined as

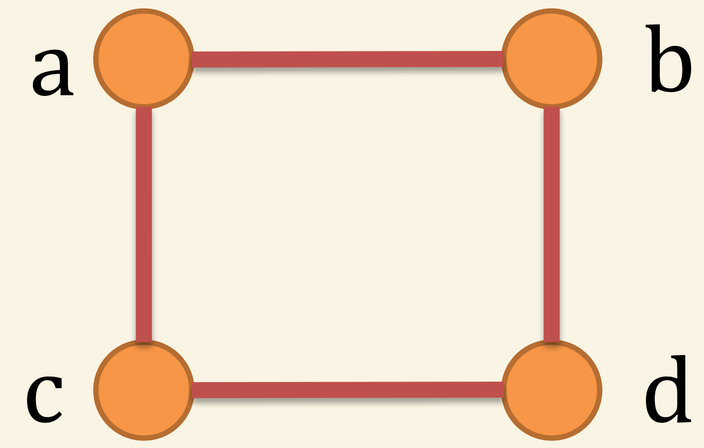
G = (V, E)

- Where V is a set of all vertices and
- E is a set of all edges in the graph.



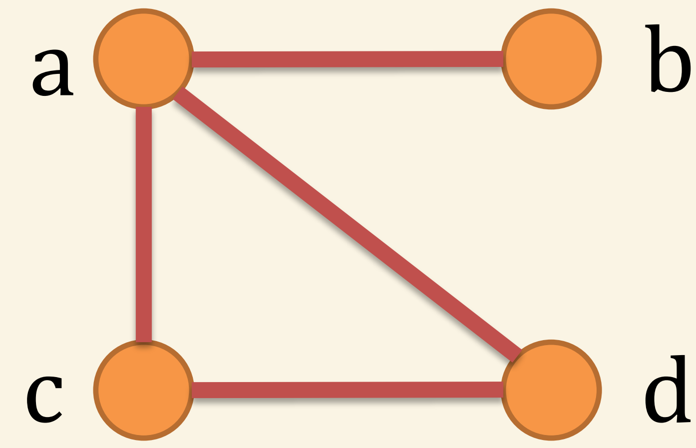
5. Graph

- In the example, **ab**, **ac**, **cd**, and **bd** are the **edges** of the graph.
- Similarly, **a**, **b**, **c**, and **d** are the **vertices** of the graph.



5. Graph

- **Example 2**
- In this graph, there are four vertices **a**, **b**, **c**, and **d**, and four edges **ab**, **ac**, **ad**, and **cd**.



V = {a, b, c, d}

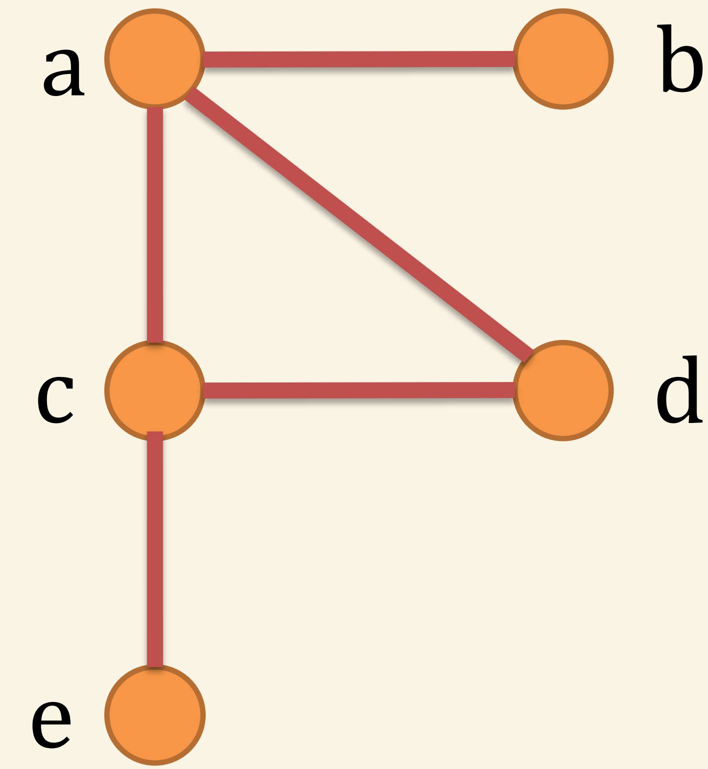
E = {ab, ac, ad, cd}

5. Graph

- **Example 3**

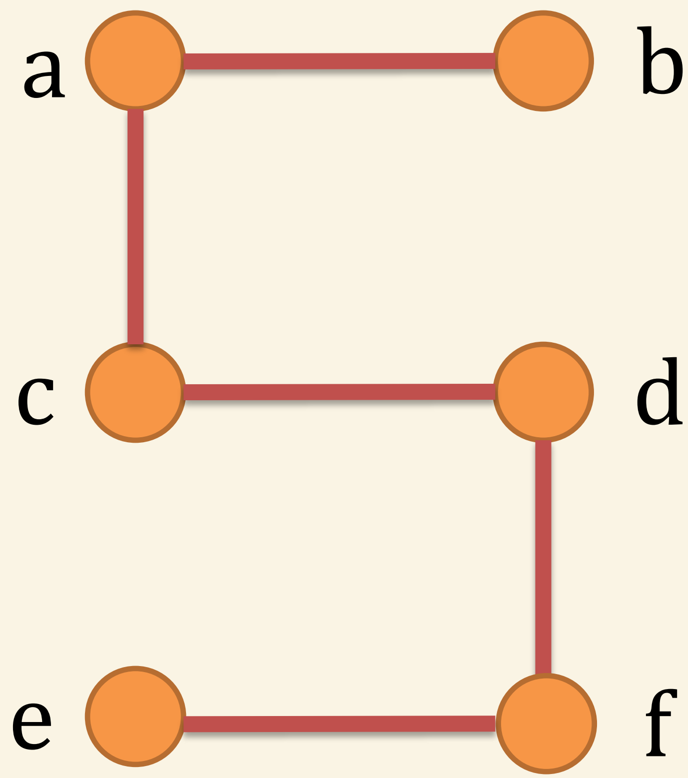
$V = \{a, b, c, d, e\}$

$E = \{ab, ac, ad, cd, ce\}$



Question

- What is the set of **vertex (V)** and **edge** (E) of following graph?

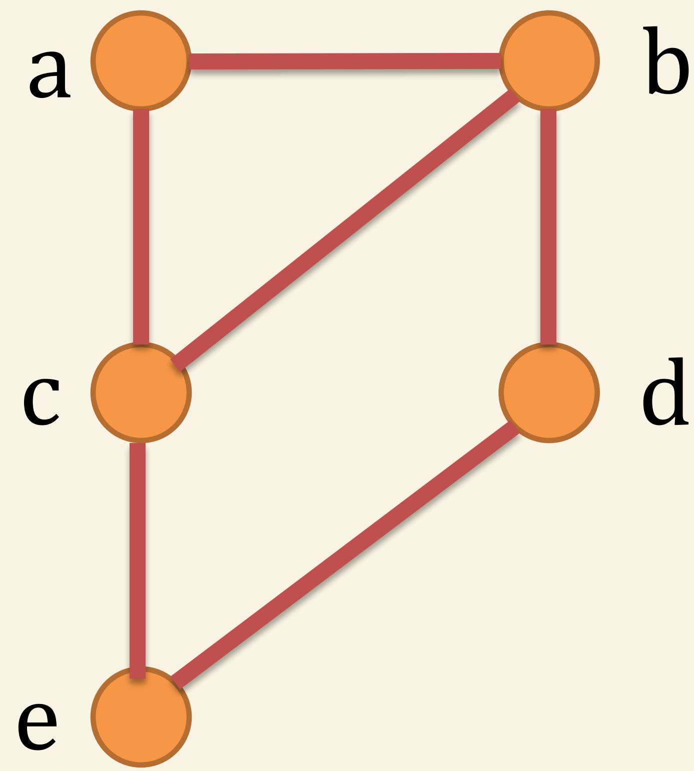


V = {a, b, c, d, e, f}

E = {ab, ac, cd, df, fe}

Question

- What is the set of **vertex (V)** and **edge** (E) of following graph?

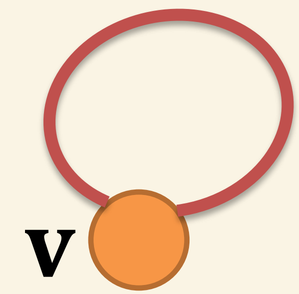


V = {a, b, c, d, e}

E = {ab, ac, cb, ce, bd, de}

6. Loop

- In a graph, if an edge is drawn from vertex to itself, it is called a **loop**.
- In the graph, V is a vertex for which it has an edge (V, V) forming a **loop**.



6. Loop

- **Example 2**:

  In this graph, there are two loops which are formed at **vertex a**, and **vertex b**.

Degree of Vertex

Degree of Vertex

- It is the number of vertices incident with the **vertex V**.

Notation : **deg(V)**

- In a simple graph with **n** number of vertices, the degree of any vertices is

$$\mathbf{deg(v) \leq n - 1}$$

Degree of Vertex

- A vertex can form an edge with all other vertices not including by **itself**.
- So the degree of a vertex will be up to the number of vertices in the **graph minus 1**.
- This 1 is for the **self-vertex** as it cannot form a **loop by itself**. If there is a loop at any of the vertices, then it is not a Simple Graph.

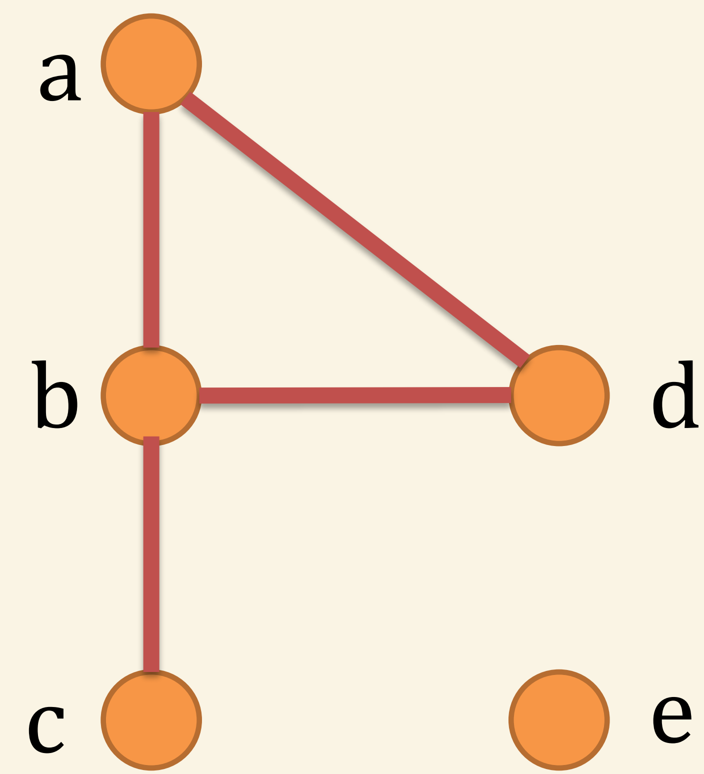
Degree of Vertex

- Degree of vertex can be considered under two cases of graphs

1. **Undirected Graph**
2. **Directed Graph**

Degree of vertex in Undirected Graph

- An undirected graph has no **directed edges**.
- Consider the following examples.

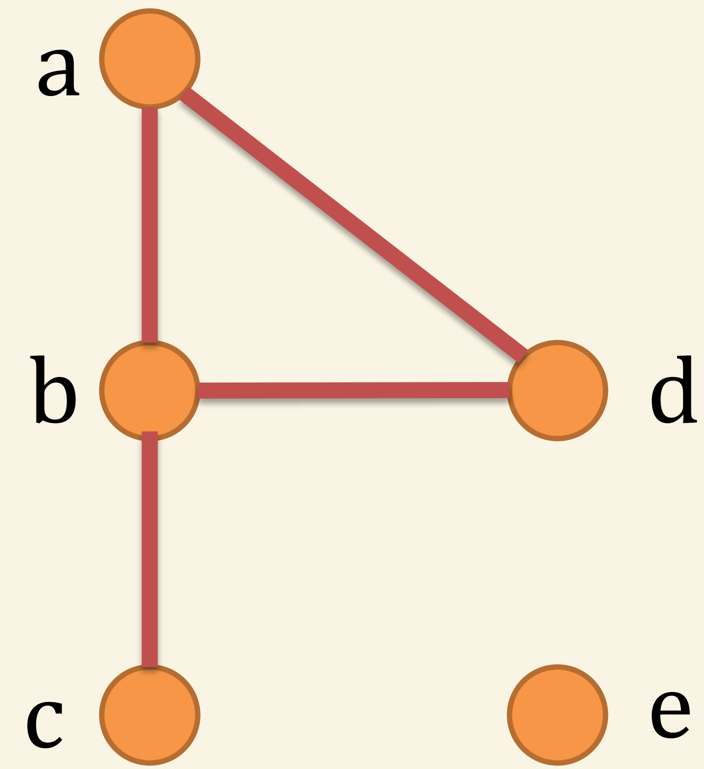


Degree of vertex in Undirected Graph

In the Undirected Graph,

- **deg(a)** = 2, as there are 2 edges meeting at vertex ‘a’.
- **deg(b)** = 3, as there are 3 edges meeting at vertex ‘b’.
- **deg(c)** = 1, as there is 1 edge made at vertex ‘c’

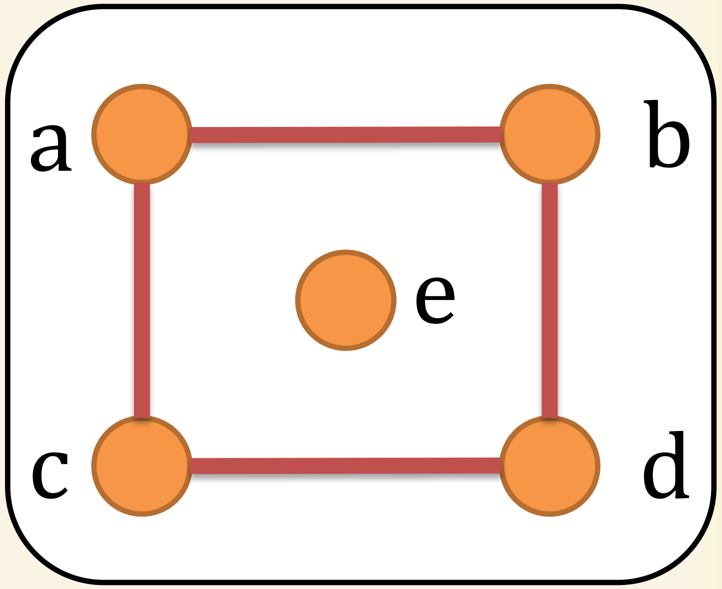
  So ‘**c**’ is a **pendent vertex**.
- **deg(d)** = 2, as there are 2 edges meeting at vertex ‘d’.
- **deg(e)** = 0, as there are 0 edges formed at vertex ‘e’.
- So ‘e’ is **an isolated vertex**.



Degree of vertex in Undirected Graph

In the Undirected Graph,

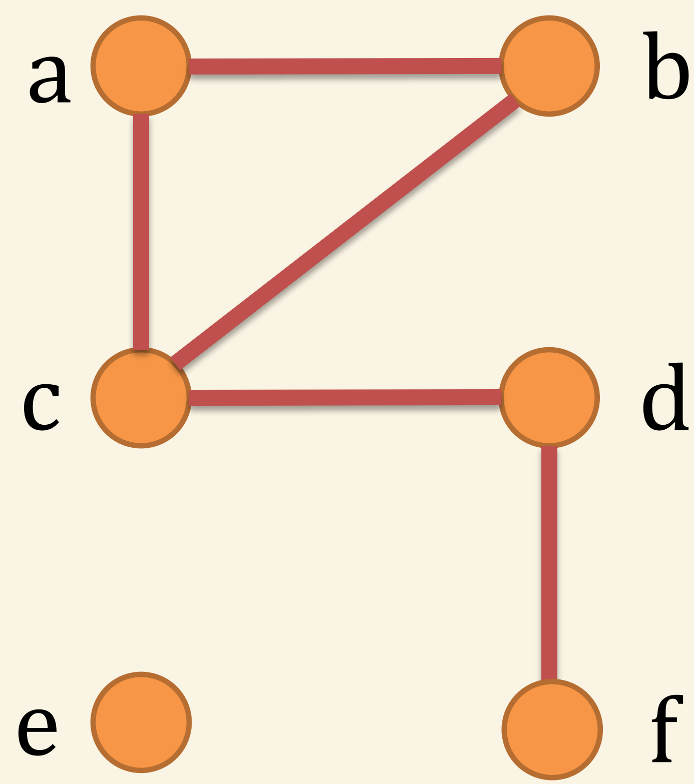
- **deg(a)** = 2, as there are 2 edges meeting at vertex 'a'.
- **deg(b)** = 2, as there are 2 edges meeting at vertex 'b'.
- **deg(c)** = 2, as there is 2 edge made at vertex 'c'
- **deg(d)** = 2, as there are 2 edges meeting at vertex 'd'.
- **deg(e)** = 0, as there are 0 edges formed at vertex 'e'.



So 'e' is **an isolated vertex**.

Question

- What is the Degree of vertex of following undirected graph?



deg(a) = 2

deg(b) = 2

deg(c) = 3

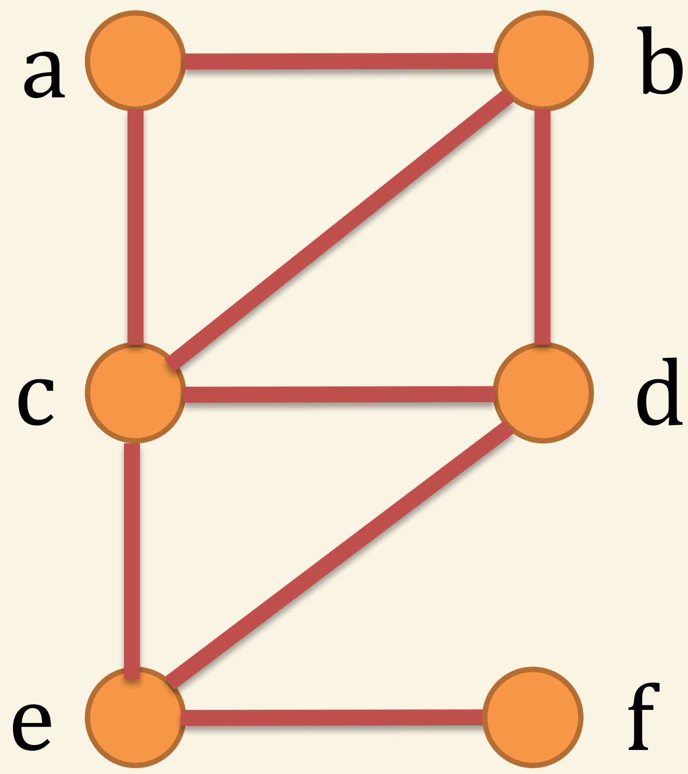
deg(d) = 2

deg(f) = 1

deg(e) = 0

Question

- What is the Degree of vertex of following undirected graph?



deg(a) = 2

deg(b) = 3

deg(c) = 4

deg(d) = 3

deg(e) = 3

deg(f) = 1

Degree of vertex in Directed Graph

- In a directed graph, each vertex has an **indegree** and an **outdegree**.
- **Indegree of a Graph: -**
- **Indegree** of vertex V is the number of edges which are **coming** into the vertex V.

Notation – deg+(V)

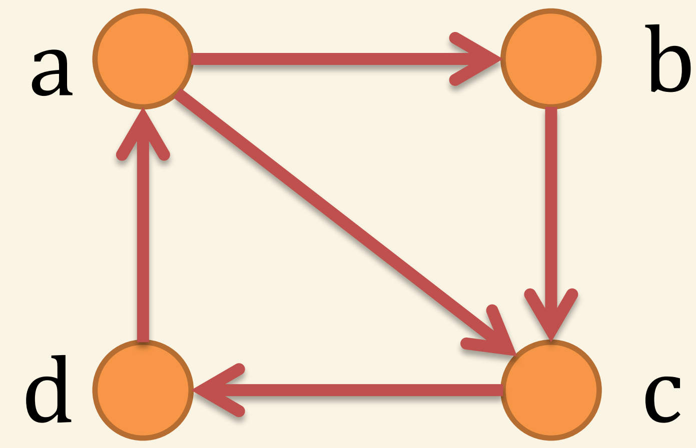
Degree of vertex in Directed Graph

- **Outdegree of a Graph: -**
- Outdegree of vertex V is the number of edges which are **going out** from the vertex V.

Notation – deg-(V)

Degree of vertex in Directed Graph

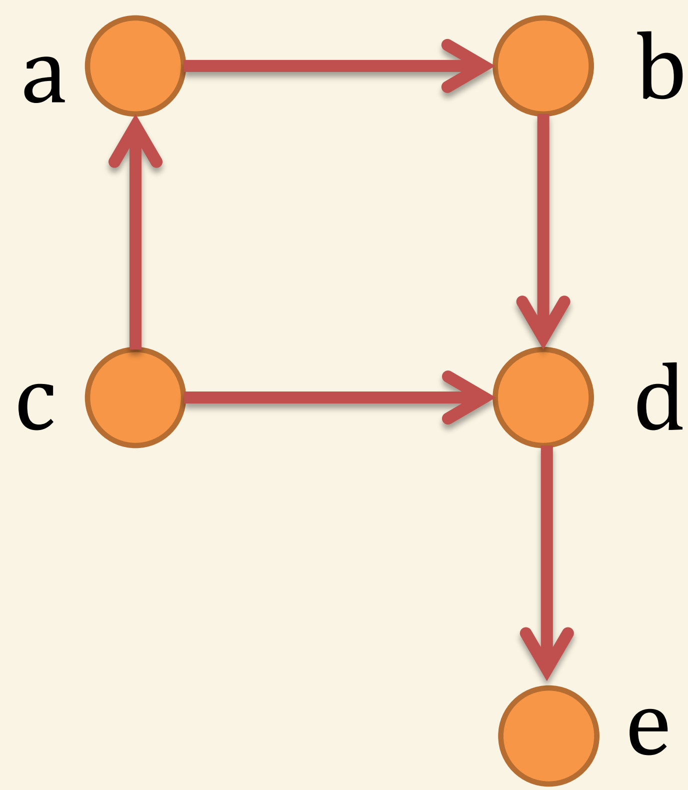
- The **indegree** and **outdegree** of other vertices are shown in the following table



Vertex	Indegree	Outdegree
a	1	2
b	1	1
c	2	1
d	1	1

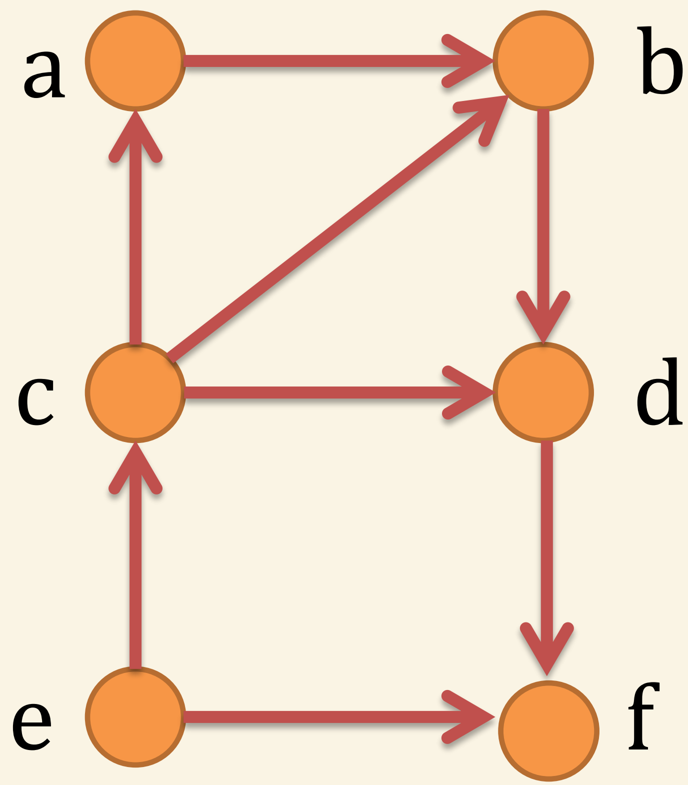
Question

- What is the Degree of vertex of following directed graph?



Question

- What is the Degree of vertex of following directed graph?



Isomorphic Graph

Isomorphic Graph

- **Two graphs** which contain the **same** number of graph **vertices** connected in the same way are said to be isomorphic.
- In short, out of the two **isomorphic graphs**, one is a twisted version of the other. An unlabeled graph also can be isomorphic graph.

Isomorphic Graph

- Two graphs G1 and G2 are said to be isomorphic if –

1. Their number of components (vertices and edges) are same.
2. Their edge connectivity is fixed.
3. They are simple graph (no loop)

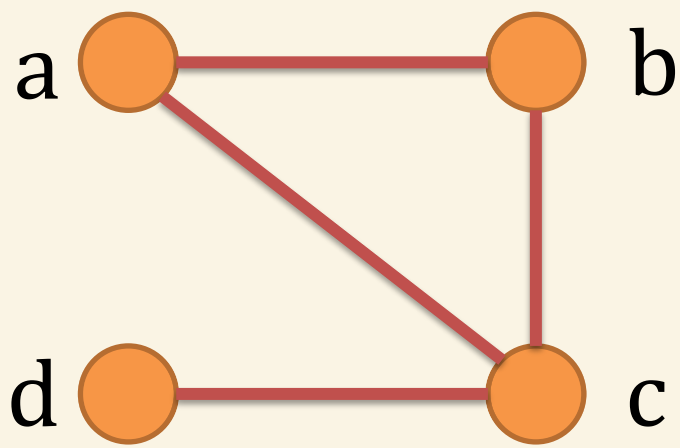
Isomorphic Graph

- Graph **G1** and **G2** are **isomorphic** if it satisfy the below property.

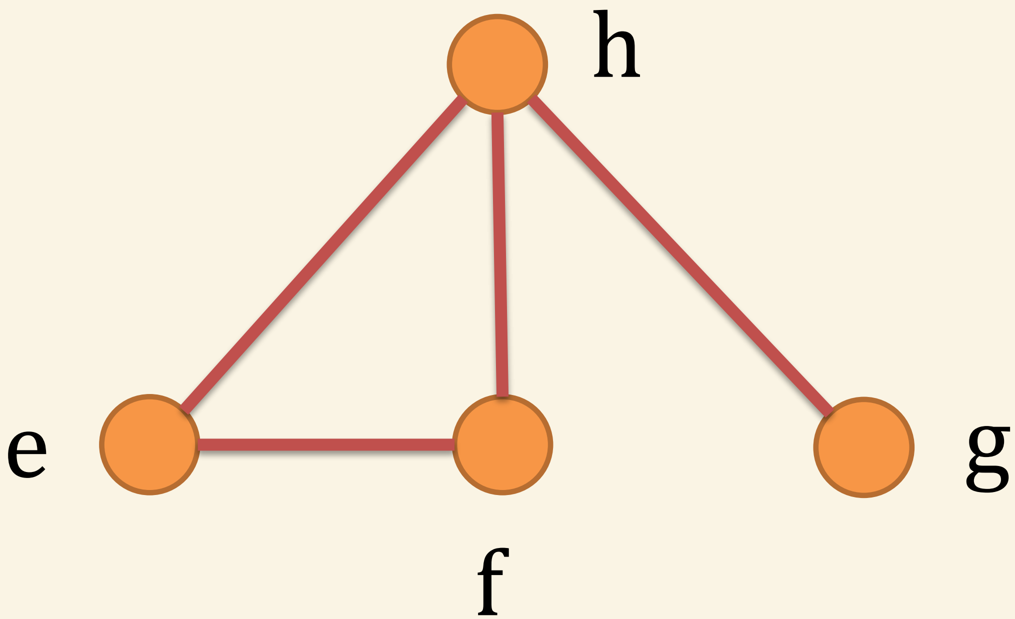
1. |V(G1)| = |V(G2)|
2. |E(G1)| = |E(G2)|
3. **Degree sequence** of G1 and G2 are same.

Isomorphic Graph

Q1. Which of the following graph are isomorphic?



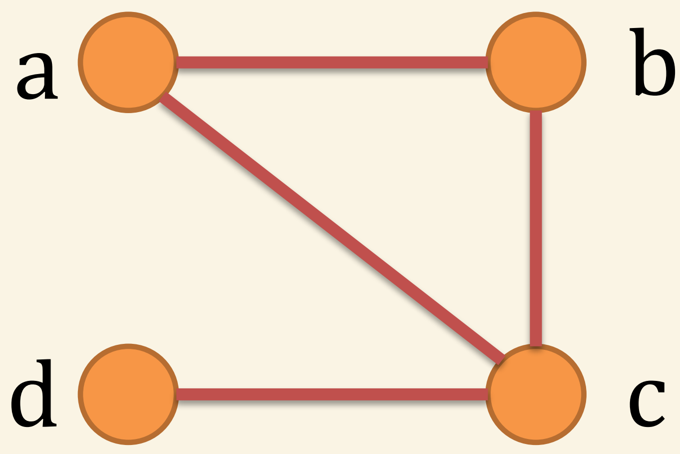
(G1)



(G2)

Isomorphic Graph

Q1. Which of the following graph are isomorphic?



(G1)

V(G1) = {a, b, c, d}

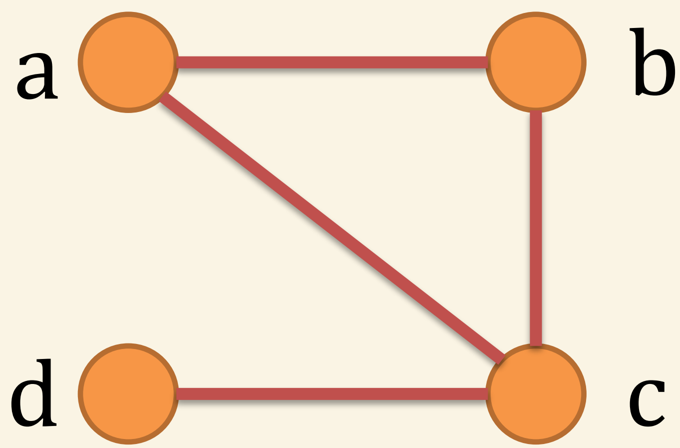
|V(G1)| = 4

E(G1) = {ab, ac, cb, cd}

|E(G1)| = 4

Isomorphic Graph

Q1. Which of the following graph are isomorphic?



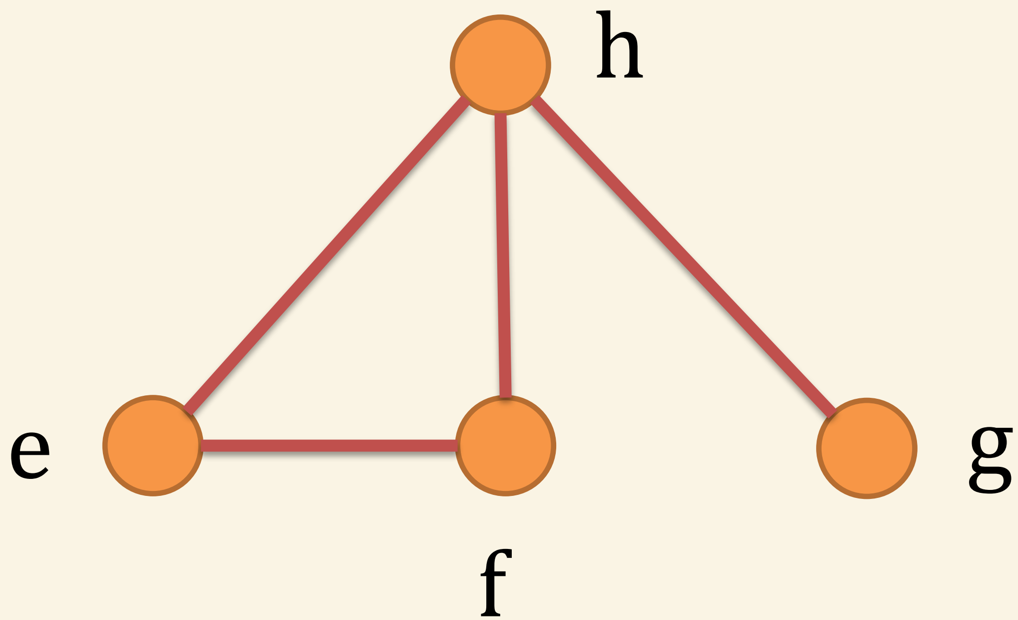
(G1)

Degree sequence

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

Isomorphic Graph

Q1. Which of the following graph are isomorphic?



(G2)

$V(G2) = \{e, f, h, g\}$

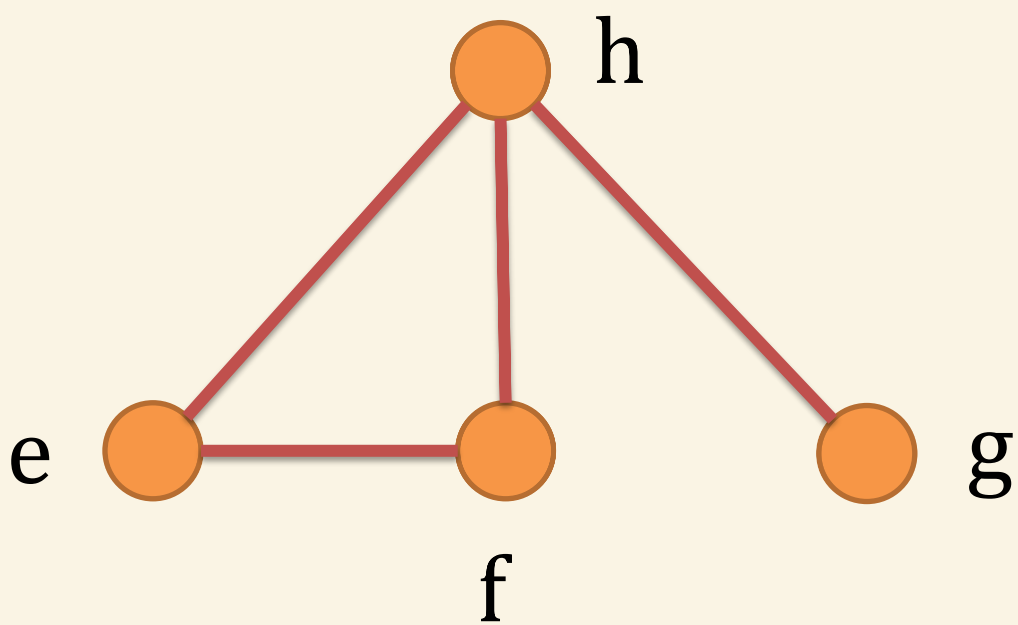
$|V(G2)| = 4$

$E(G2) = \{ef, eh, hg, hf\}$

$|E(G2)| = 4$

Isomorphic Graph

Q1. Which of the following graph are isomorphic?



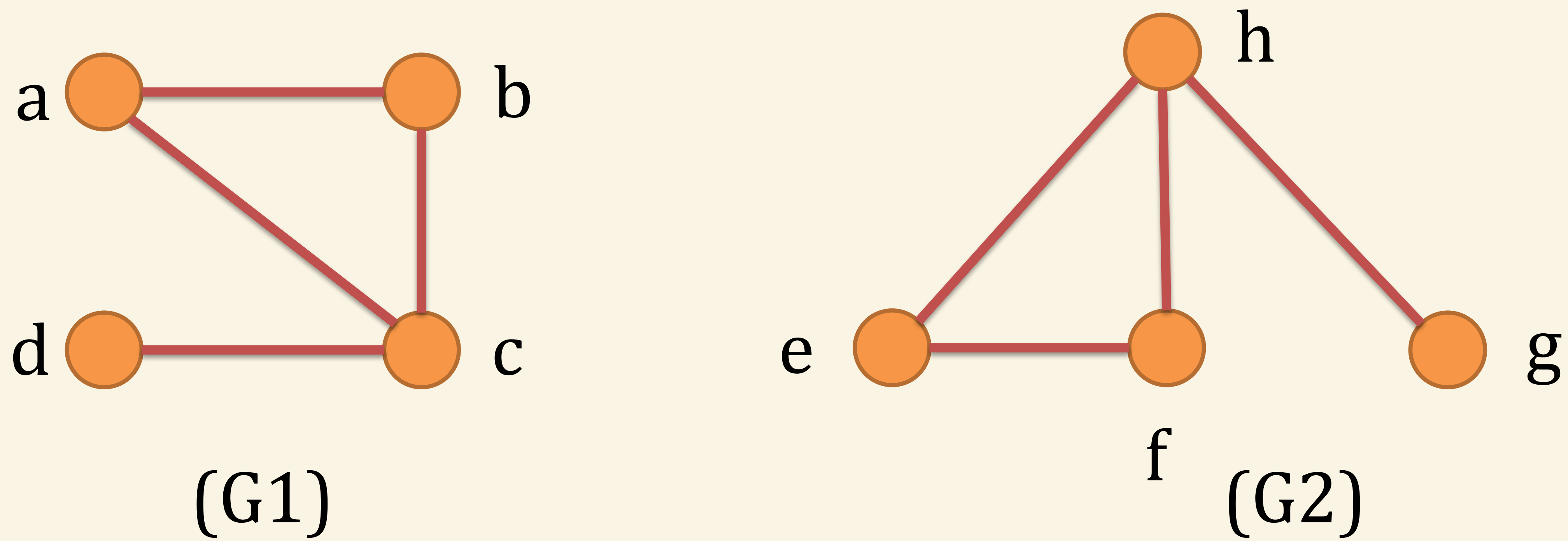
(G2)

Degree sequence

Vertex	e	f	h	g
Connecting to Vertex	h,f	e,h	e,f,g	h
Degree	2	2	3	1

Isomorphic Graph

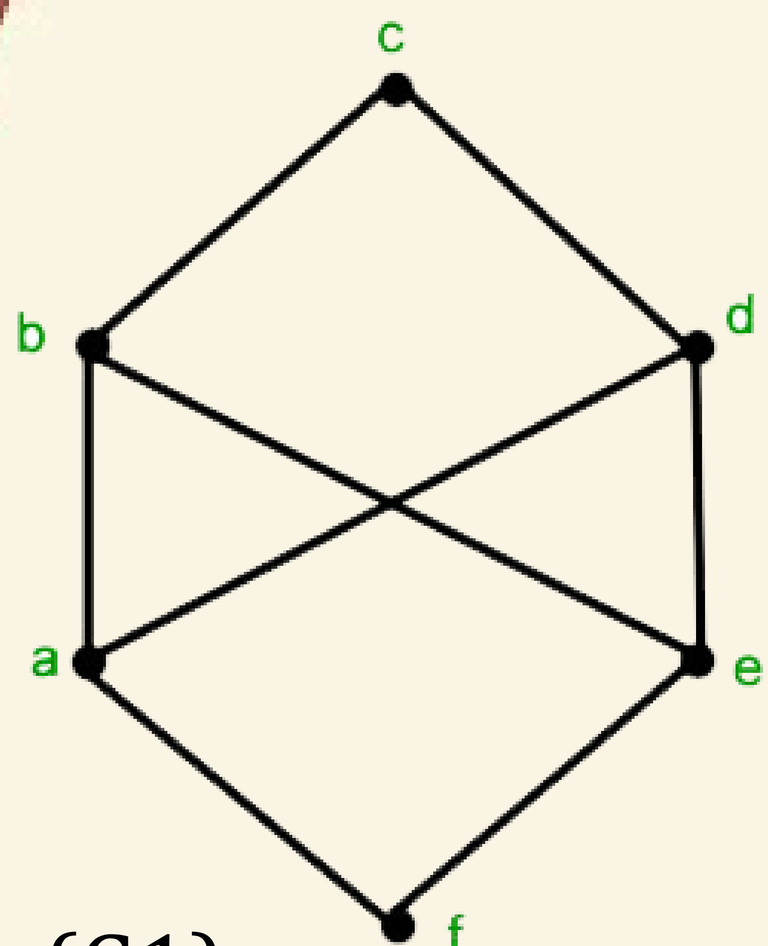
Q1. Which of the following graph are isomorphic?



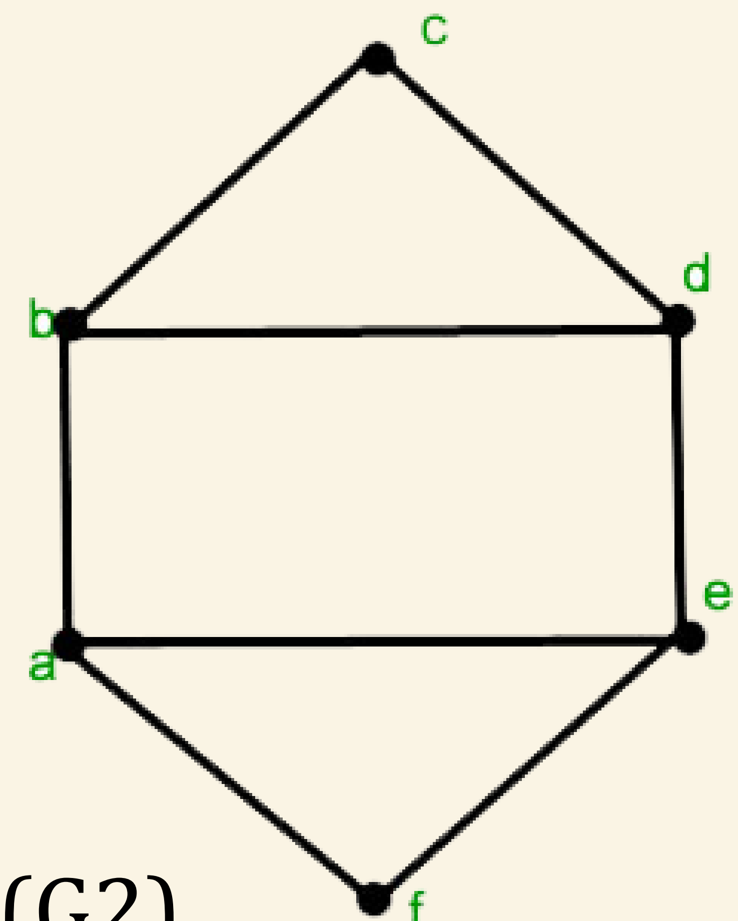
Here graph G1 and G2 are isomorphic graph. Because both graph satisfied the isomorphic properties.

Question

Q1. Which of the following graph are isomorphic?



(G1)



(G2)

$V(G1) = ?$

$|V(G1)| = ?$

$E(G1) = ?$

$|E(G1)| = ?$

$V(G2) = ?$

$|V(G2)| = ?$

$E(G2) = ?$

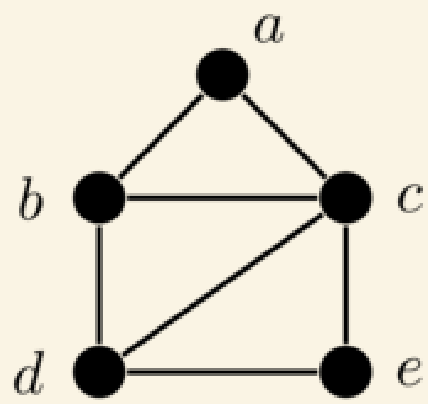
$|E(G2)| = ?$

Subgraph

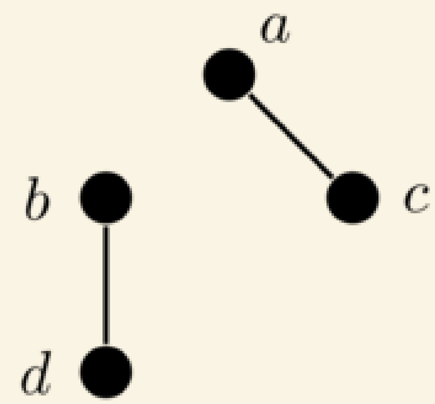
Subgraph

- A **subgraph** ‘S’ of graph ‘G’ is a graph whose set of **vertices** and **set** of **edges** are all **subsets** of ‘G’.
- Since every set is a subset of itself, every graph is subgraph of itself.

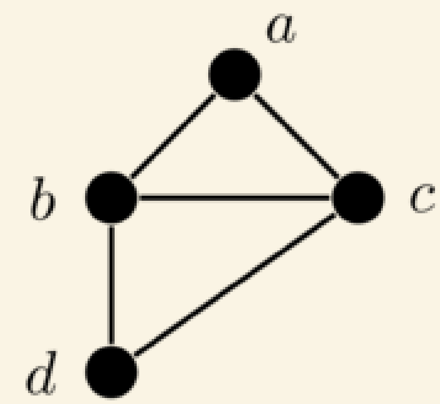
Subgraph



(G)



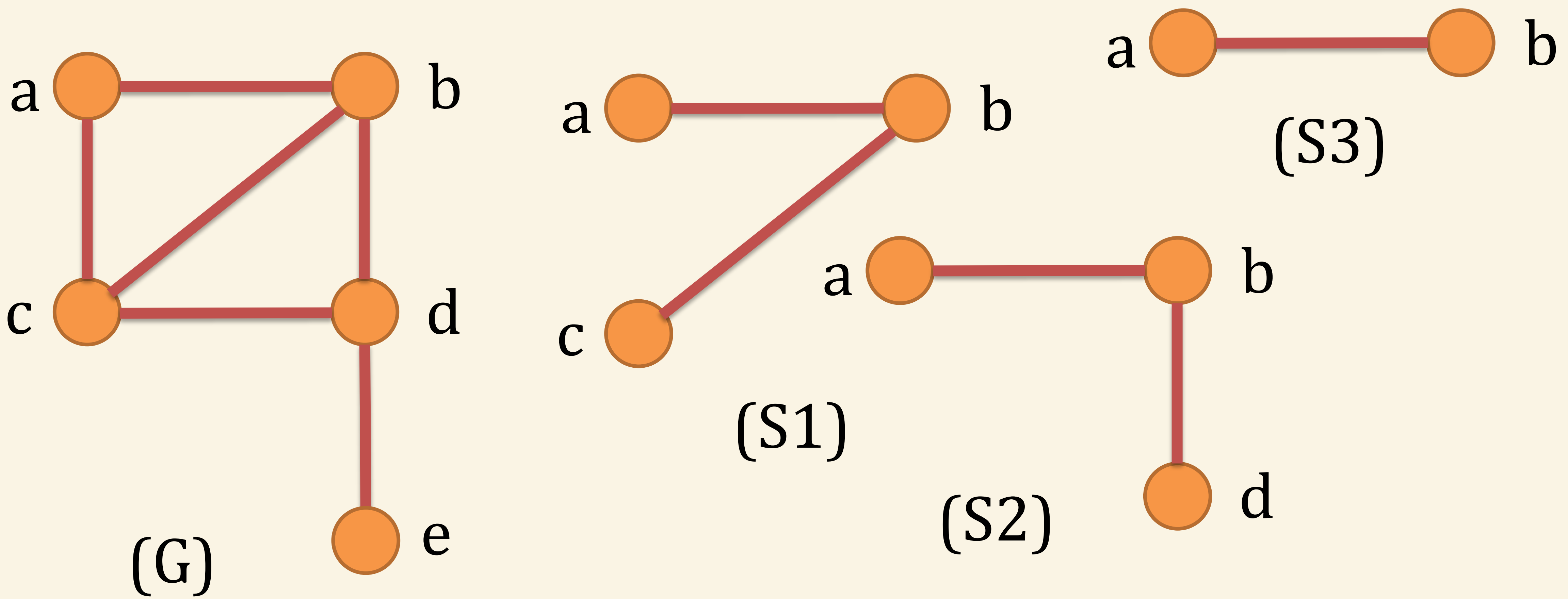
(S1)



(S2)

Question

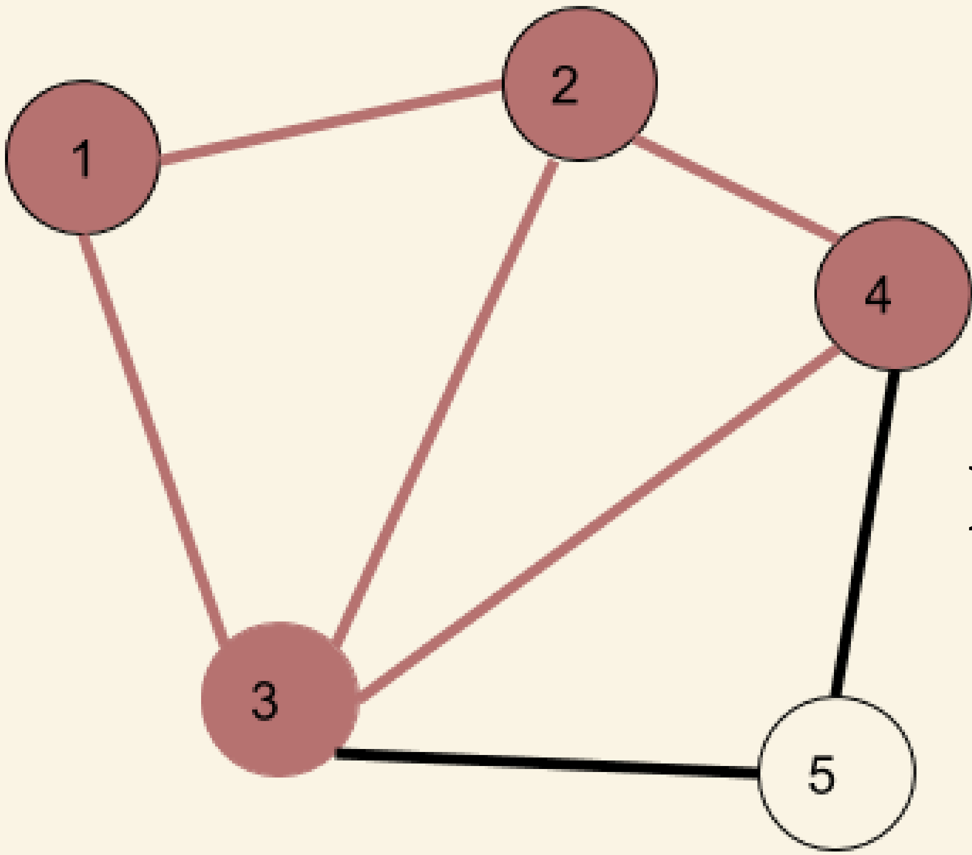
Q1. Create three subgraphs for graph G?



Walk

Walk

- A walk is a sequence of **vertices** and **edges** of a **graph** i.e. if **we traverse** a graph then we get a **walk**.

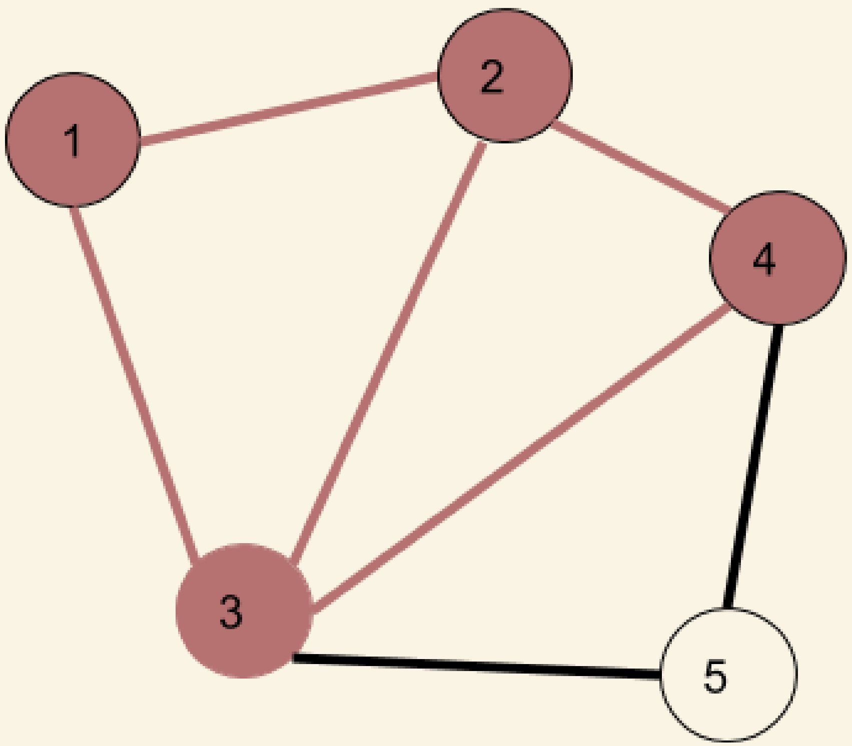


- **Vertex can be repeated**
- **Edges can be repeated**

Here 1->2->3->4->2->1->3 is a **walk**

Walk

- **Open walk** - A walk is said to be an open walk if the **starting** and **ending vertices** are **different** i.e. the origin vertex and terminal vertex are different.



Here,

1->2->3->4->5->3

is an open walk.

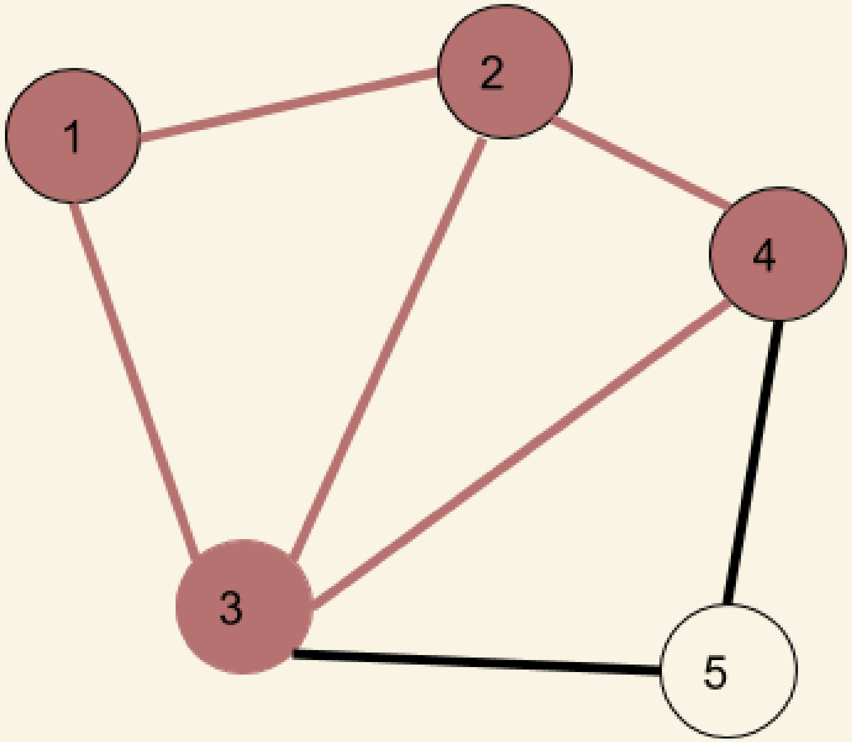
Walk

- **Closed walk** - A walk is said to be a **closed walk** if the **starting** and **ending vertices** are **identical** i.e. if a walk starts and ends at the same vertex, then it is said to be a **closed walk**.

Here,

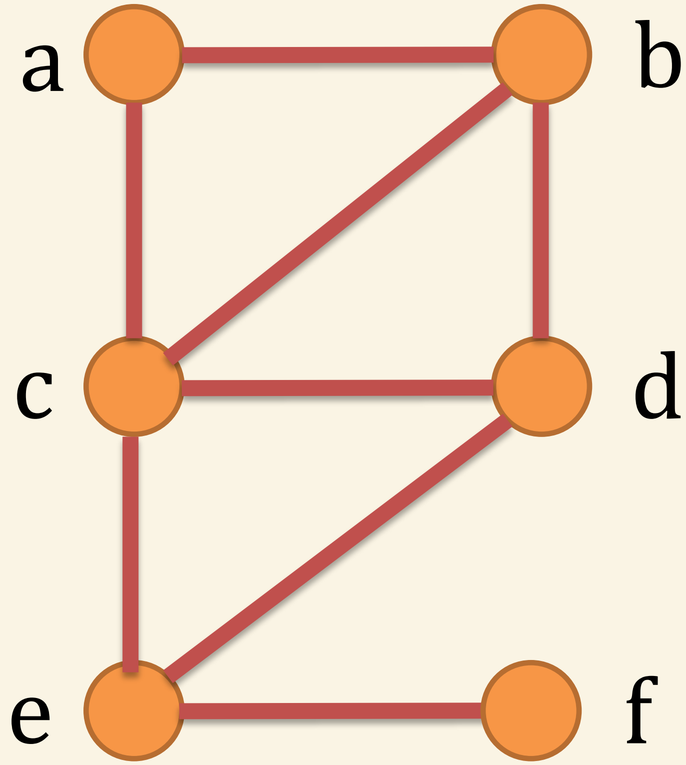
1->2->3->4->5->3->1

is an closed walk.



Question

Q1. What is the open (a-f) and closed walk for graph G?



(G)

Open Walk- a->c->e->f

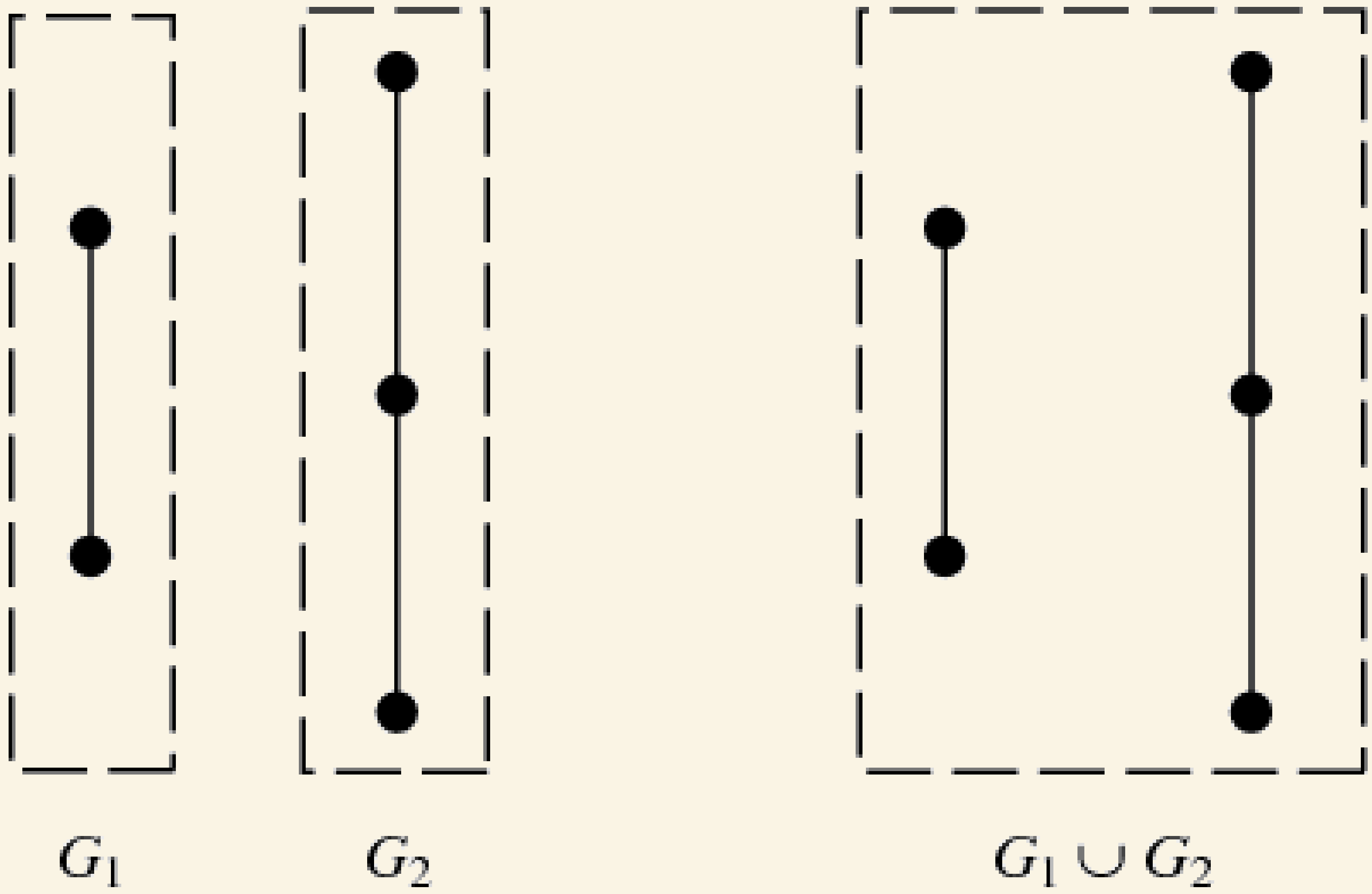
Closed Walk- a->c->e->d->b->a

Operation on Graph

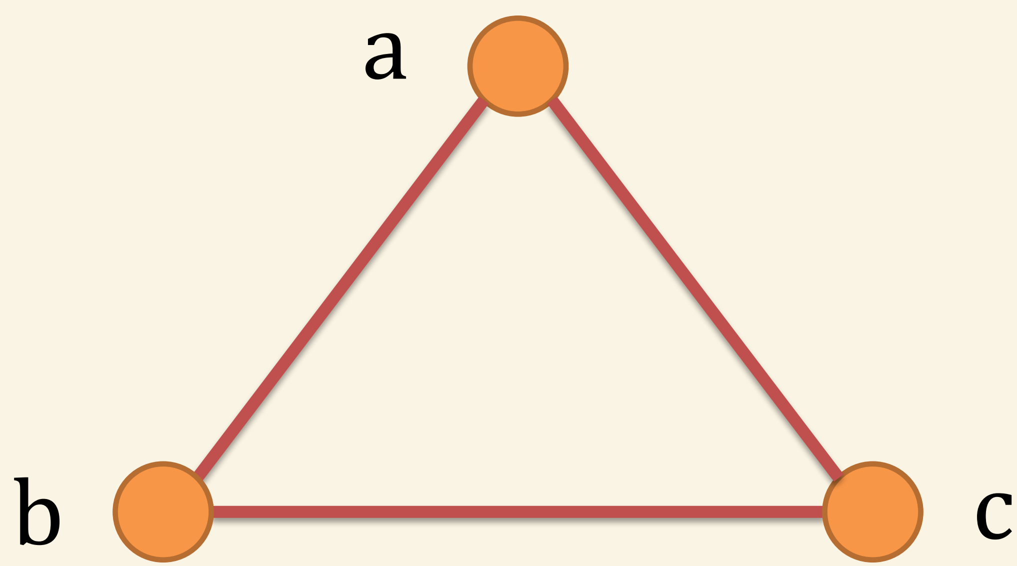
Operation on Graph

- **Union of Sets:** Union of Sets A and B is defined to be the set of **all** those **elements** which belong to A or B or both and is denoted by **$A \cup B$**.
- **Intersection of Sets:** Intersection of two sets A and B is the set of all those elements which **belong** to **both** A and B and is denoted by **$A \cap B$**.

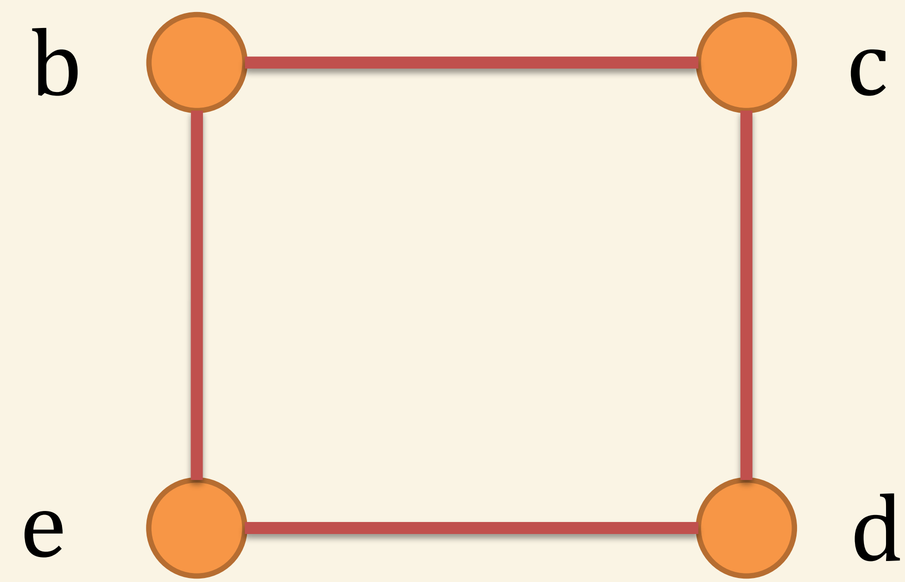
Operation on Graph



Operation on Graph

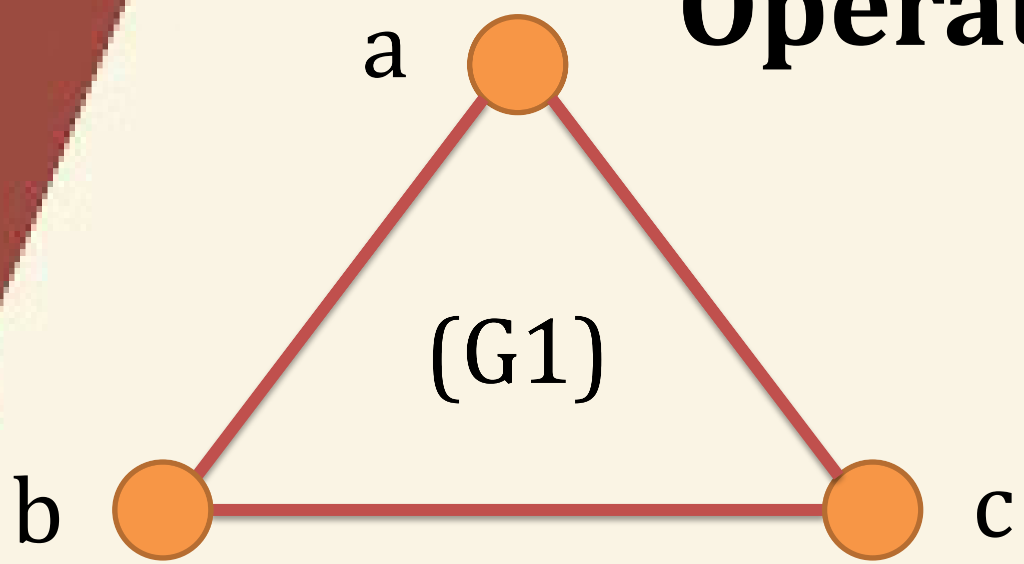


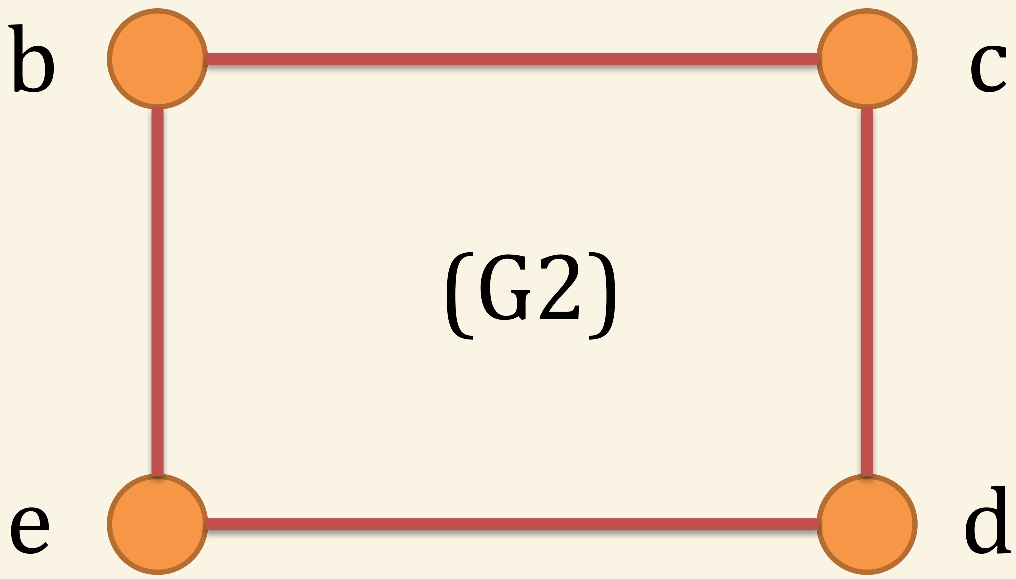
(G1)



(G2)

Operation on Graph





G1 Graph Vertex and Edge Set

V (G1) = {a, b, c}

E (G1) = {ab, ac, bc }

G2 Graph Vertex and Edge Set

V (G2) = {b, c, d, e}

E (G2) = {bc, be, de, dc}

Operation on Graph (Union)

Union of Vertex (G1 and G2)

V(G1) = {a, b, c}

V(G2) = {b, c, d, e}

V(G1) ∪ V(G2) = {a, b, c, d, e}

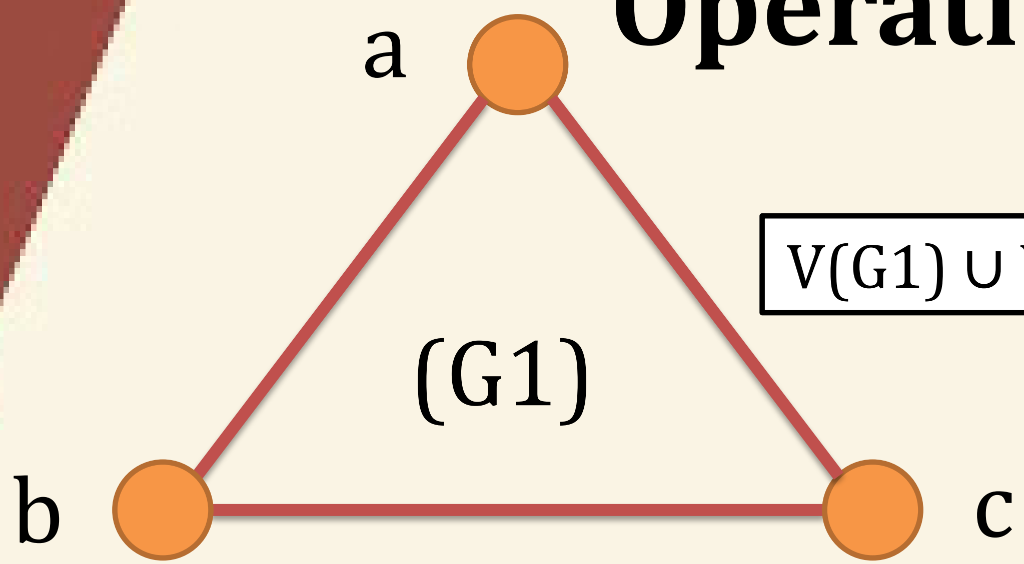
Union of Edge (G1 and G2)

E(G1) = {ab, ac, bc }

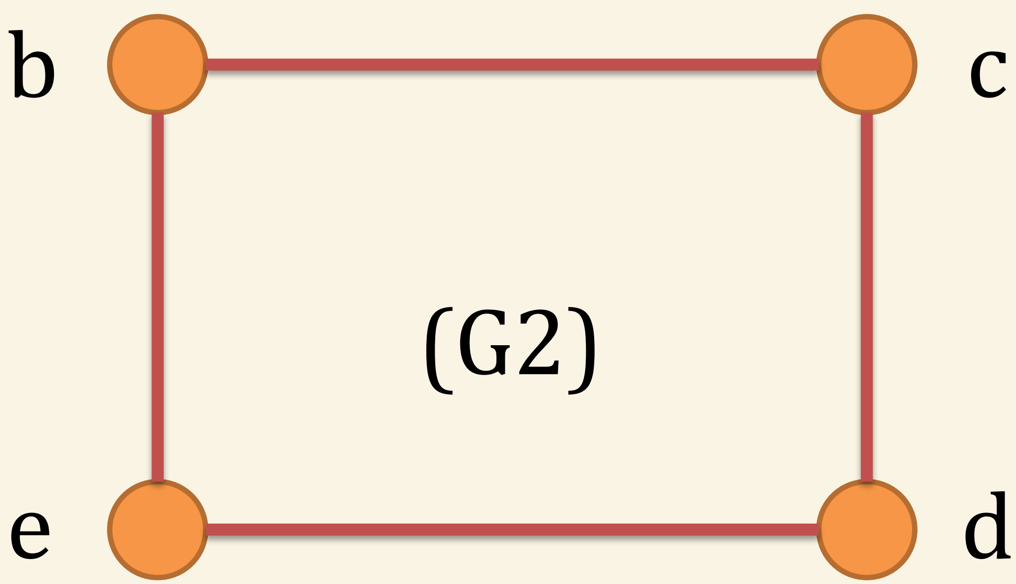
E (G2) = {bc, be, de, dc}

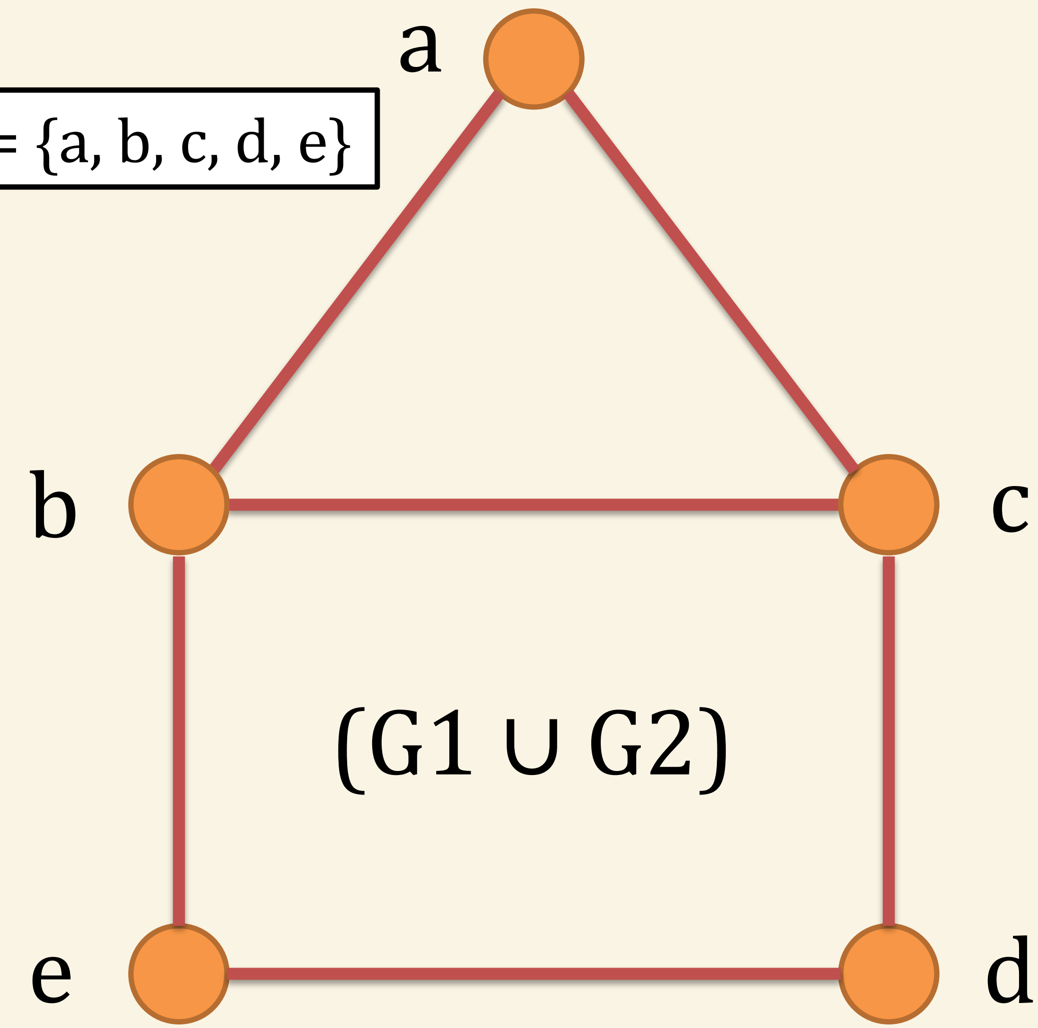
E(G1) ∪E(G2) = {ab, ac, bc, be, de, dc}

Operation on Graph (Union)



V(G1) ∪ V(G2) = {a, b, c, d, e}





E(G1) ∪E(G2) = {ab, ac, bc, be, de, dc}

Operation on Graph (Intersection)

Intersection of Vertex (G1 and G2)

V(G1) = {a, **b**, **c**}

V(G2) = {**b**, **c**, d, e}

V(G1) ∩ V(G2) = {b, c}

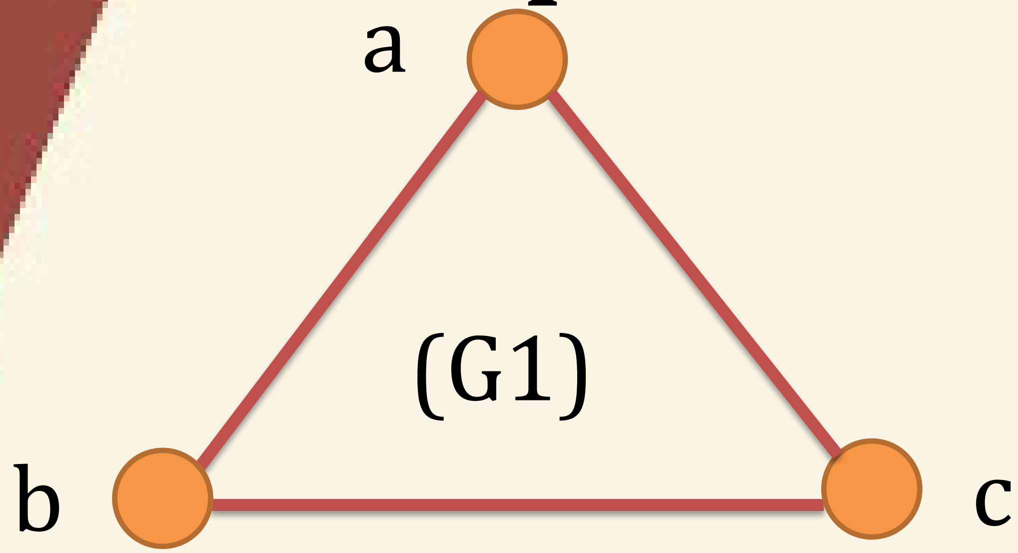
Intersection of Edge (G1 and G2)

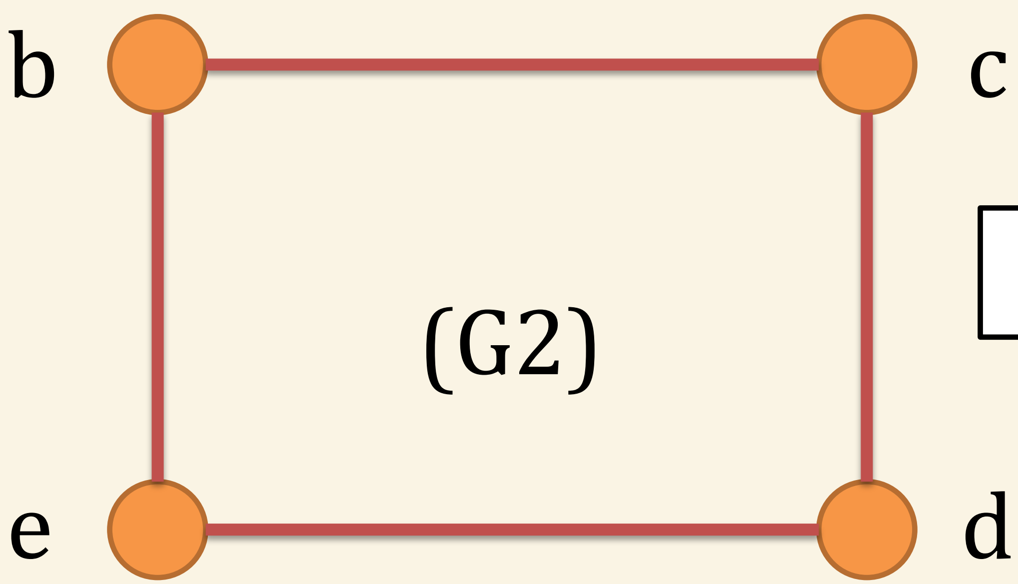
E(G1) = {ab, ac, **bc** }

E (G2) = {**bc**, be, de, dc}

E(G1) ∩ E(G2) = {bc}

Operation on Graph (Intersection)





(G1 ∩ G2)

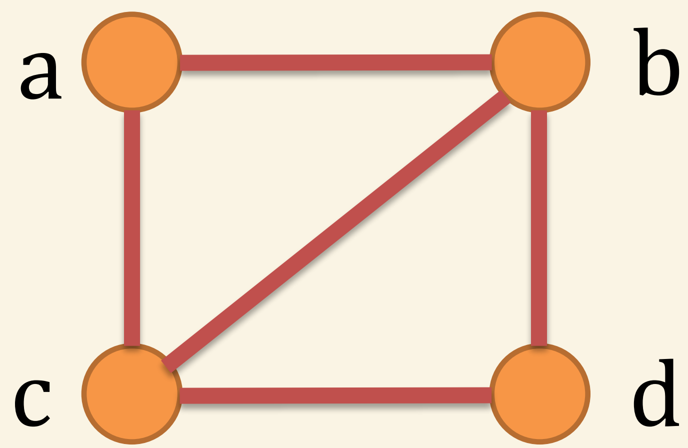
V(G1) ∩ V(G2) = {b, c}

E(G1) ∩ E(G2) = {bc}

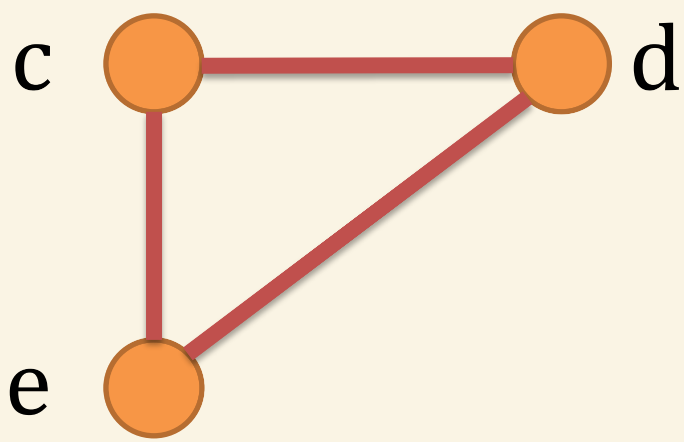
d

Question

Q1. Perform the **Union** and **Intersection operation** and **draw the graph** after **performing** the **operation** on following graph G1 and G2.



(G1)



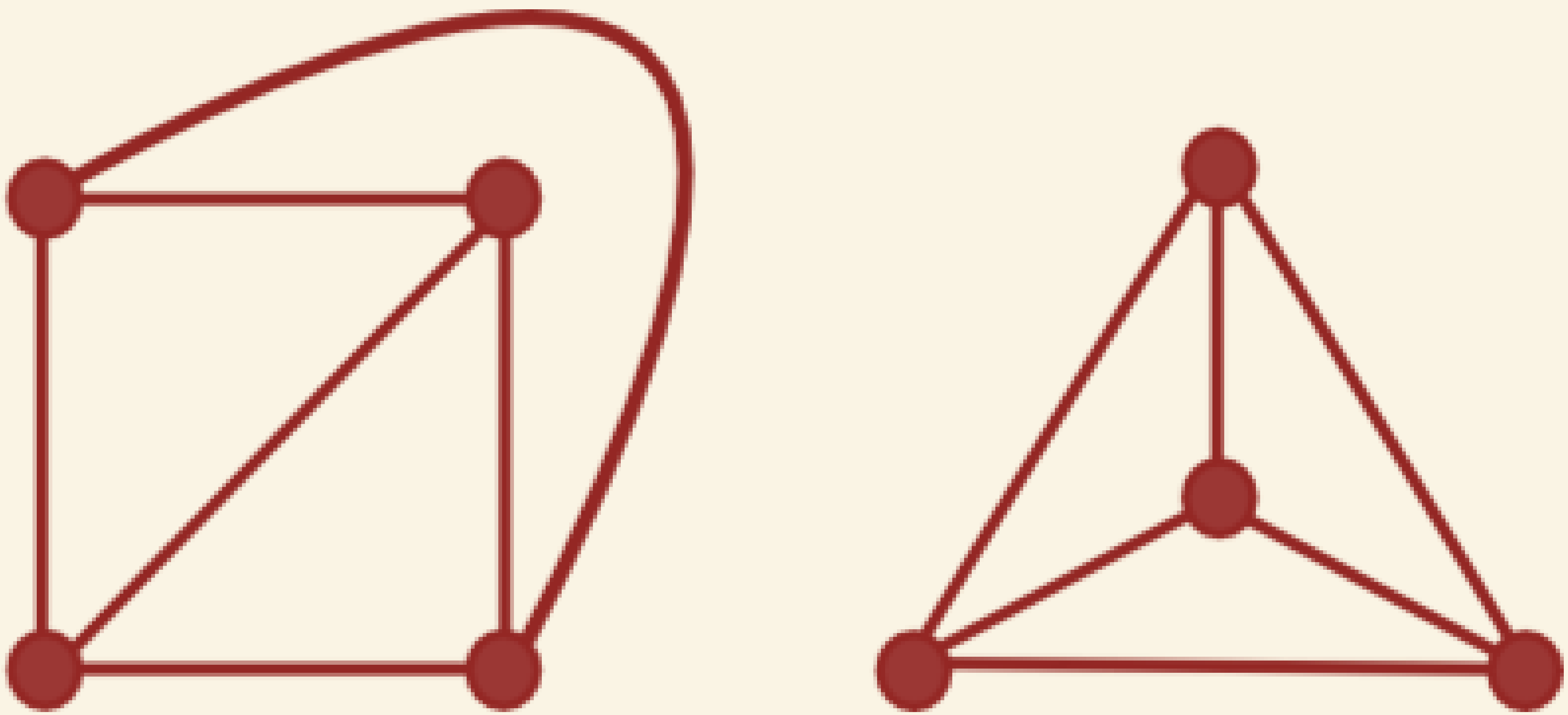
(G2)

Planar Graph

Planar Graph

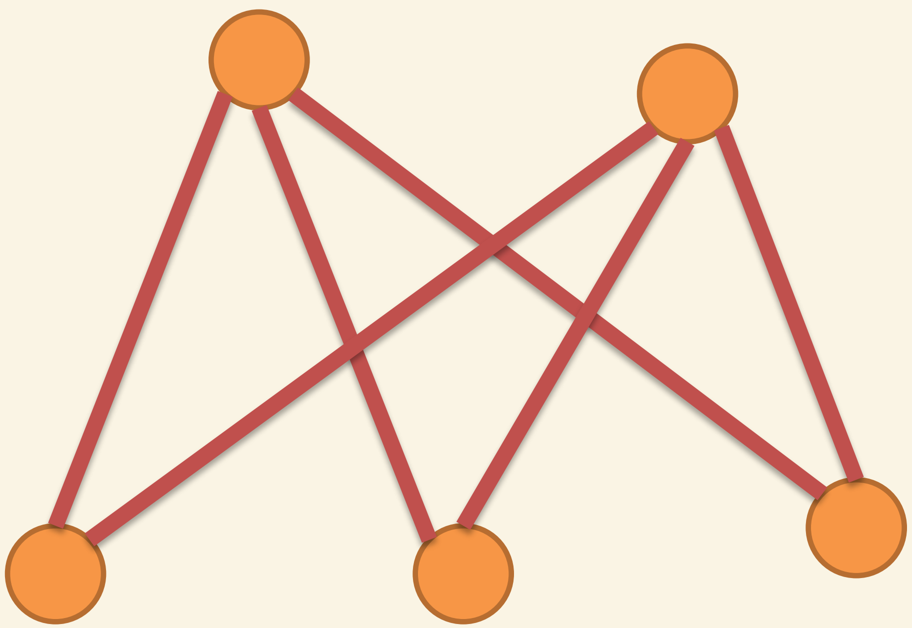
- When a **connected graph** can be drawn without **any edges crossing**, it is called **planar**.
- Such a drawing is called a **planar representation** of the graph.

Planar Graph

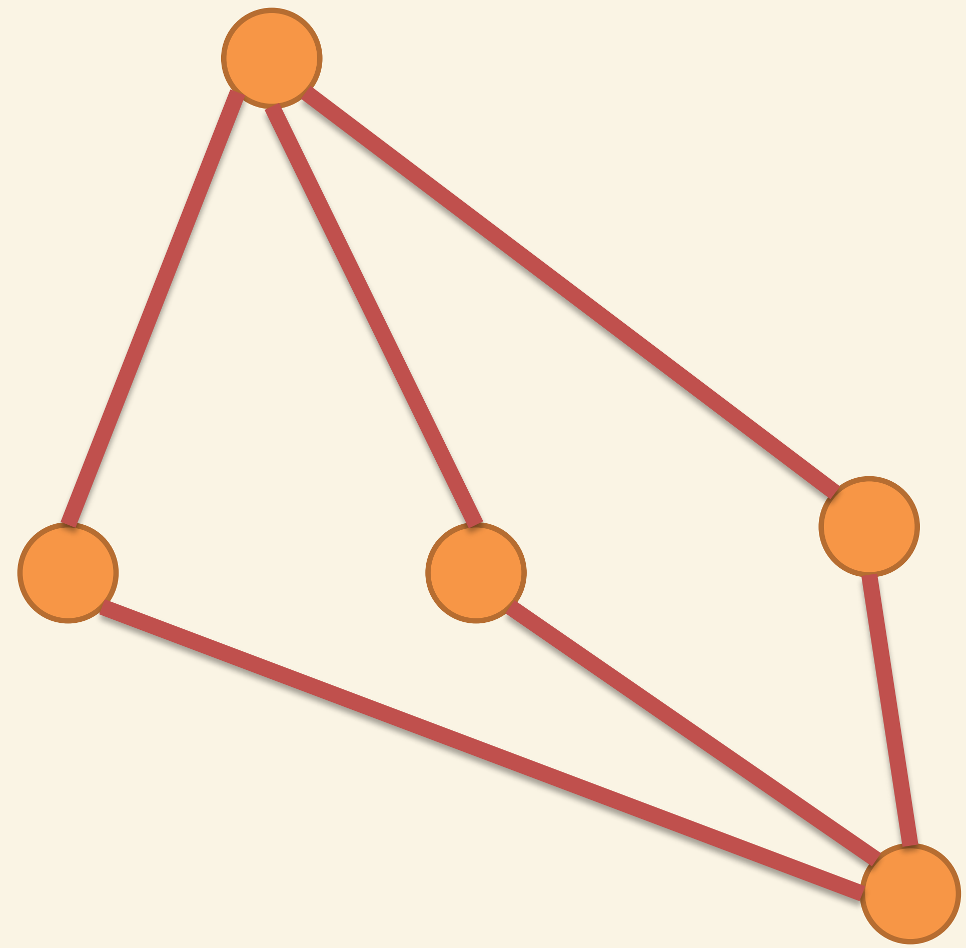


Planar Graph

Example – Is the graph **G1** is planar?

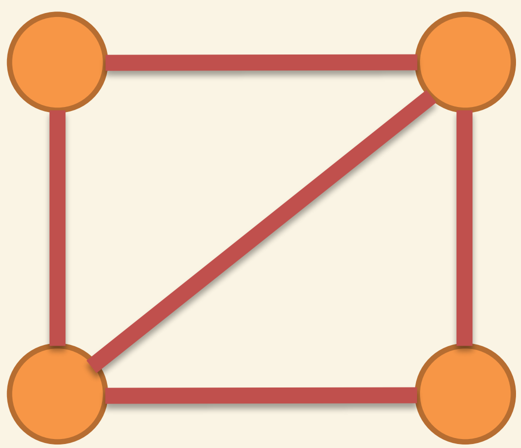


G1

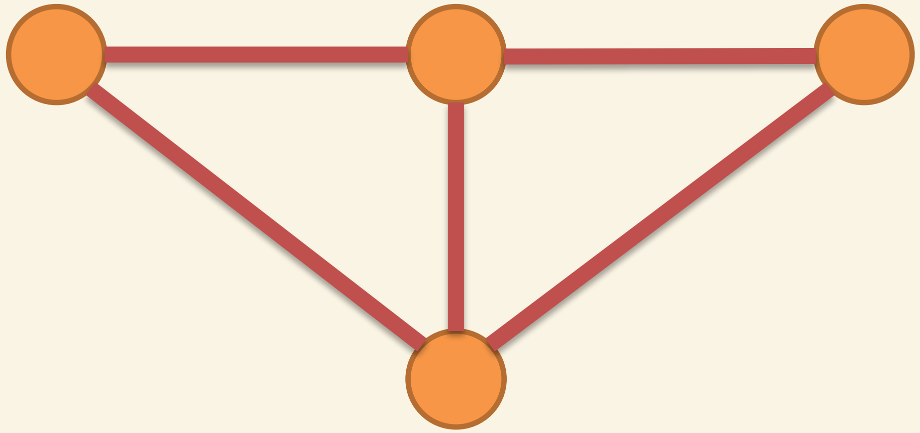


Question

Q1. Draw, two different **planar graphs** with the same number of vertices and edges.



(G1)



(G2)

Thank You