Finals -Lecture Notes #3 – More on Graph Theory, Spanning Tree, & Kruskal Algorithm 05.03.2023

REMINDER!

1) ALWAYS FOLLOW MY CLASSROOM PROCEDURES.

2) ACTIVITIES ARE ALWAYS UPDATED IN GOOGLE CLASSROOM, MYOPENMATH

3) ALWAYS SUBMIT YOUR WORK ON TIME. IF YOU MISSED SOME ACTIVITIES, BE RESPONSIBLE ENOUGH TO DO THEM.

4) THE WEEKLY HOMEWORK IS ALWAYS DUE EVERY MONDAY.

5) CHECK THE UPDATES ON GOOGLE CLASSROOM AND DO YOUR MISSING ASSIGNMENTS as the GRADES are always UPDATED.

Activities for today

MyOpenMath

1) **Bellwork#3 (Finals) – Check Your Knowledge 05.03.23.**
2) **Activity # 3 (Finals) – Graph Theory & Spanning Tree 05.03.23**
3) **Finals - Weekly Homework #3 – Let Me Review What I Learned 05.03-05.9.23**
4) **Finals – Lecture Notes #3** More on Graph Theory & Spanning Tree 05.03.2023

At the end of the lesson, students will be able to

1) **identify graph theory.**
2) **problems associated with the graph theory and spanning tree**

Graph Theory

In the domain of mathematics and computer science, ***graph theory is the study of graphs that concerns with the relationship among edges and vertices.*** It is a popular subject having its applications in computer science, information technology, biosciences, mathematics, and linguistics to name a few.

What is a Graph?

We begin by considering Figs. 1.1 and 1.2, which depict part of a road map and part of an electrical network.

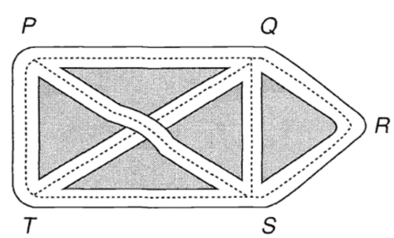


Fig. 1.1

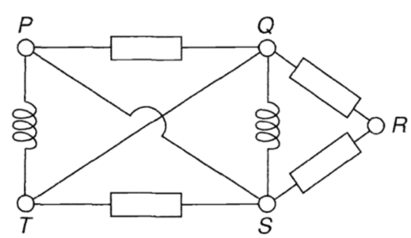
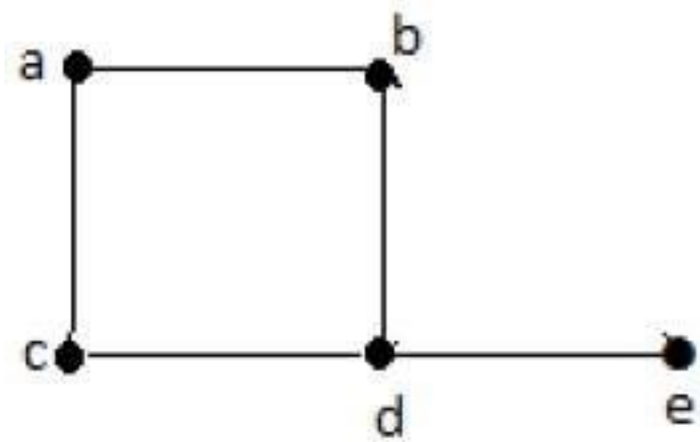


Fig. 1.2

Either of these situations can be represented diagrammatically by means of points and lines, as in Fig. 1.3. The points P, Q, R, S and T are called ***vertices***, the lines are called ***edges***, and the whole diagram is called a graph. Note that the ***intersection*** of the lines ***PS and QT*** is not a vertex since it does not correspond to cross-roads or to the meeting of two wires. The ***degree of a vertex*** is the number of edges with that vertex as an endpoint; it corresponds in Fig. 1.1 to the number of roads at an intersection. For example, **the degree** of the vertex Q is 4.

Key Idea. 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***.

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***. Take a look at the following graph.

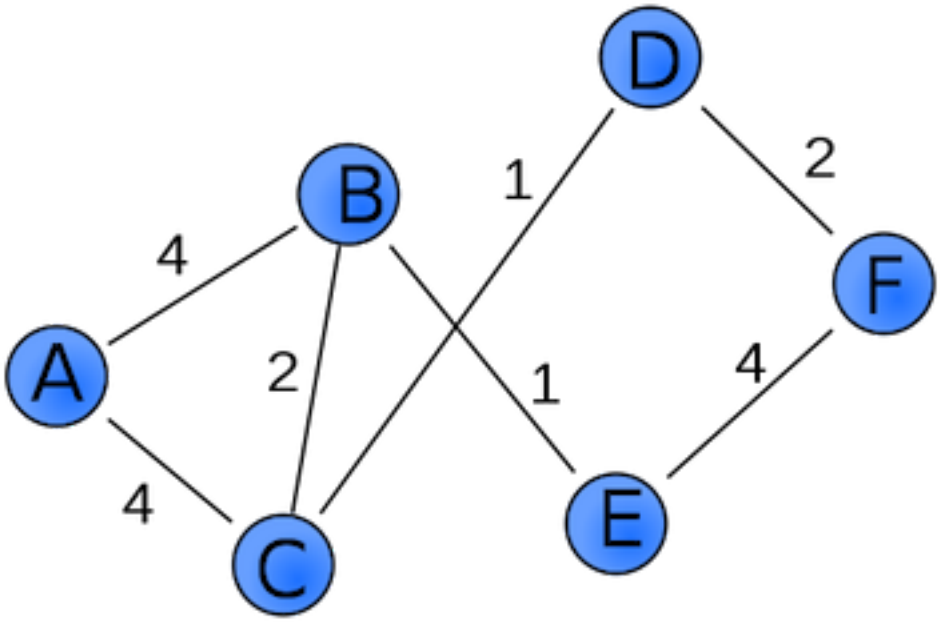


In the above graph,

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

Edges, E = {ab, ac, bd, cd, de}

A **weighted graph** is a graph with weighted edges. The weights may represent factors like cost, or the distance required to travel between nodes.



Example of a weighted graph

Applications of Graph Theory

Graph theory has its applications in diverse fields of engineering −

1) Electrical Engineering − The concepts of graph theory is used extensively in designing circuit connections. 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.

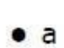
6) General − Routes between the cities can be represented using graphs. Depicting hierarchical ordered information such as family tree can be used as a special type of graph called tree.

Now I know… **A graph is a diagram of points and lines connected to the points.** It has at least one line joining a set of two vertices with no vertex connecting itself. The concept of graphs in graph theory stands up on some basic terms such as ***point, line, vertex, edge, degree of vertices, properties of graphs, etc.***

Common Graph Terms

1) Point. It is particular position in a ***one-dimensional, two-dimensional, or three-dimensional space.*** It can be denoted by an alphabet. It can be represented with a dot.

Example



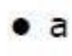
2) **Line** is a connection between two points. It can be represented with a solid line.

Example

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.

Example



Here, the vertex is named with an alphabet ‘a’.

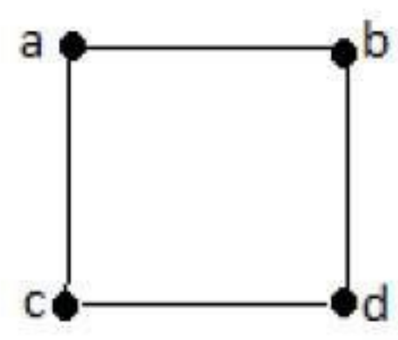
4) **Edge.** An ***edge*** is the mathematical term for a line that connects two vertices. Many edges can be formed from a single vertex. Without a vertex, an edge cannot be formed. There must be a starting vertex and an ending vertex for an edge.

Example

Here, ‘a’ and ‘b’ are the two vertices and the link between them is called an edge.

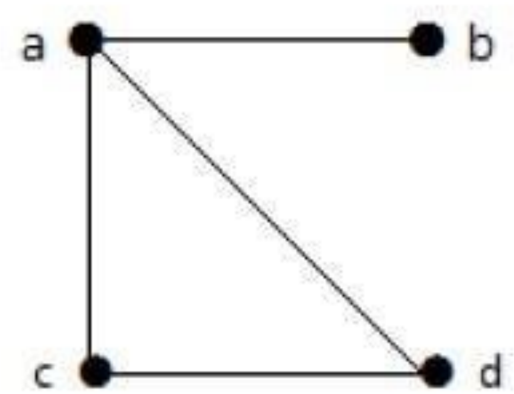
5) Graph. A graph ‘G’ is defined as G = (V, E) Where V is a set of all vertices and E is a set of all edges in the graph.

Example 1



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

Example 2



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

6) Loop. In a graph, if an edge is drawn from vertex to itself, it is ***called a loop.***

Example 1

In the given example, ***V*** is a vertex for which it has an ***edge (V, V)*** forming a loop.

Example 2

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

Degree of Vertex

It is the number of vertices adjacent to a vertex V.

Notation − deg(V).

The symbol ∀ means "**for all" or "for any"**.

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

$\deg(v) \leq n - 1 \; \forall \, v \in G$

Key Idea. A vertex can form an edge with all other vertices except 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.***

Key Idea. The number of edges of a given tree is equal to number of vertices minus 1 (n of edges =n of vertices – 1)

Question 1: What is the number of edges in a tree with 11 vertices?

Question 2: What is the number of edges in a tree with 15 vertices?

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

- Undirected Graph
- Directed Graph

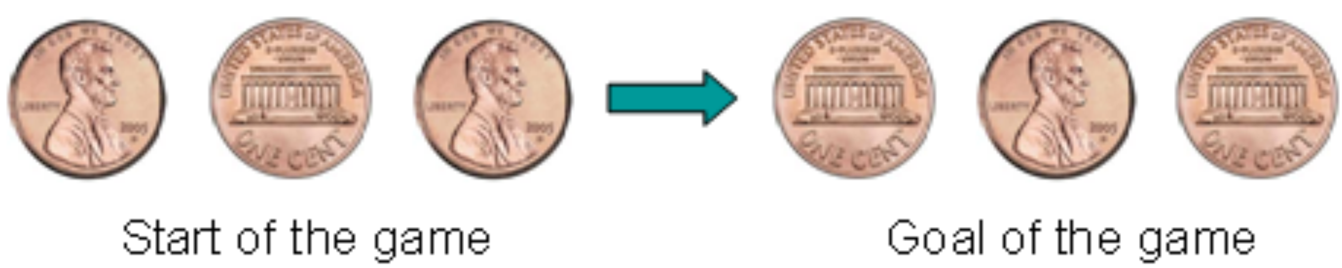
Degree of Vertex in an Undirected Graph

Undirected graph definition:

- An undirected graph is a set of nodes and a set of links between the nodes.
- Each node is called a **vertex**, each link is called an **edge**, and each edge connects two vertices.
- The order of the two connected vertices is unimportant.
- An undirected graph is a finite set of vertices together with a finite set of edges. Both sets might be empty, which is called the empty graph.

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

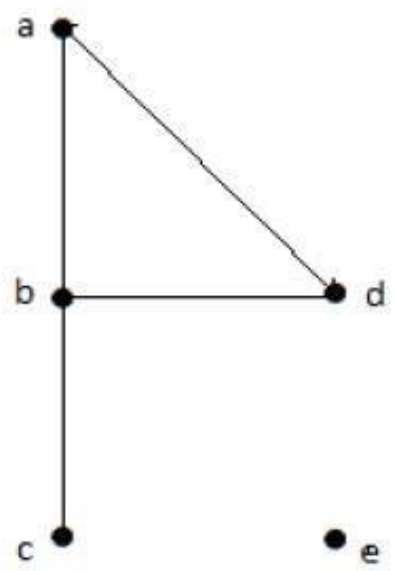
A Motivating Example: Coin Game (Undirected State Graphs)



Rules:

1. You may flip the middle coin whenever you want to.
2. You may flip one of the end coin only if the other two coins are the same as each other.

Example 1

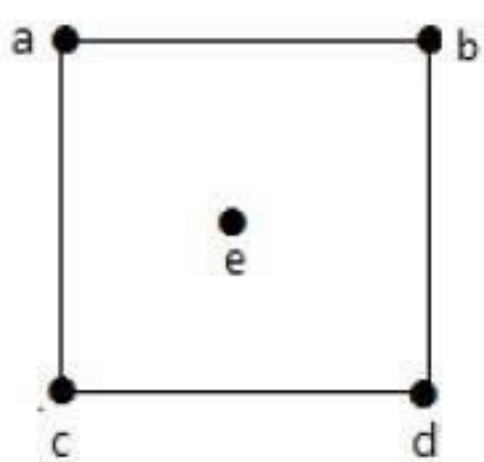


In the given undirected graph,

- ***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 formed at vertex ‘c’, so it 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.

Example 2

Take a look at the following graph −



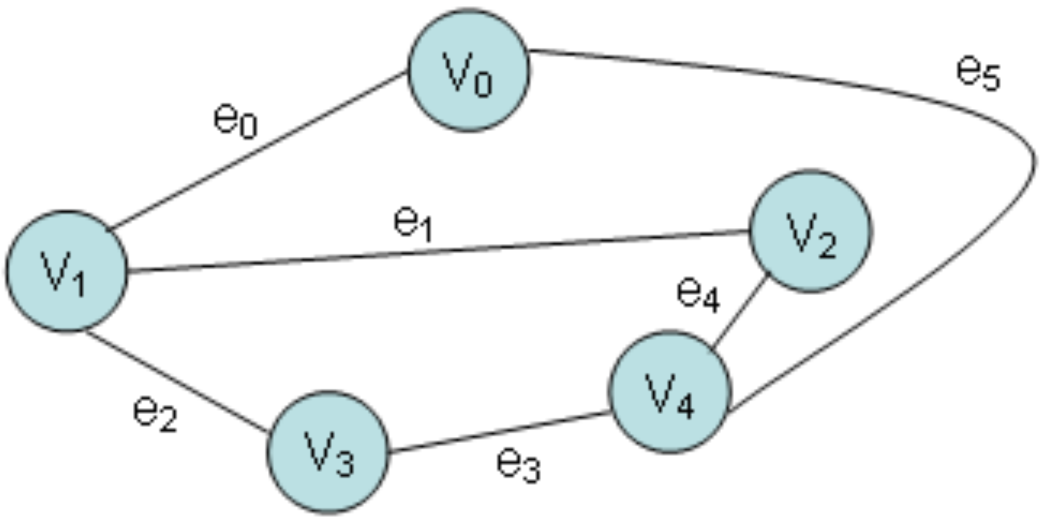
In the given graph,

deg(a) = 2, deg(b) = 2, deg(c) = 2, deg(d) = 2, and deg(e) = 0.

The vertex ‘e’ is an ***isolated vertex***. The graph does not have any pendent vertex.

CHECK FOR UNDERSTANDING 1 – (UNDIRECTED GRAPH)

Identify the degree of Vertex in the given undirected Graph



Deg (V_0) =

Deg (V_1) =

Deg (V_2) =

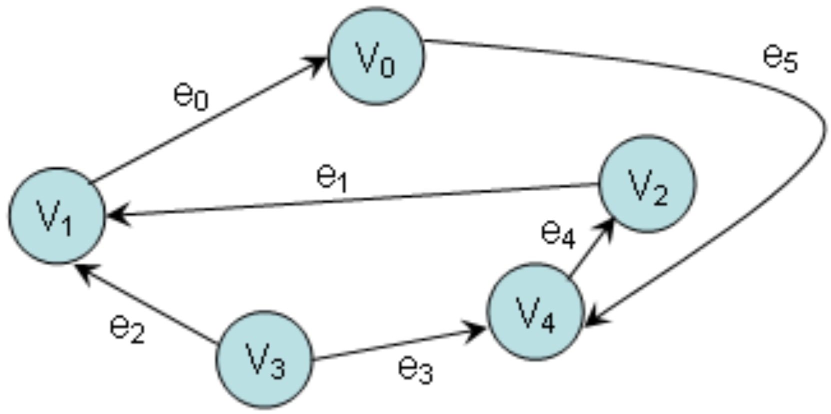
Deg (V_3) =

Deg (V_4) =

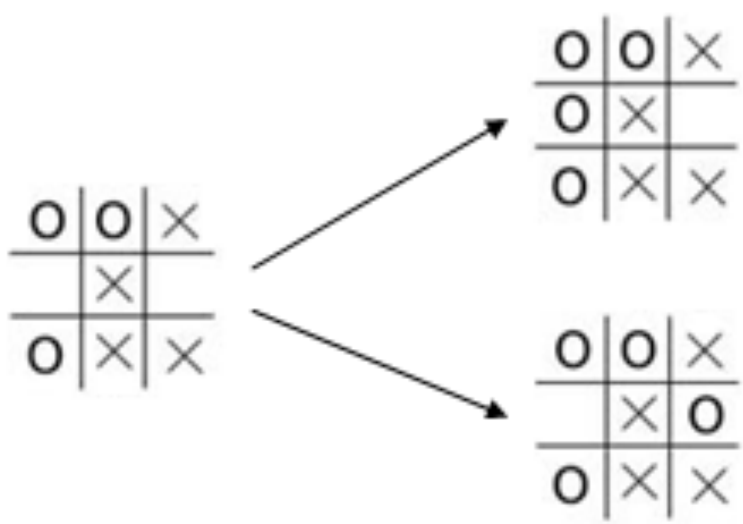
Degree of Vertex in a Directed Graph

Directed graph definition

- A directed graph is a finite set of vertices together with a finite set of edges. Both sets might be empty, which is called the ***empty graph.***
- Each edge is associated with two vertices, called its **source** and **target** vertices.
- We say that the edge connects its source to its target.
- The **order** of the two connected vertices is important



One application of directed graphs is a state graph for a game where reversing a move is sometimes forbidden. For example, tic-tac-toe.



In a directed graph, each vertex has an ***indegree*** and an ***outdegree.***

Indegree of a Graph

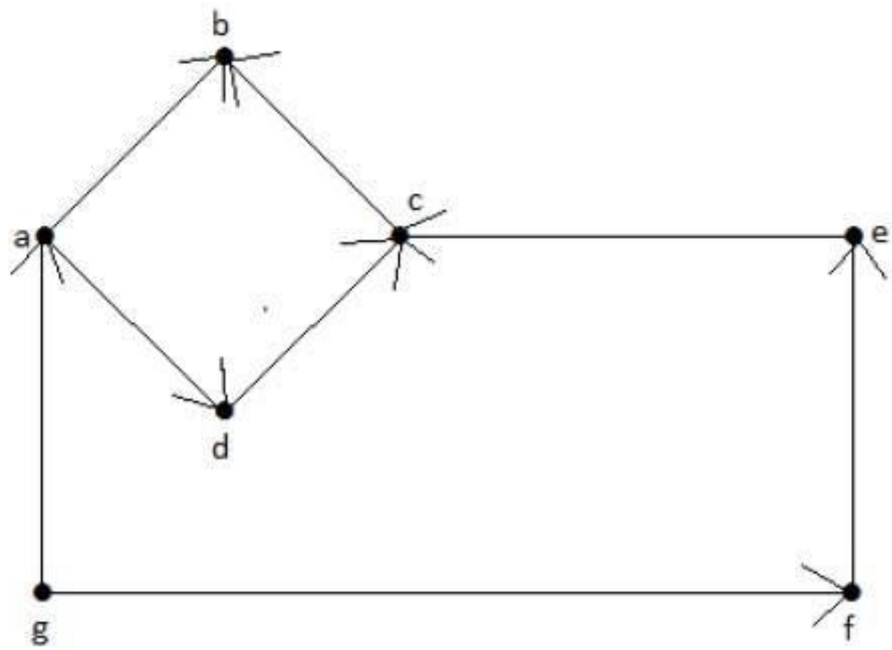
- Indegree of vertex V is the number of edges which are coming into the vertex V.
- **Notation** − deg−(V).

Outdegree of a Graph

- Outdegree of vertex V is the number of edges which are going out from the vertex V.
- **Notation** − deg+(V).

Example 1

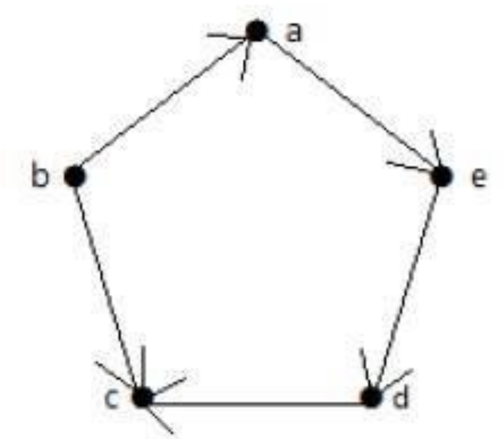
Take a look at the following directed graph. Vertex 'a' has two edges, 'ad' and 'ab', which are going outwards. Hence its outdegree is 2. Similarly, there is an edge 'ga', coming towards vertex 'a'. Hence the indegree of 'a' is 1.



The indegree and outdegree of the given vertices are shown in the following table.

Vertex	Indegree	Outdegree
a	1	2
b	2	0
c	2	1
d	1	1
e	1	1
f	1	1
g	0	2

CHECK FOR UNDERSTANDING 2 – (DIRECTED GRAPH)



List down the indegree and outdegree of the given vertices

Vertex	Indegree	Outdegree
a		
b		
c		
d		
e		

Additional Graph Terminologies:

Loop: an edge that connects a vertex to itself.

Path: a sequence of vertices, $p_0, p_1, ..., p_m$, such that each adjacent pair of vertices p_i and p_{i+1} are connected by an edge.

Cycle: a simple path with no repeated vertices or edges other than the starting and ending vertices. A cycle in a directed graph is called a directed cycle.

Multiple edges: in principle, a graph can have two or more edges connecting the same two vertices in the same direction.

Simple graphs: the graphs that have no loops and no multiple edges. In fact, many applications require only simple directed graphs or even simple undirected graphs.

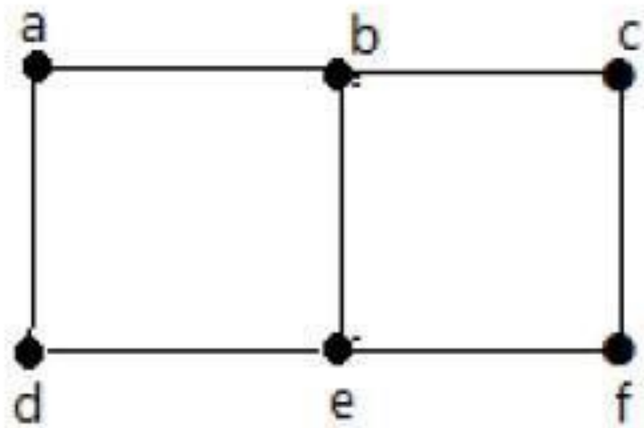
Adjacency

Here are the norms of adjacency –

- In a graph, ***two vertices*** are said to be **adjacent** if there is an edge between the two vertices. Here, the adjacency of vertices is maintained by the single edge that is connecting those two vertices.

- In a graph, ***two edges*** are said to be ***adjacent*** if there is a common vertex between the two edges. Here, the adjacency of edges is maintained by the single vertex that is connecting two edges.

Example 1

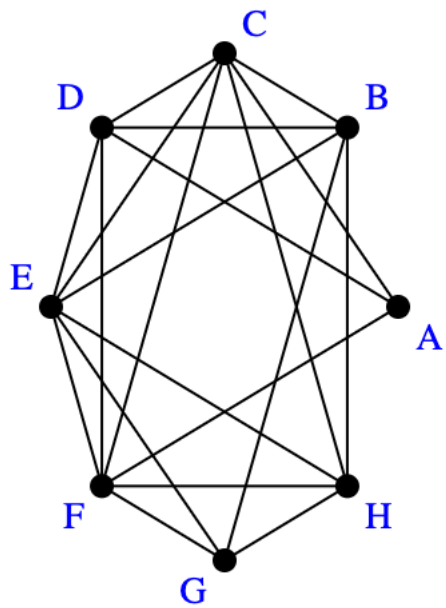


In the above graph −

- ‘a’ and ‘b’ are the ***adjacent vertices***, as there is a ***common edge*** ‘ab’ between them.
- ‘a’ and ‘d’ are the ***adjacent vertices,*** as there is a ***common edge*** ‘ad’ between them.
- ab’ and ‘be’ are the ***adjacent edges***, as there is a ***common vertex*** ‘b’ between them.
- be’ and ‘de’ are the ***adjacent edges,*** as there is a ***common vertex*** **‘e’ between them.**

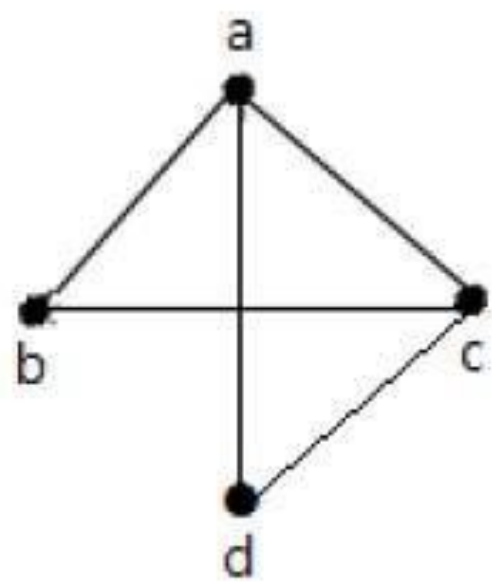
CHECK FOR UNDERSTANDING 3 – (ADJACENCY)

Which vertices are **adjacent** to vertex C?

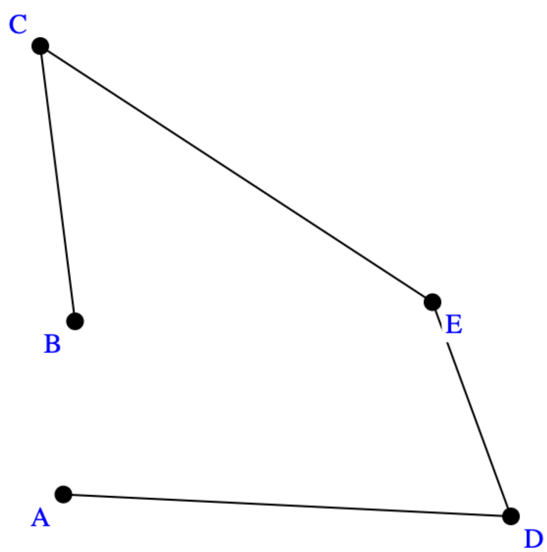


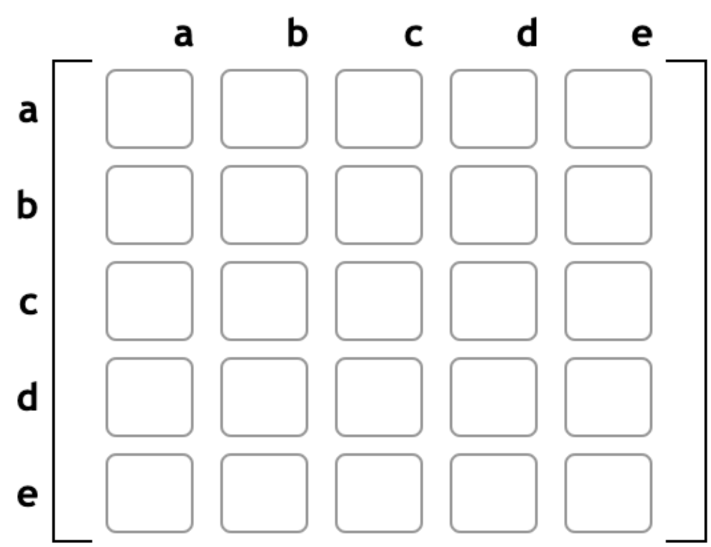
CHECK FOR UNDERSTANDING 4 – (ADJACENCY)

a. List down at least two adjacent vertices and adjacent edges of the given graph below.



b. What is the adjacency matrix for this graph (with the given the labeling on the vertices)?

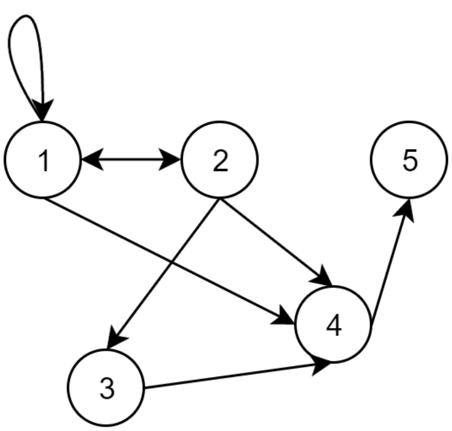




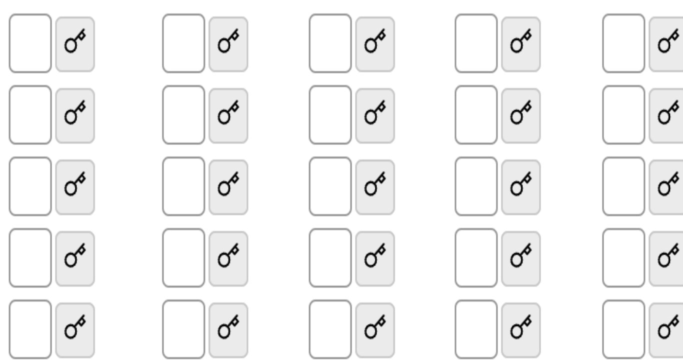
Key: Adjacency Matrix (b)

$$
\begin{array}{c|ccccc|}
 & a & b & c & d & e \\
\hline
a & 0 & 0 & 0 & 1 & 0 \\
b & 0 & 0 & 1 & 0 & 0 \\
c & 0 & 1 & 0 & 0 & 1 \\
d & 1 & 0 & 0 & 0 & 1 \\
e & 0 & 0 & 1 & 1 & 0
\end{array}
$$

c. What is the adjacency matrix for this graph (with the given the labeling on the vertices)?



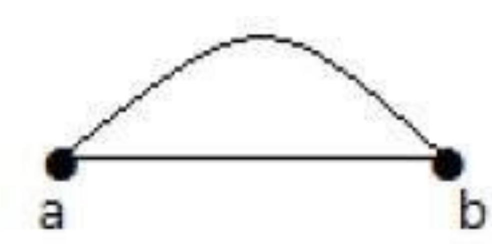
What is the adjacency matrix for this graph (given the natural ordering on vertices)?



1	1	0	1	0
1	0	1	1	0
0	0	0	1	0
0	0	0	0	1
0	0	0	0	0

Parallel Edges

In a graph, if a pair of vertices is connected by more than one edge, then those edges are called ***parallel edges.***

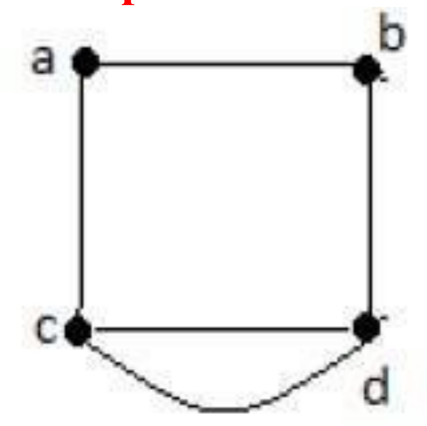


In the above graph, 'a' and 'b' are the two vertices which are connected by two edges 'ab' and 'ab' between them. So it is called as a ***parallel edge.***

Multi Graph

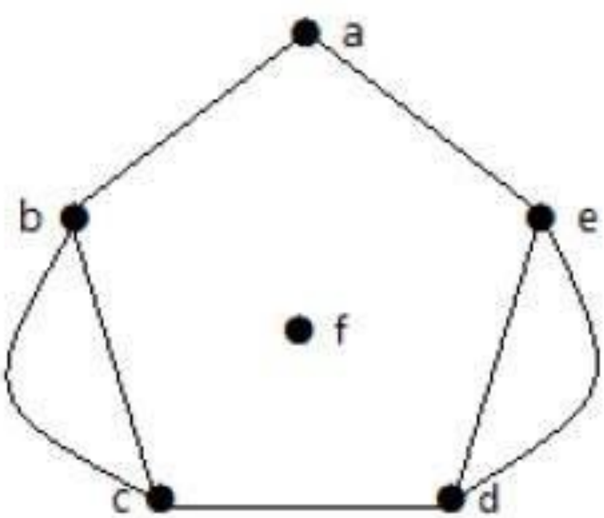
A graph having parallel edges is known as a ***Multigraph.***

Example 1



In the given graph, there are five edges 'ab', 'ac', 'cd', 'cd', and 'bd'. Since 'c' and 'd' have two parallel edges between them, it a ***Multigraph.***

Example 2

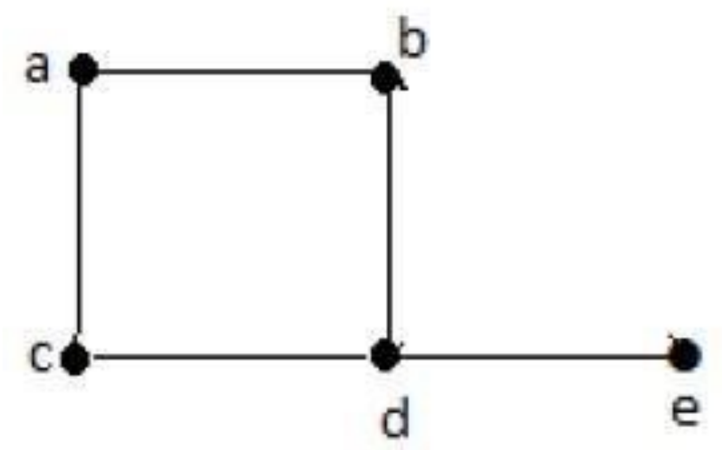


In the above graph, the vertices 'b' and 'c' have two edges. The vertices 'e' and 'd' also have two edges between them. Hence it is a Multigraph.

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



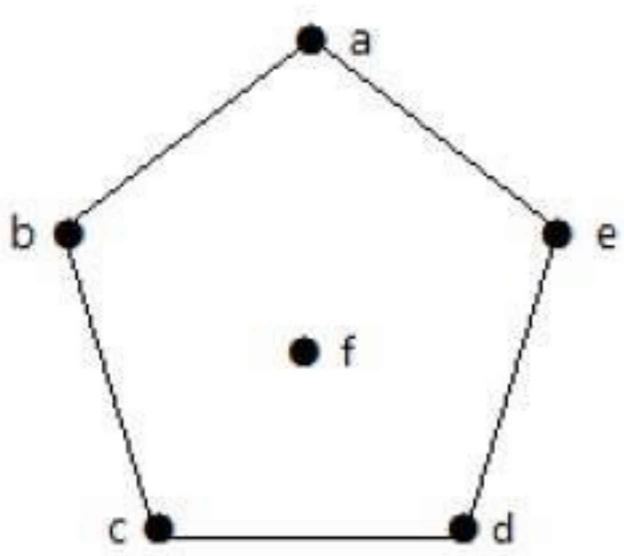
Vertex	a	b	c	d	e
Connecting to	b, c	a, d	a, d	c, b, e	d
Degree	2	2	2	3	1

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

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

CHECK FOR UNDERSTANDING 5 (DEGREE SEQUENCE)

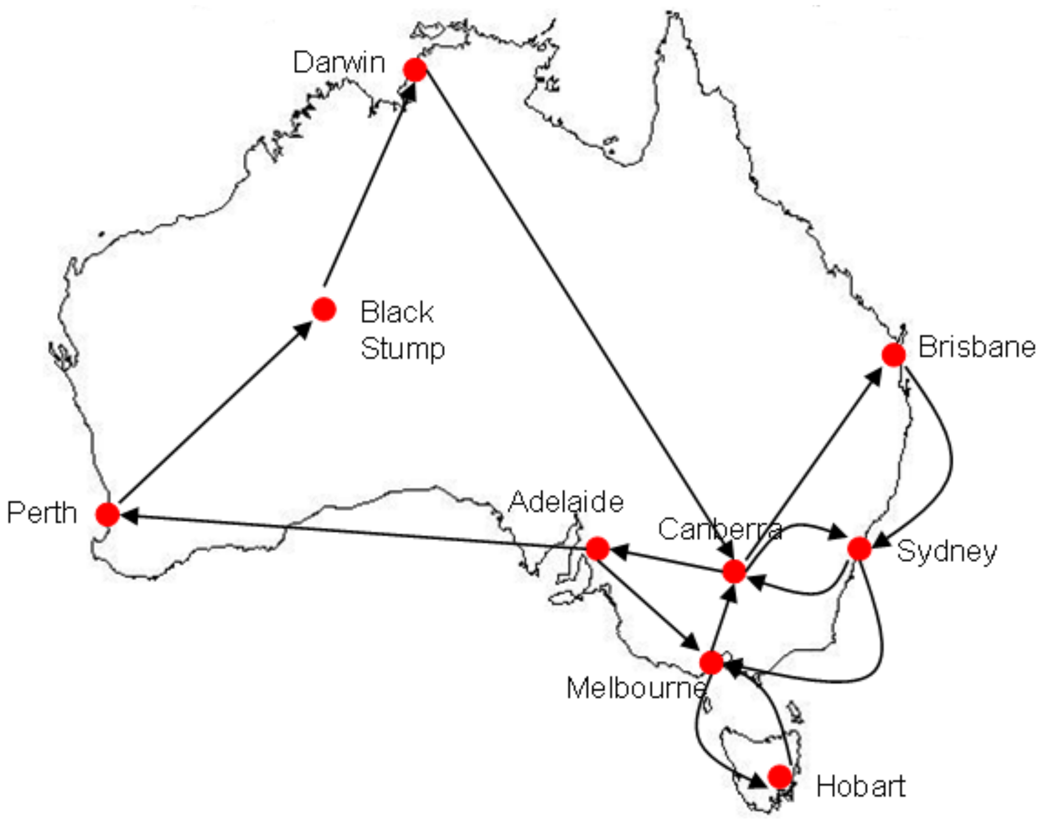
Fill out the table by identifying the vertices connected to a given vertex and identify the degree. Also, arrange the degree sequence in both ascending and descending order.



Vertex	a	b	c	d	e	f
Connecting to						
Degree						

Activity #2 – All About Graph Theories

Airline Routing Example: Crocodile Airlines Routes



Questions:

1. **How many vertices and edges does the graph have? How many loops?**

2. **Is it a simple graph? Why or why not?**

3. **What is the shortest path from "Black Stump" to "Melbourne"? -- The shortest path problem.**

Pendent Vertex

By using degree of a vertex, we have a two special types of vertices. A vertex with degree one is called a ***pendent vertex.***

Example

Here, in this example, vertex 'a' and vertex 'b' have a connected edge 'ab'. So with respect to the vertex 'a', there is only one edge towards vertex 'b' and similarly with respect to the vertex 'b', there is only one edge towards vertex 'a'. Finally, vertex 'a' and vertex 'b' has degree as one which are also called as the pendent vertex.

Isolated Vertex

A vertex with degree zero is called an isolated vertex.

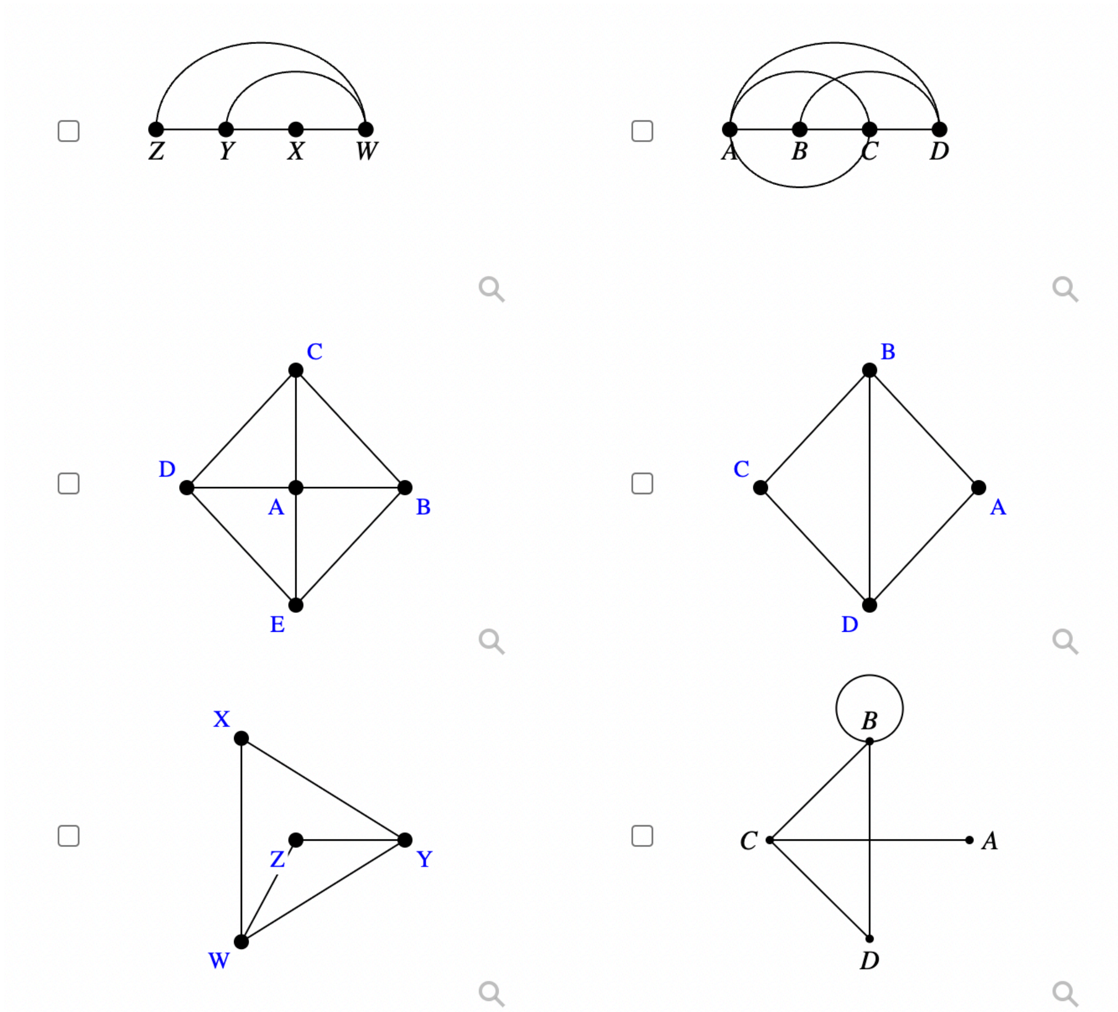
Example

Here, the vertex 'a' and vertex 'b' has a no connectivity between each other and also to any other vertices. So, the degree of both the vertices 'a' and 'b' are zero. These are also called as ***isolated vertices.***

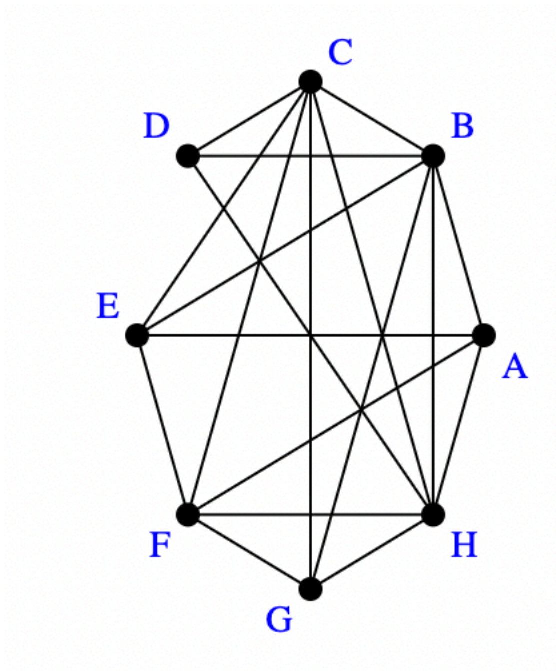
Equivalent Graphs

Graphs are **equivalent** if they have the same number of vertices and the same edge connections. The vertices do *not* need to have the same labels, and they do not have to be drawn in the same positions.

Select all the graphs that are equivalent.



EVEN & ODD VERTICES



Which vertices are **even**?

- ☐ A
- ☐ B
- ☐ C
- ☐ D
- ☐ E
- ☐ F
- ☐ G
- ☐ H
- ☐ None are even

Which vertices are **odd**?

- ☐ A
- ☐ B
- ☐ C
- ☐ D
- ☐ E
- ☐ F
- ☐ G
- ☐ H
- ☐ None are odd

Sources

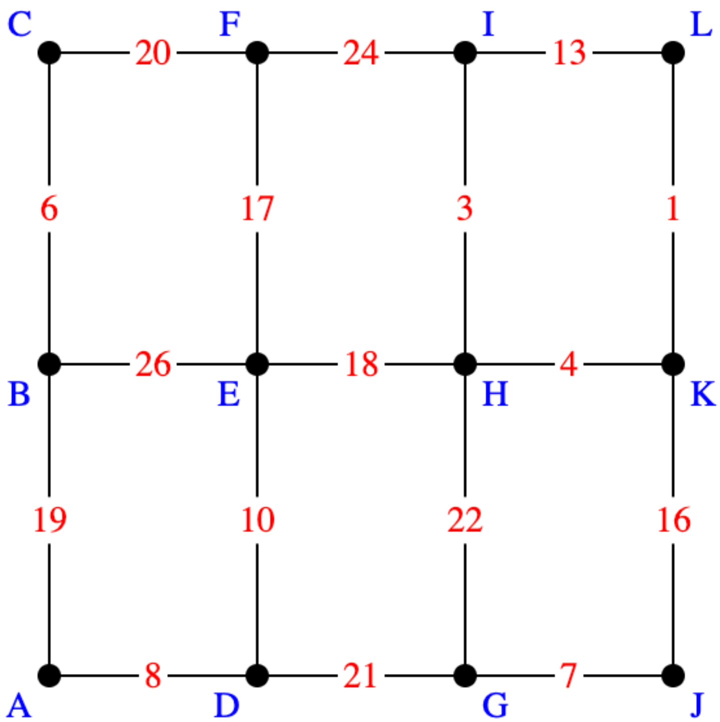
https://www.cpp.edu/~ftang/courses/CS241/notes/graph.htm

https://www.tutorialspoint.com/graph_theory/graph_theory_fundamentals.htm

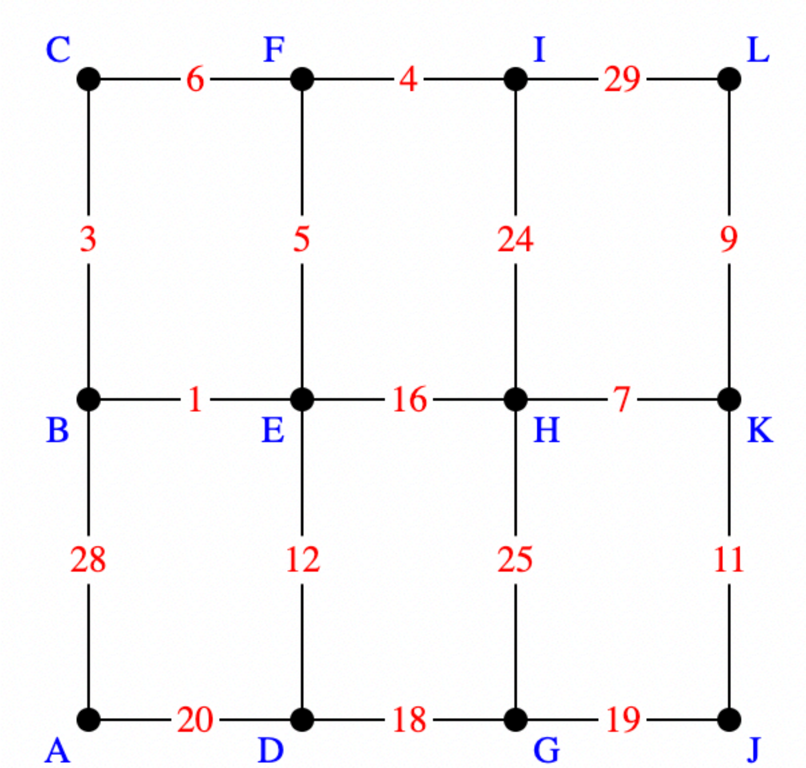
Sources:
https://www.javatpoint.com/sets-operations

QUICK REVIEW

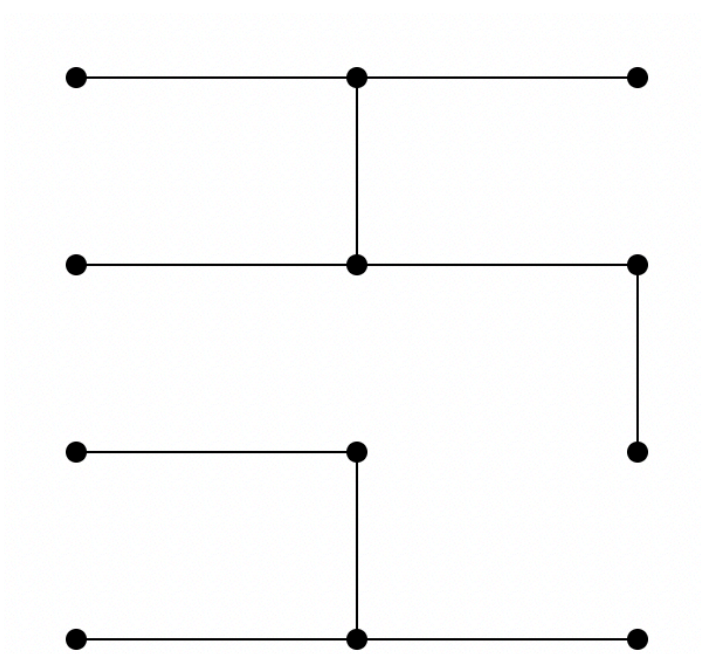
QUESTION #1. Find the length of the shortest path from vertex A to vertex L.



QUESTION #2. Find the length of the shortest path from vertex A to vertex L.

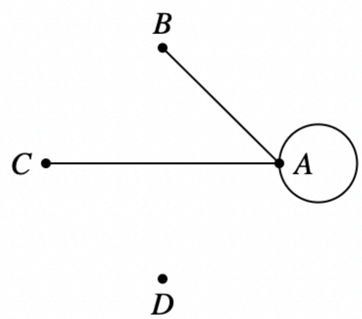


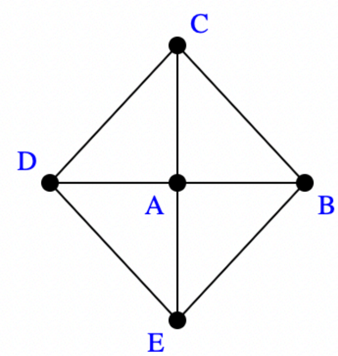
3) Is this graph connected?

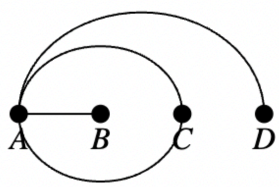


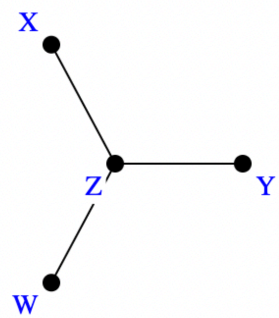
A. Connected
B. Not Connected

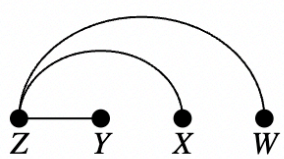
Question #4. Graphs are **equivalent** if they have the same number of vertices and the same edge connections. The vertices do *not* need to have the same labels, and they do not have to be drawn in the same positions. Select all the graphs that are equivalent.

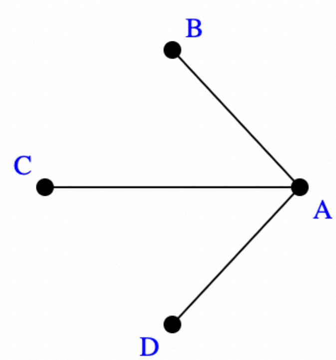






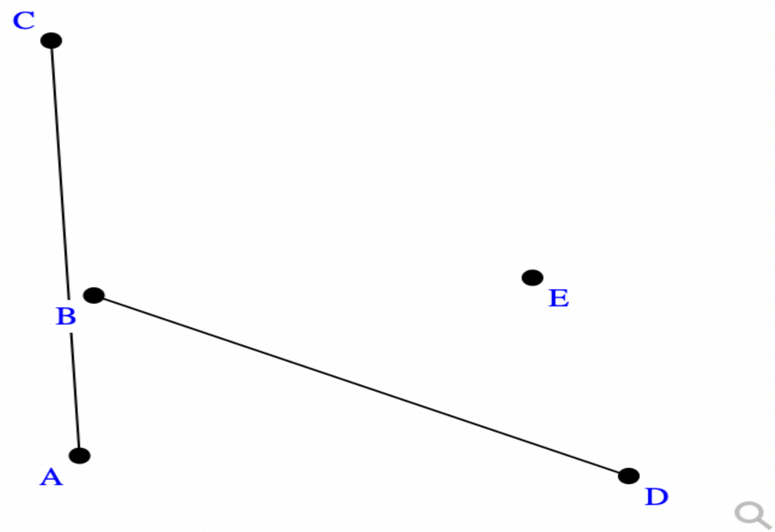






Question #5.

What is the adjacency matrix for this graph (with the given the labeling on the vertices)?



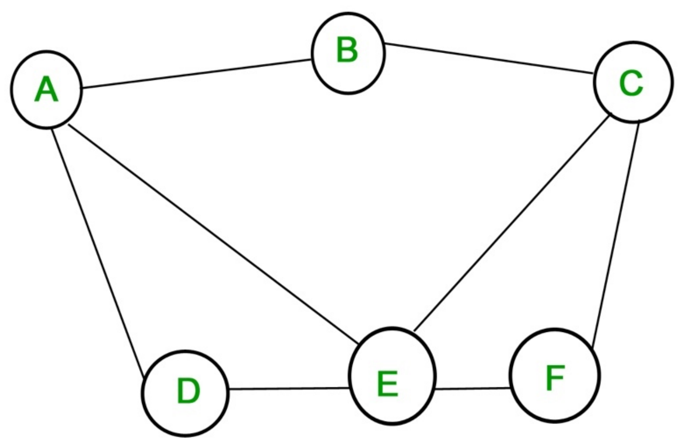
$$\begin{array}{c c}
 & \begin{array}{ccccc} \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} & \mathbf{e} \end{array} \\
\begin{array}{c} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \\ \mathbf{e} \end{array} & \begin{bmatrix} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \end{bmatrix}
\end{array}$$

Quick Review

Graphs come with various properties which are used for characterization of graphs depending on their structures. These properties are defined in specific terms pertaining to the domain of graph theory. In this chapter, we will discuss a few basic properties that are common in all graphs.

A ***graph*** is defined as set of points known as 'Vertices' and line joining these points is known as 'Edges'. It is a set consisting of where 'V' is vertices and 'E' is edge.

Vertices: {A, B, C, D, E, F}
Edges: {{A, B}, {A, D}, {A, E}, {B, C}, {C, E}, {C, F}, {D, E}, {E, F}}



Graph measurements: length, distance, diameter, eccentricity, radius, center

1) Distance between Two Vertices

It is number of edges in a shortest path between Vertex U and Vertex V. If there are multiple paths connecting two vertices, then the shortest path is considered as the distance between the two vertices.

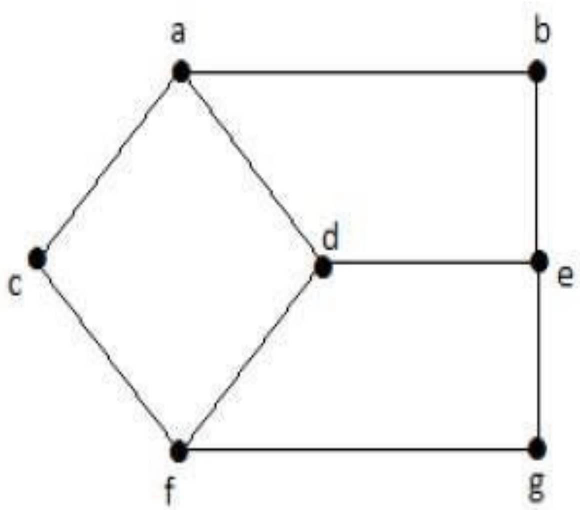
Notation − d (U, V)

There can be any number of paths present from one vertex to other. Among those, you need to choose only the shortest one.

Key Idea. The distance between two vertices in a graph is the number of edges in a shortest or minimal path. It gives the available minimum distance between two edges. There can exist more than one shortest path between two vertices.

Example 1)

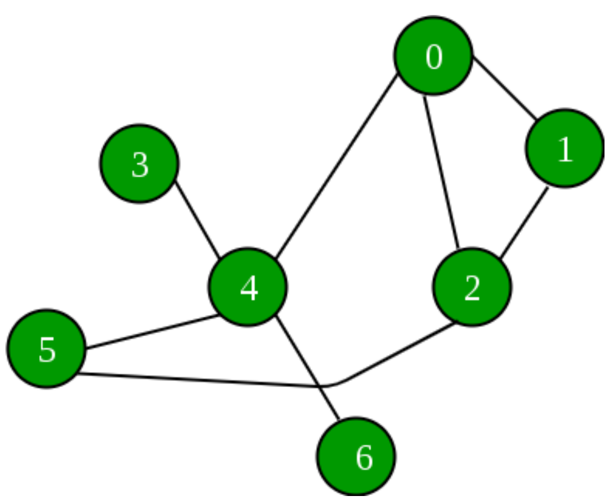
Take a look at the following graph −



Here, the distance from vertex 'd' to vertex 'e' or simply 'de' is 1 as there is one edge between them. There are many paths from vertex 'd' to vertex 'e' −

- da, ab, be
- df, fg, ge
- de (It is considered for distance between the vertices)
- df, fc, ca, ab, be
- da, ac, cf, fg, ge

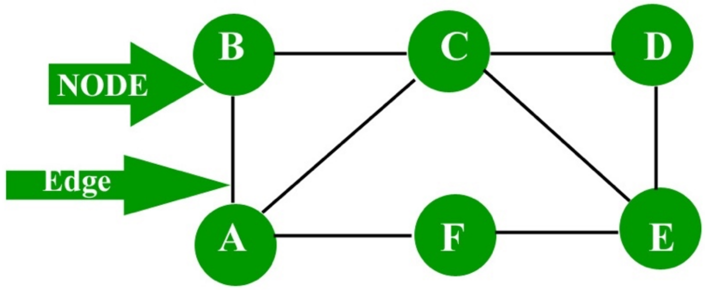
Example 2)



Shortest Distance between 1 - 5 is **2**

1 → 2 → 5

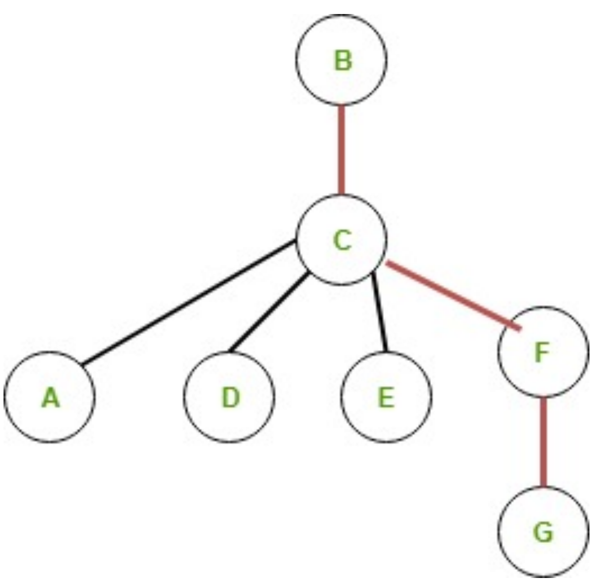
2) Length – Length of the graph is defined as the number of edges contained in the graph.



Length of the graph: 8

AB, BC, CD, DE, EF, FA, AC, CE

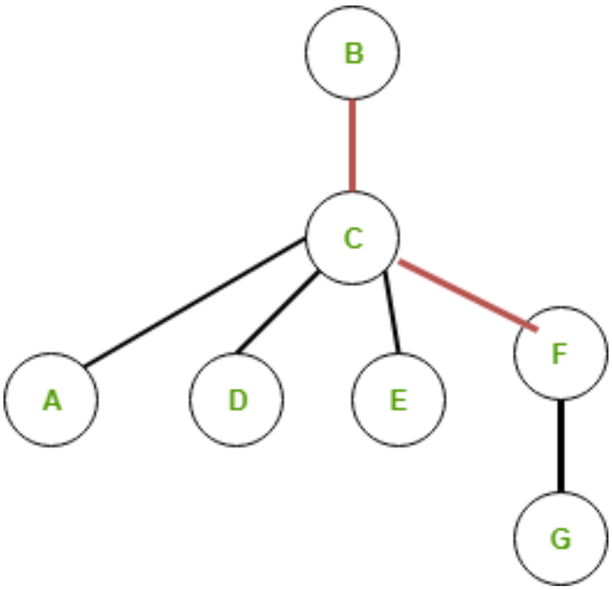
3) Diameter of graph – The diameter of graph is the maximum distance between the pair of vertices. It can also be defined as the maximal distance between the pair of vertices. Way to solve it is to find all the paths and then find the maximum of all.



Diameter: **3**

BC → CF → FG

4) Radius of graph –A radius of the graph exists only if it has the diameter. The minimum among all the maximum distances between a vertex to all other vertices is considered as the radius of the Graph G. It is denoted as r(G).



Radius: 2

All available minimum radius:

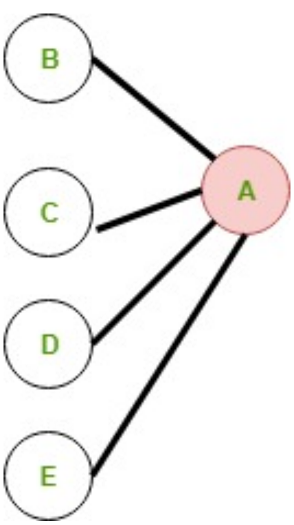
BC → CF,

BC → CE,

BC → CD,

BC → CA

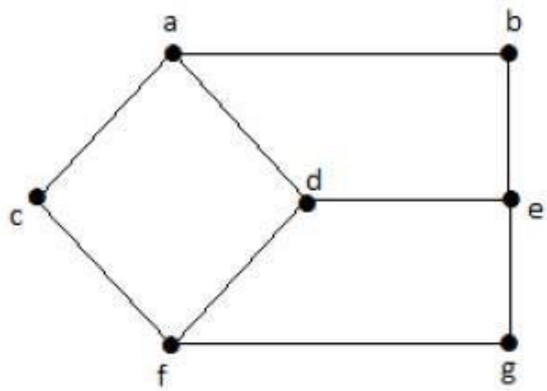
5) Centre of graph – It consists of all the vertices whose eccentricity is minimum. Here the eccentricity is equal to the radius. For example, if the school is at the centre of town, it will reduce the distance buses has to travel.



`Centre: A`

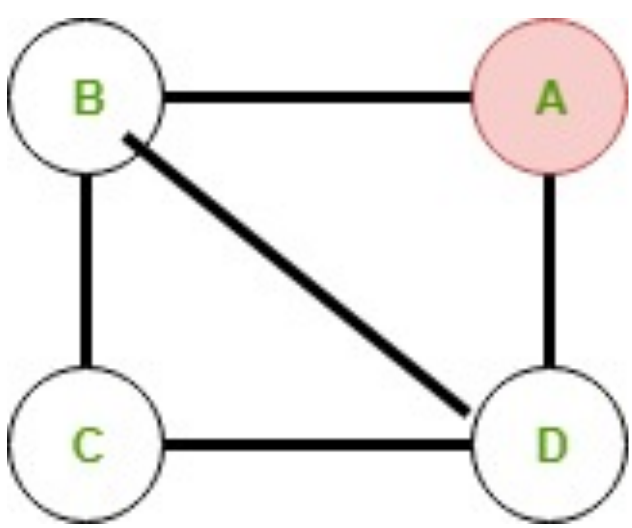
Key idea. The set of all central points of 'G' is called the centre of the Graph.

Example



In the example graph, {'d'} is the centre of the Graph.

6) **Eccentricity of graph –** It is defined as the maximum distance of one vertex from other vertex. The maximum distance between a vertex to all other vertices is considered as the eccentricity of the vertex. It is denoted by e(V).



Eccentricity from:

(A, A) = 0

(A, B) = 1

(A, C) = 2

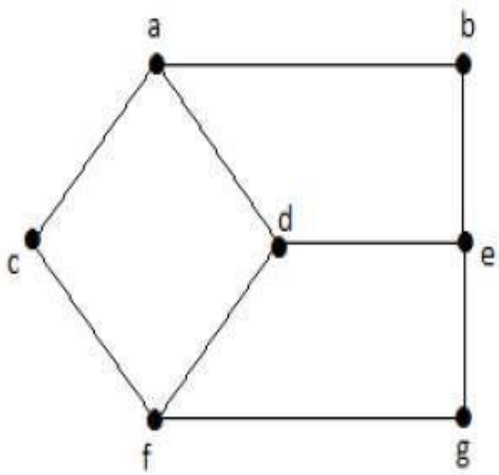
(A, D) = 1

Maximum value is 2, So Eccentricity is **2**

Key Idea. The maximum distance between a vertex to all other vertices is considered as the eccentricity of vertex. The distance from a particular vertex to all other vertices in the graph is taken and among those distances, the eccentricity is the highest of distances.

Example

In the given graph, the eccentricity of ‘a’ is 3.



The distance from ‘a’ to ‘b’ is 1 (‘ab’),

from ‘a’ to ‘c’ is 1 (‘ac’),

from ‘a’ to ‘d’ is 1 (‘ad’),

from ‘a’ to ‘e’ is 2 (‘ab’-‘be’) or (‘ad’-‘de’),

from ‘a’ to ‘f’ is 2 (‘ac’-‘cf’) or (‘ad’-‘df’),

from ‘a’ to ‘g’ is 3 (‘ac’-‘cf’-‘fg’) or (‘ad’-‘df’-‘fg’).

So, the eccentricity is 3, which is a maximum from vertex ‘a’ from the distance between ‘ag’ which is maximum.

In other words,

e(b) = 3

e(c) = 3

e(d) = 2

e(e) = 3

e(f) = 3

e(g) = 3

CHECK FOR UNDERSTANDING 6 (3 minutes)

1) It is number of edges in a shortest path between Vertex U and Vertex V.

A) Distance
B) Length
C) Radius
D) Diameter

2) It is defined as the maximum distance of one vertex from other vertex.

A) Diameter of a graph
B) Radius of a graph
C) Center of a graph
D) Eccentricity of graph

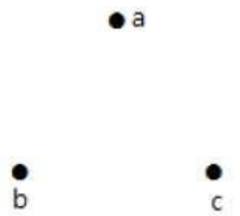
Types of Graphs

There are various types of graphs depending upon the number of vertices, number of edges, interconnectivity, and their overall structure.

1) Null Graph. A graph having no edges is called a Null Graph.

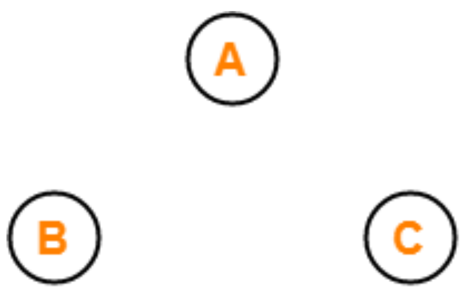
Key Idea. A graph whose edge set is empty is called as a null graph. In other words, a null graph does not contain any edges in it.

Example 1



In the above graph, there are three vertices named ‘a’, ‘b’, and ‘c’, but there are no edges among them. Hence it is a Null Graph.

Example 2



Example of Null Graph

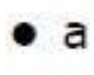
Here,

- This graph consists only of the vertices and there are no edges in it.
- Since the edge set is empty, therefore it is a null graph.

2) Trivial Graph. A **graph with only one vertex** is called a Trivial Graph.

Key Idea. A graph having only one vertex in it is called as a trivial graph. It is the smallest possible graph.

Example 1



In the above shown graph, there is only one vertex ‘a’ with no other edges. Hence it is a Trivial graph.

Example 2

Example of Trivial Graph

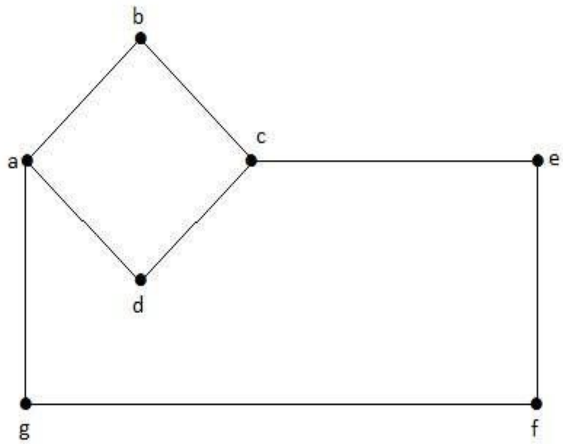
Here,

- This graph consists of only one vertex and there are no edges in it.
- Since only one vertex is present, therefore it is a trivial graph.

3) Non-Directed Graph. A non-directed graph contains edges, but the edges are not directed ones.

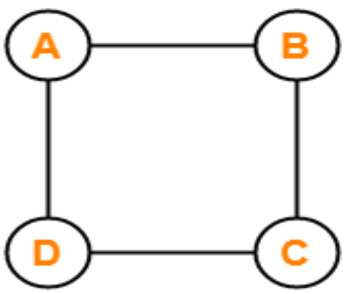
Key Idea. A graph in which all the edges are undirected is called as a non-directed graph. In other words, edges of an undirected graph do not contain any direction.

Example 1



In this graph, ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’ are the vertices, and ‘ab’, ‘bc’, ‘cd’, ‘da’, ‘ag’, ‘gf’, ‘ef’ are the edges of the graph. Since it is a non-directed graph, the edges ‘ab’ and ‘ba’ are same. Similarly other edges also considered in the same way.

Example 2



Example of Non-Directed Graph

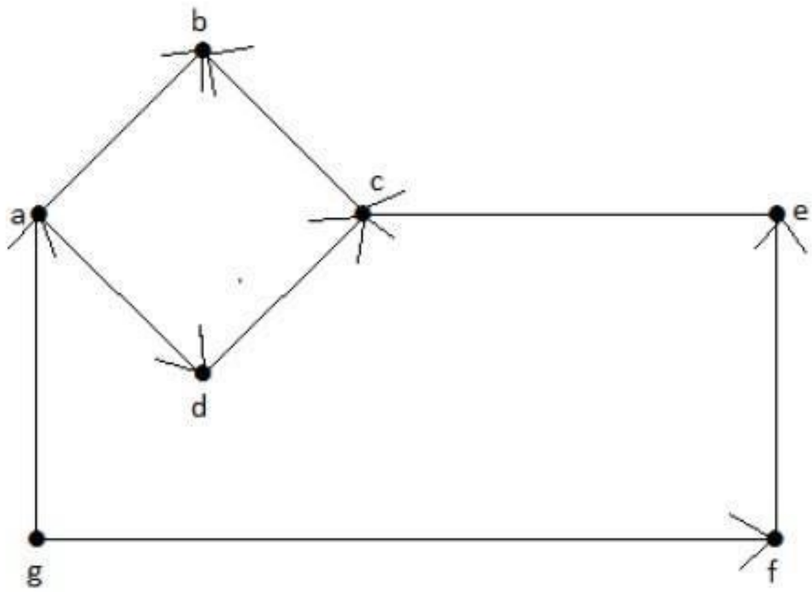
Here,

- This graph consists of four vertices and four undirected edges.
- Since all the edges are undirected, therefore it is a non-directed graph.

4) Directed Graph. In a directed graph, each edge has a direction.

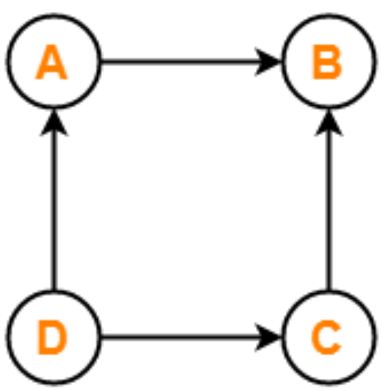
Key idea. A graph in which all the edges are directed is called as a directed graph. In other words, all the edges of a directed graph contain some direction. Directed graphs are also called as **digraphs**.

Example 1



In the given graph, we have seven vertices 'a', 'b', 'c', 'd', 'e', 'f', and 'g', and eight edges 'ab', 'cb', 'dc', 'ad', 'ec', 'fe', 'gf', and 'ga'. As it is a directed graph, each edge bears an arrow mark that shows its direction. Note that in a directed graph, 'ab' is different from 'ba'.

Example 2



Example of Directed Graph

Here,

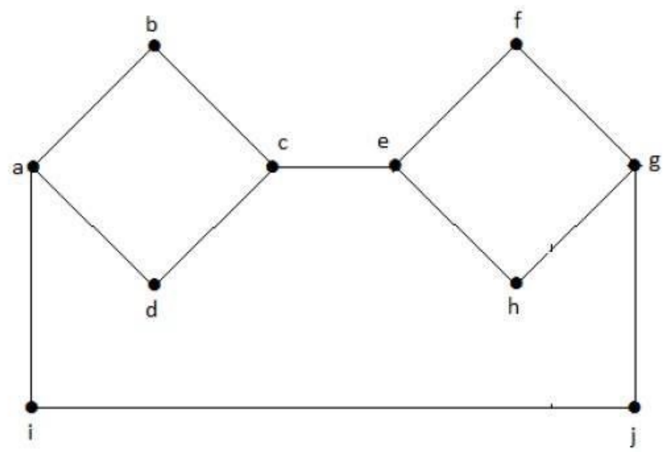
- This graph consists of four vertices and four directed edges.
- Since all the edges are directed, therefore it is a directed graph.

5) Connected Graph. A graph *G* is said to be connected **if there exists a path between every pair of vertices**. There should be at least one edge for every vertex in the graph. So that we can say that it is connected to some other vertex at the other side of the edge.

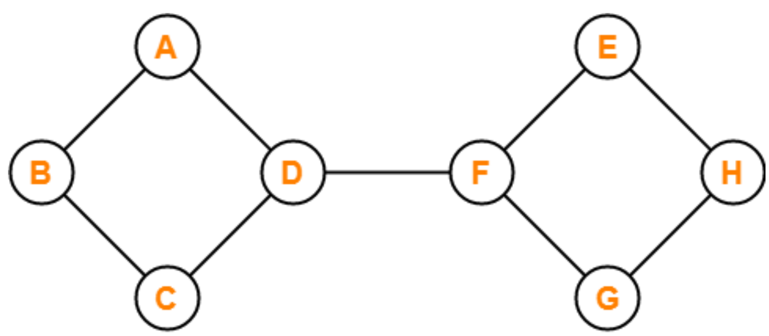
Key Idea. A graph in which we can visit from any one vertex to any other vertex is called as a connected graph. In connected graph, at least one path exists between every pair of vertices.

Example 1

In the following graph, each vertex has its own edge connected to other edge. Hence it is a connected graph.



Example 2



Example of Connected Graph

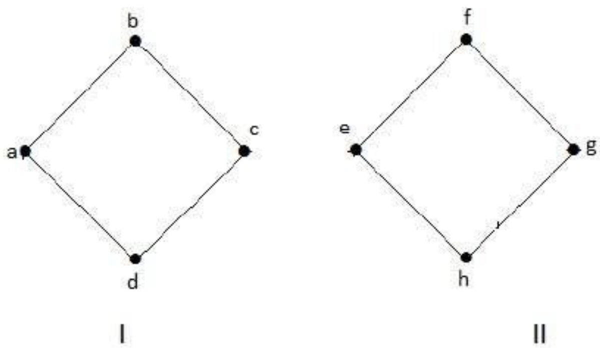
Here,

- In this graph, we can visit from any one vertex to any other vertex.
- There exists at least one path between every pair of vertices.
- Therefore, it is a connected graph.

6) **Disconnected Graph.** A graph *G* is disconnected, if it does not contain at least two connected vertices.

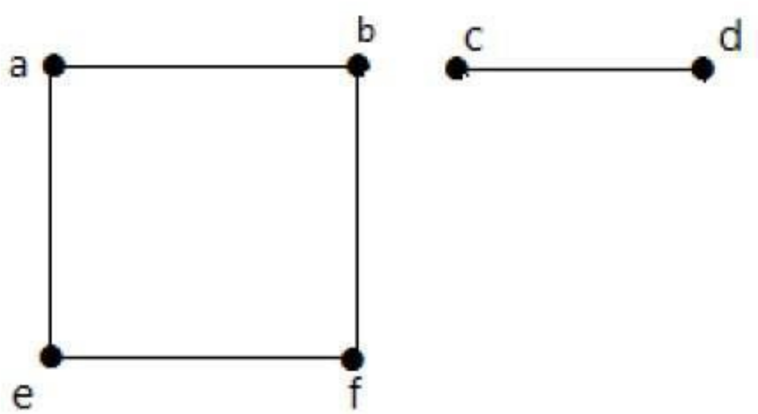
Example 1

The following graph is an example of a Disconnected Graph, where there are two components, one with ‘a’, ‘b’, ‘c’, ‘d’ vertices and another with ‘e’, ’f’, ‘g’, ‘h’ vertices.



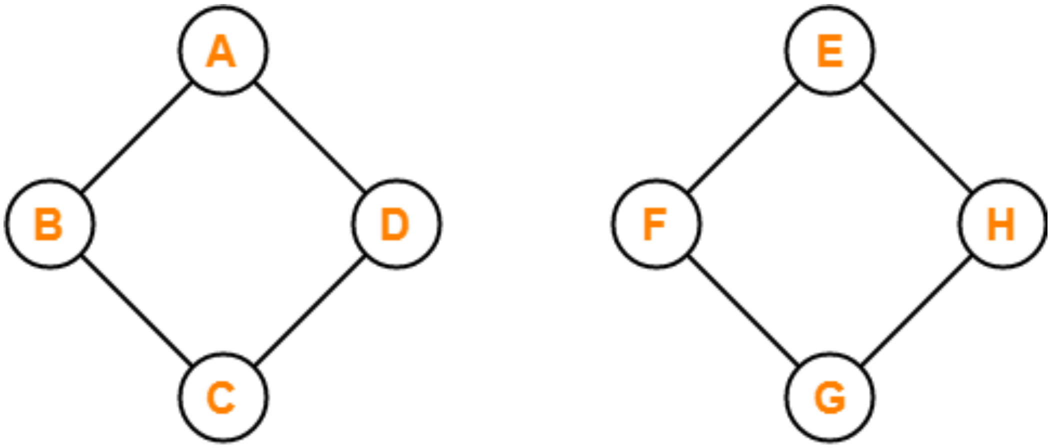
The two components are independent and not connected to each other. Hence it is called disconnected graph.

Example 2



In this example, there are two independent components, a-b-f-e and c-d, which are not connected to each other. Hence this is a disconnected graph.

Example 3



Example of Disconnected Graph

Here,

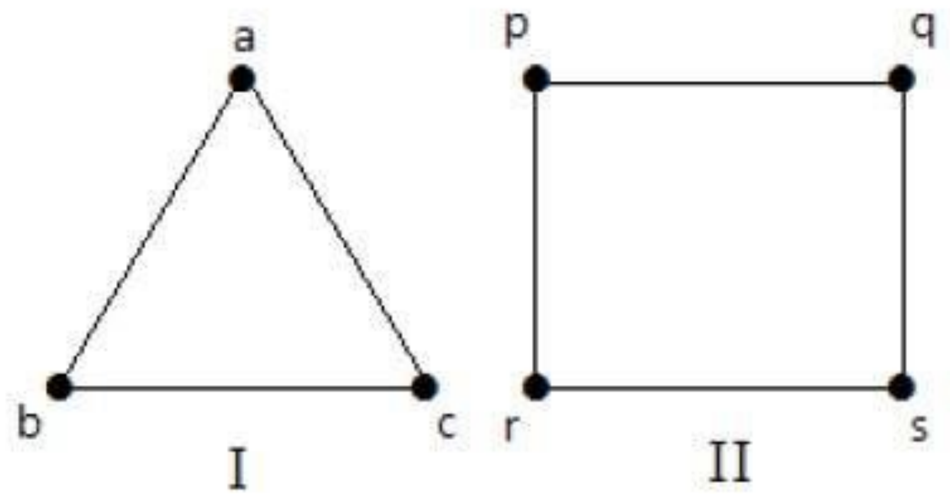
- This graph consists of two independent components which are disconnected.
- It is not possible to visit from the vertices of one component to the vertices of other component.
- Therefore, it is a disconnected graph.

7) Regular Graph. A graph G is said to be regular, **if all its vertices have the same degree**. In a graph, if the degree of each vertex is ‘k’, then the graph is called a ‘k-regular graph’.

Key Idea. A graph in which degree of all the vertices is same is called as a regular graph. If all the vertices in a graph are of degree ‘k’, then it is called as a “**k-regular graph**“.

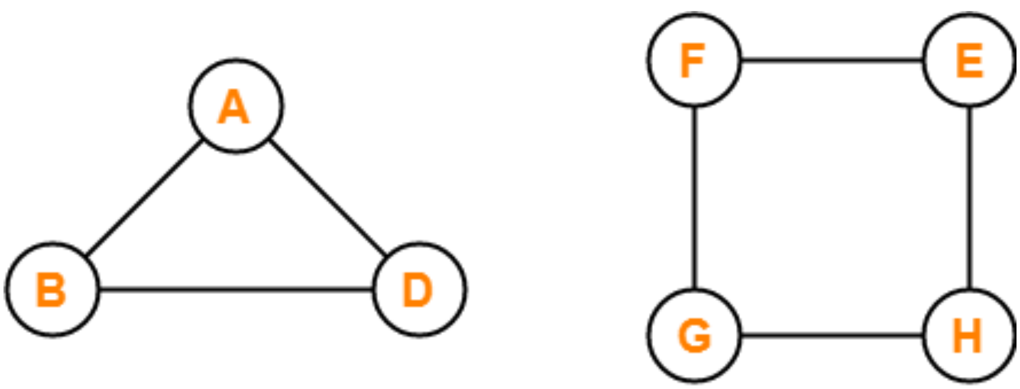
Example 1

In the following graphs, all the vertices have the same degree. So these graphs are called regular graphs.



In both the graphs, all the vertices have degree 2. They are called 2-Regular Graphs.

Example 2



Examples of Regular Graph

In these graphs,

- All the vertices have degree-2.
- Therefore, they are 2-Regular graphs.

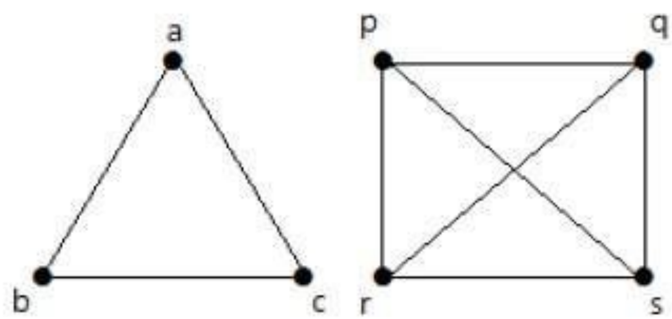
8) Complete Graph. A simple graph with ‘n’ mutual vertices is called a complete graph and it is **denoted by ‘K_n’**. In the graph, **a vertex should have edges with all other vertices**, then it called a complete graph.

In other words, if a vertex is connected to all other vertices in a graph, then it is called a complete graph.

Key idea. A graph in which exactly one edge is present between every pair of vertices is called as a complete graph. A complete graph of ‘n’ vertices contains exactly $^nC_2$ edges. A complete graph of ‘n’ vertices is represented as $\mathbf{K_n}$.

Example 1

In the following graphs, each vertex in the graph is connected with all the remaining vertices in the graph except by itself.



I II

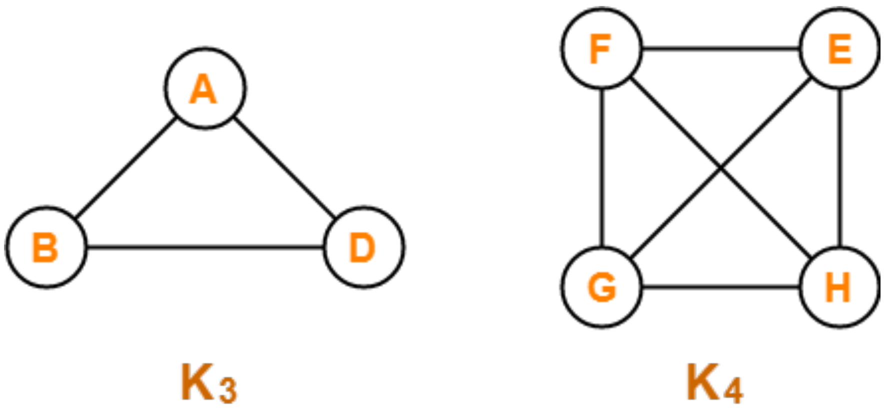
In graph I,

	a	b	c
a	Not Connected	Connected	Connected
b	Connected	Not Connected	Connected
c	Connected	Connected	Not Connected

In graph II,

	p	q	r	s
p	Not Connected	Connected	Connected	Connected
q	Connected	Not Connected	Connected	Connected
r	Connected	Connected	Not Connected	Connected
s	Connected	Connected	Connected	Not Connected

Example 2



Examples of Complete Graph

In these graphs,

- Each vertex is connected with all the remaining vertices through exactly one edge.
- Therefore, they are complete graphs.

9) Cycle Graph. A simple graph with 'n' vertices (n >= 3) and 'n' edges is called a cycle graph if all its edges form a cycle of length 'n'.

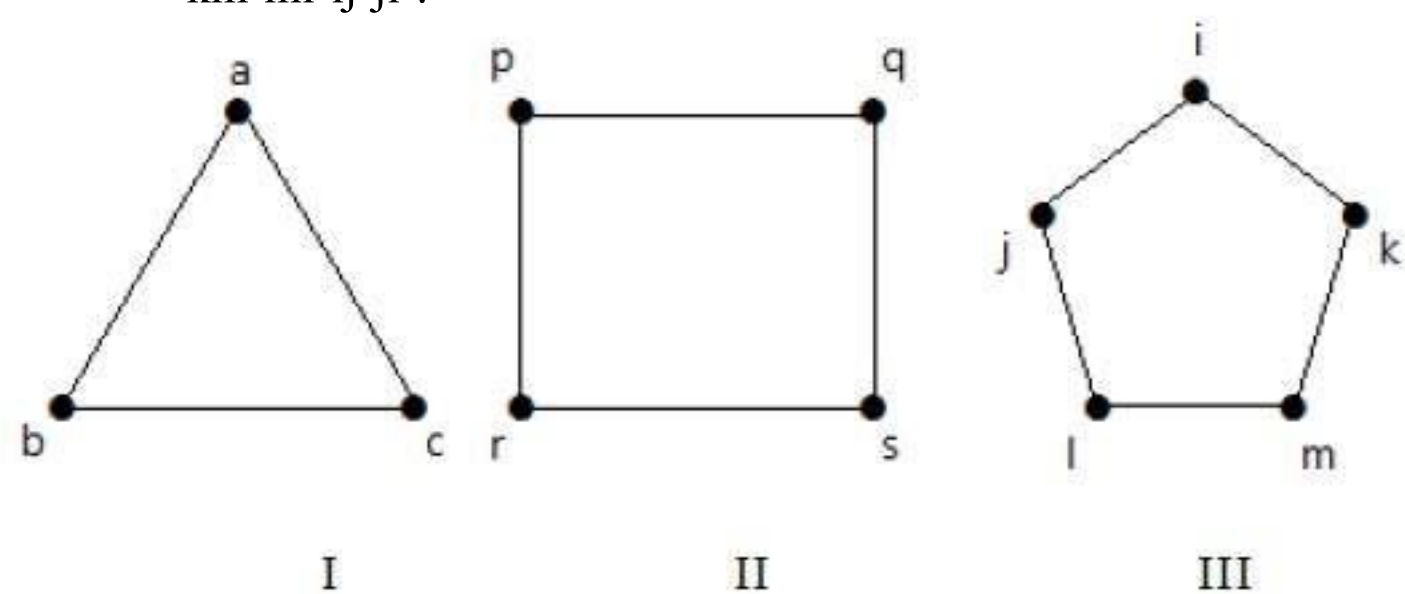
If the degree of each vertex in the graph is two, then it is called a Cycle Graph. **Notation** − C_n

Key Idea. A simple graph of 'n' vertices (n>=3) and n edges forming a cycle of length 'n' is called as a cycle graph. In a cycle graph, all the vertices are of degree 2.

Example 1

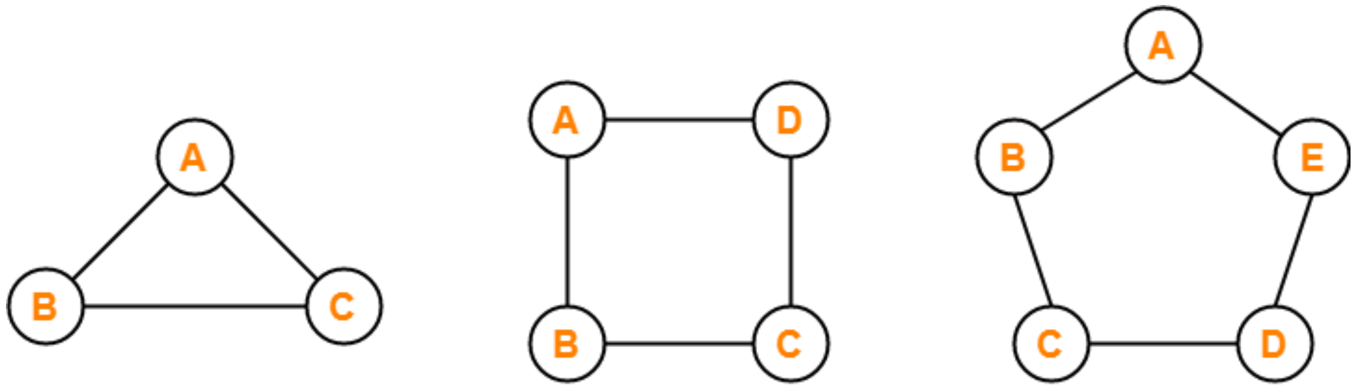
Take a look at the following graphs −

- Graph I has 3 vertices with 3 edges which is forming a cycle 'ab-bc-ca'.
- Graph II has 4 vertices with 4 edges which is forming a cycle 'pq-qs-sr-rp'.
- Graph III has 5 vertices with 5 edges which is forming a cycle 'ik-km-ml-lj-ji'.



Hence all the given graphs are cycle graphs.

Example 2



Examples of Cycle Graph

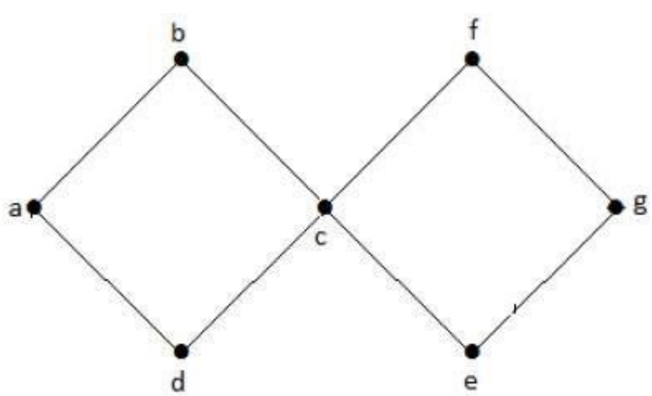
In these graphs,

- Each vertex is having degree 2.
- Therefore, they are cycle graphs.

10) **Cyclic Graph.** A graph **with at least one** cycle is called a cyclic graph.

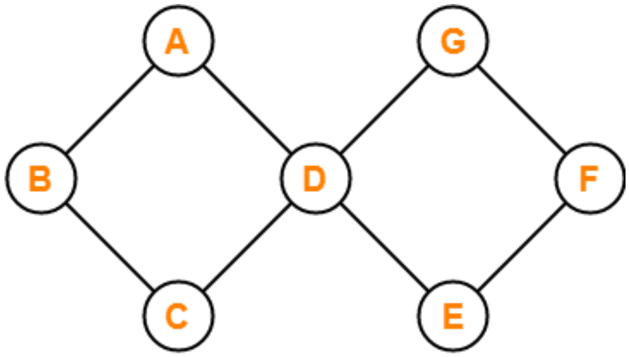
Key Idea. A graph containing at least one cycle in it is called as a cyclic graph.

Example 1



In the above example graph, we have two cycles a-b-c-d-a and c-f-g-e-c. Hence it is called a ***cyclic graph.***

Example 2



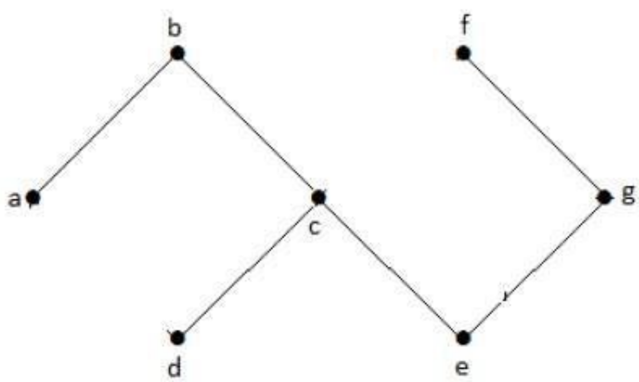
Example of Cyclic Graph

Here,

- This graph contains two cycles in it.
- Therefore, it is a cyclic graph.

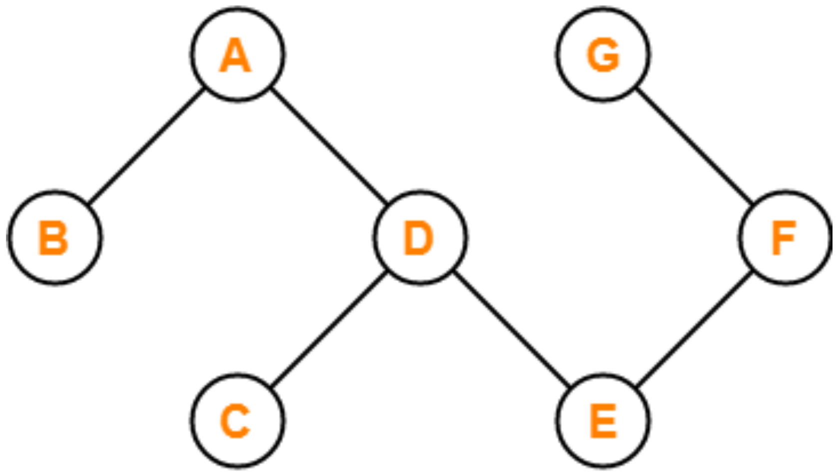
11) Acyclic Graph. A graph **with no cycles** is called an acyclic graph.
Key Idea. A graph not containing any cycle in it is called as an acyclic graph.

Example 1



In the above example graph, we do not have any cycles. Hence it is a non-cyclic graph.

Example 2



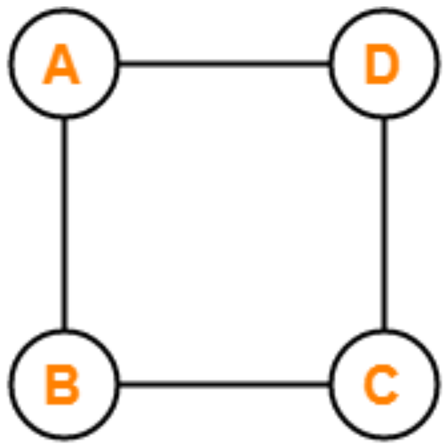
Example of Acyclic Graph

Here,

- This graph do not contain any cycle in it.
- Therefore, it is an acyclic graph.

12) Finite Graph. A graph consisting of finite number of vertices and edges is called as a finite graph.

Example 1

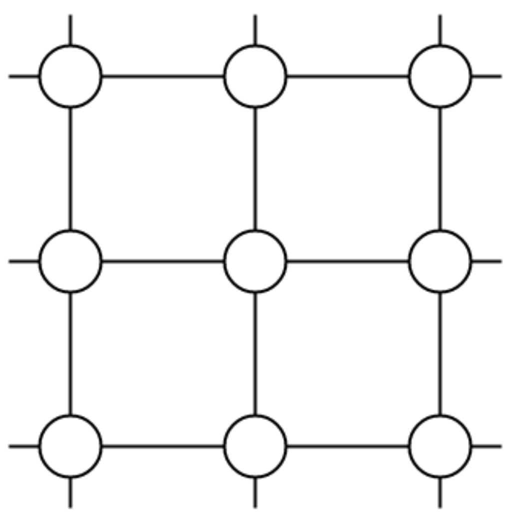


Example of Finite Graph

Here,

- This graph consists of finite number of vertices and edges.
- Therefore, it is a finite graph.

13) Infinite Graph. A graph consisting of infinite number of vertices and edges is called as an infinite graph.



Example of Infinite Graph

Here,

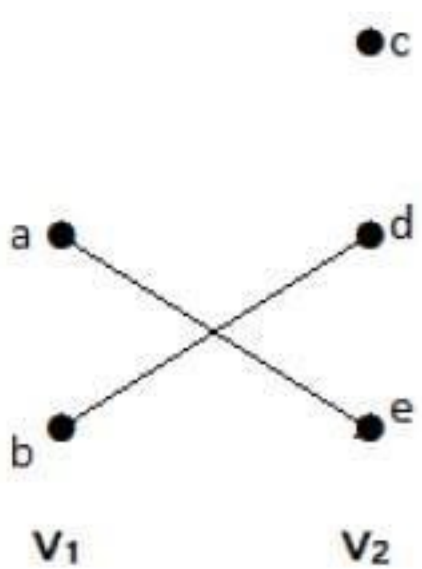
- This graph consists of infinite number of vertices and edges.
- Therefore, it is an infinite graph.

14) Bipartite Graph. A simple graph G = (V, E) with vertex partition V = {V_1, V_2} is called a bipartite graph **if every edge of E joins a vertex in V1 to a vertex in V_2.**

In general, a Bipartite graph has two sets of vertices, let us say, V1 and V2, and if an edge is drawn, it should connect any vertex in set V_1 to any vertex in set V_2.

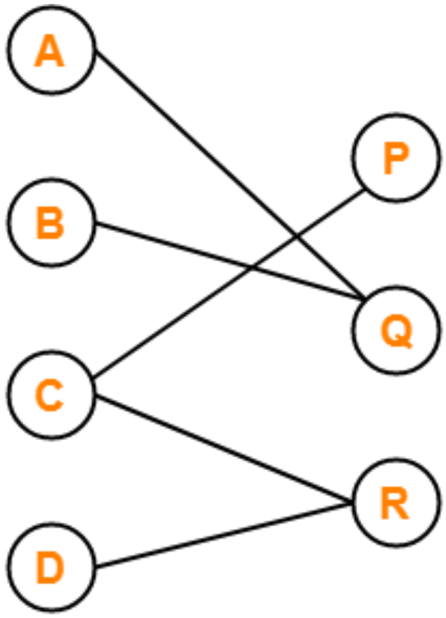
Key Idea. A bipartite graph is a graph where- Vertices can be divided into two sets X and Y. The vertices of set X only join with the vertices of set Y. None of the vertices belonging to the same set join each other.

Example 1



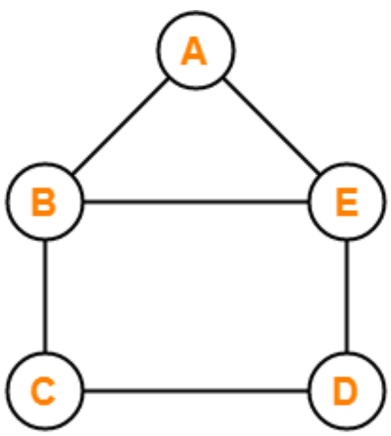
In this graph, you can observe two sets of vertices − V_1 and V_2. Here, two edges named 'ae' and 'bd' are connecting the vertices of two sets V_1 and V_2.

Example 1



Example of Bipartite Graph

15) Planar Graph. A planar graph is a graph that we can draw in a plane such that no two edges of it cross each other.



Example of Planar Graph

Here,

- This graph can be drawn in a plane without crossing any edges.
- Therefore, it is a planar graph.

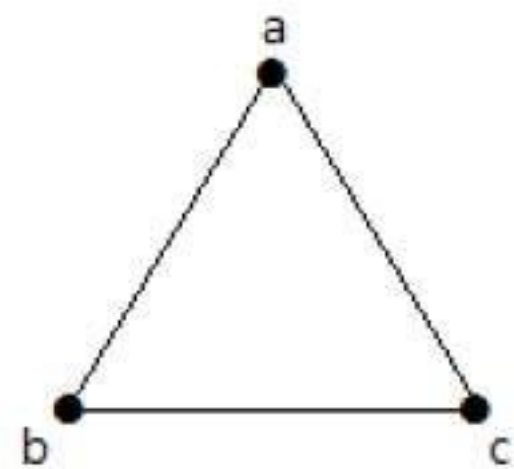
16) **Simple Graph.** A graph **with no loops** and no **parallel edges** is called a simple graph.

- The maximum number of edges possible in a single graph with 'n' vertices is ${}^{n}C_2$ where ${}^{n}C_2 = n\,(n - 1)/2$.
- The number of simple graphs possible with 'n' vertices = $2^{{}^{n}c_2} = 2^{n(n-1)/2}$.

Key Idea. A graph having no self-loops and no parallel edges in it is called as a simple graph.

Example 1

In the following graph, there are 3 vertices with 3 edges which is maximum excluding the parallel edges and loops. This can be proved by using the above formulae.



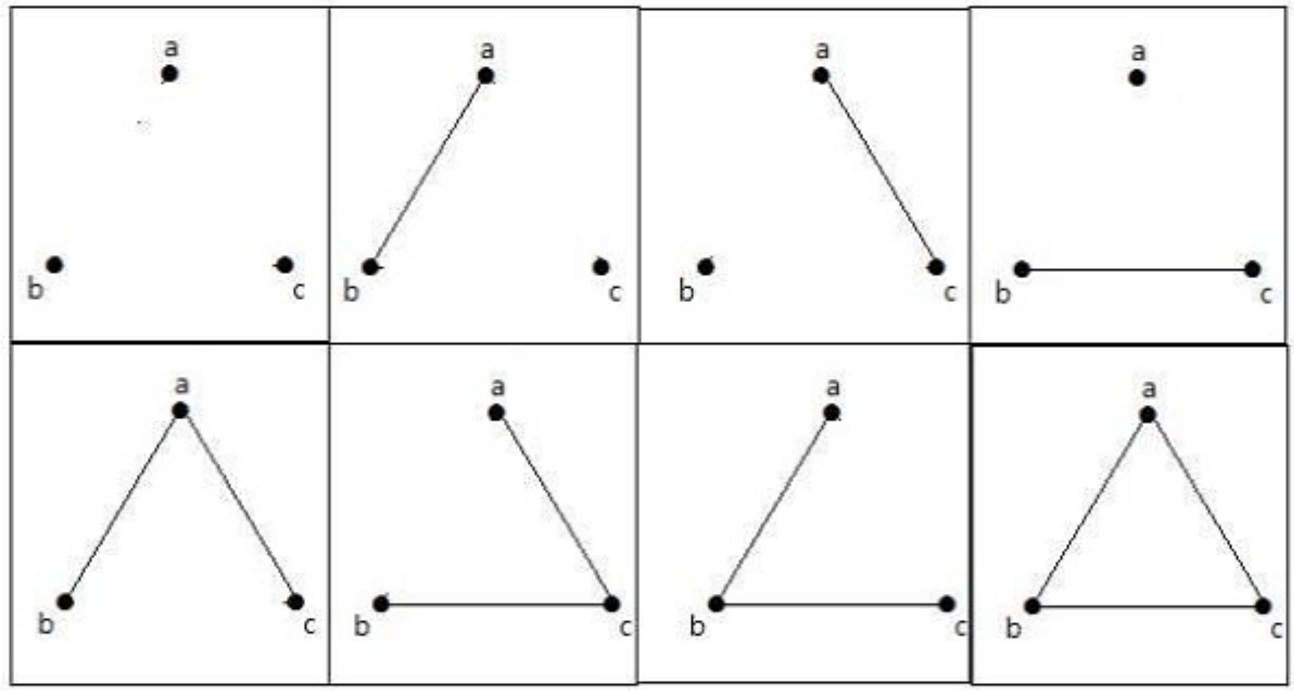
The maximum number of edges with n=3 vertices –

$$\begin{aligned} {}^{n}C_2 &= n(n-1)/2 \\ &= 3(3-1)/2 \\ &= 6/2 \\ &= 3 \text{ edges} \end{aligned}$$

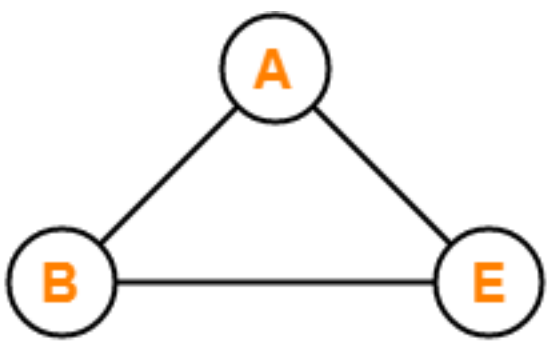
The maximum number of simple graphs with n=3 vertices –

$$\begin{aligned} 2^{{}^{n}C_2} &= 2^{n(n-1)/2} \\ &= 2^{3(3-1)/2} \\ &= 2^{3} \\ &8 \end{aligned}$$

These 8 graphs are as shown below –



Example 2

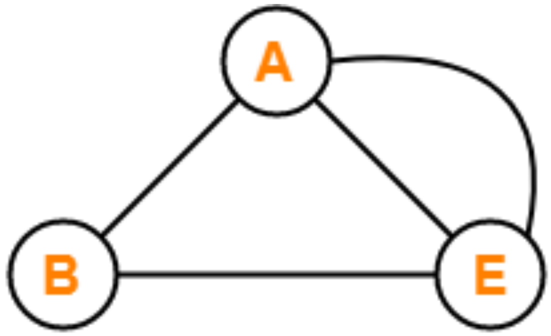


Example of Simple Graph

Here,

- This graph consists of three vertices and three edges.
- There are neither self-loops nor parallel edges.
- Therefore, it is a simple graph.

16) **Multi Graph.** A graph having no self-loops but having parallel edge(s) in it is called as a multi graph.

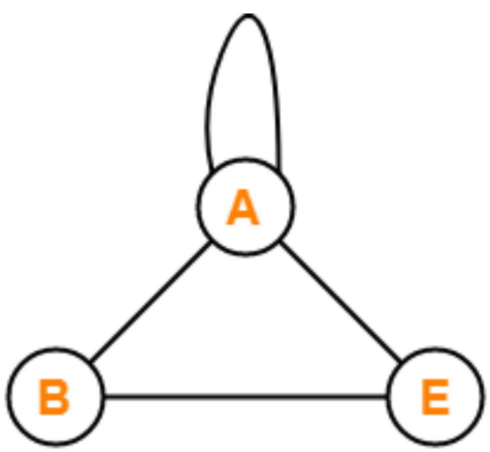


Example of Multi Graph

Here,

- This graph consists of three vertices and four edges out of which one edge is a parallel edge.
- There are no self-loops, but a parallel edge is present.
- Therefore, it is a multi-graph.

16) **Pseudo Graph.** A graph having no parallel edges but having self loop(s) in it is called as a pseudo graph.

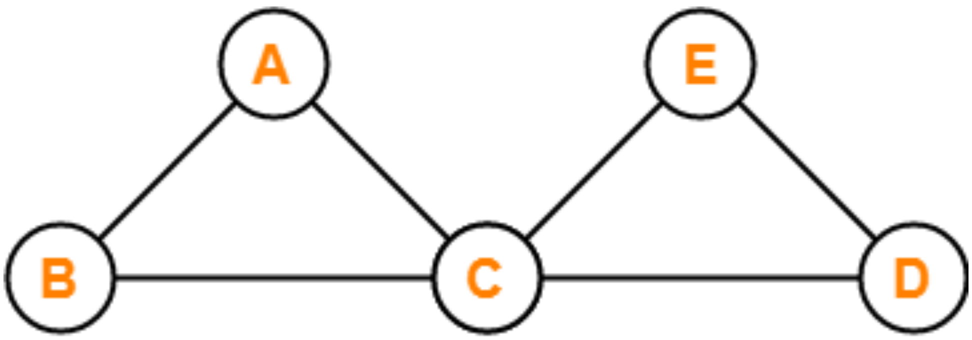


Example of Pseudo Graph

Here,

- This graph consists of three vertices and four edges out of which one edge is a self-loop.
- There are no parallel edges but a self-loop is present.
- Therefore, it is a pseudo graph.

17) **Euler Graph.** It is a connected graph in which all the vertices are even degree.



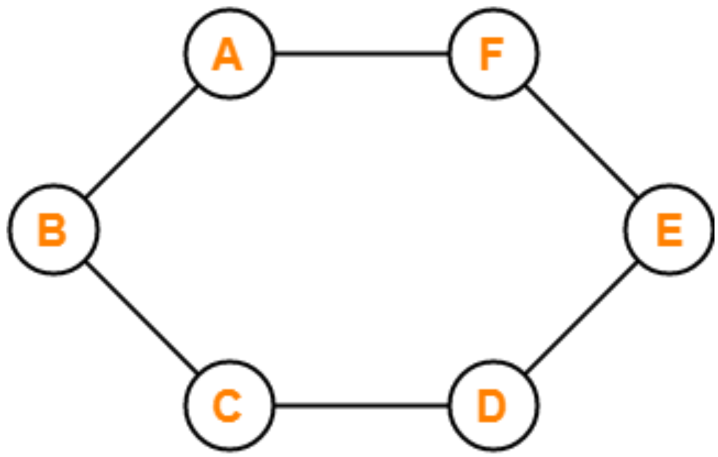
Example of Euler Graph

Here,

- This graph is a connected graph.
- The degree of all the vertices is even.
- Therefore, it is a Euler graph.

18) ***Hamiltonian Graph.*** If there exists a closed walk in the connected graph that visits every vertex of the graph exactly once (except starting vertex) without repeating the edges, then such a graph is called as a Hamiltonian graph.

Example 1



Example of Hamiltonian Graph

Here,

- This graph contains a closed walk ABCDEFG that visits all the vertices (except starting vertex) exactly once.
- All the vertices are visited without repeating the edges.
- Therefore, it is a Hamiltonian Graph.

19) Wheel Graph. A wheel graph is obtained from a cycle graph C_{n-1} by adding a new vertex. That new vertex is called a ***Hub*** which is connected to all the vertices of C_n.

Notation − W_n

No. of edges in W_n = No. of edges from hub to all other vertices +

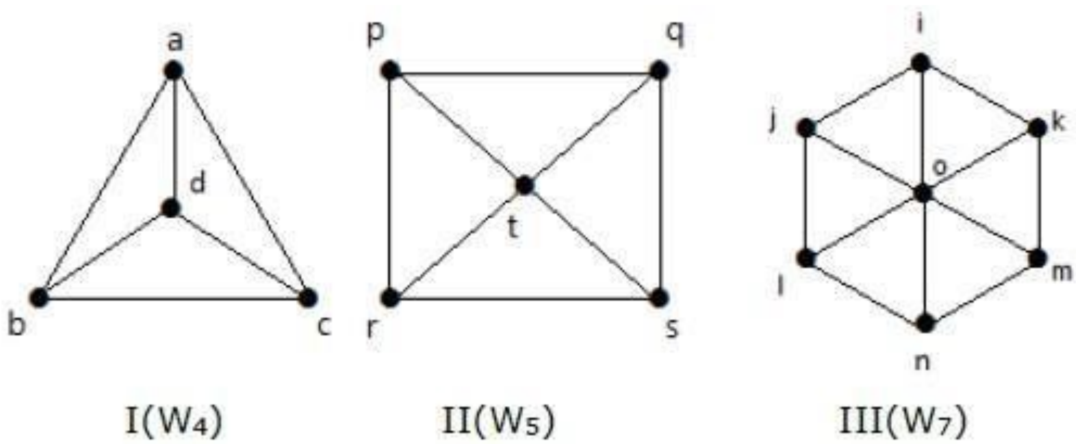
No. of edges from all other nodes in cycle graph without a hub.

= (n–1) + (n–1)

= 2(n–1)

Example 1

Take a look at the following graphs. They are all wheel graphs.



In graph I, it is obtained from C_3 by adding an vertex at the middle named as ‘d’. It is denoted as W_4.

Number of edges in W4 = 2(n-1) = 2(3) = 6

In graph II, it is obtained from C4 by adding a vertex at the middle named as ‘t’. It is denoted as W_5.

Number of edges in W5 = 2(n-1) = 2(4) = 8

In graph III, it is obtained from C_6 by adding a vertex at the middle named as ‘o’. It is denoted as W_7.

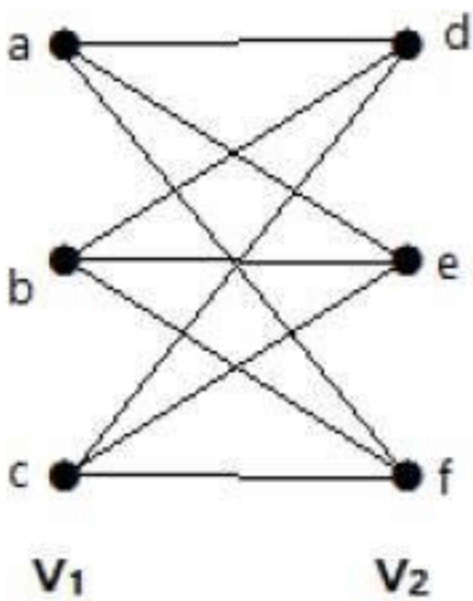
Number of edges in W4 = 2(n-1) = 2(6) = 12

20) ***Complete Bipartite Graph.*** A bipartite graph ‘G’, G = (V, E) with partition $V = \{V_1, V_2\}$ is said to be a complete bipartite graph if every vertex in V_1 is connected to every vertex of V_2.

In general, a complete bipartite graph connects each vertex from set V_1 to each vertex from set V_2.

Example 1

The following graph is a complete bipartite graph because it has edges connecting each vertex from set V_1 to each vertex from set V_2.



If $|V_1| = m$ and $|V_2| = n$, then the complete bipartite graph is denoted by $K_{m,n}$.

- $K_{m,n}$ has (m + n) vertices and (mn) edges.
- $K_{m,n}$ is a regular graph if m=n.

In general, a **complete bipartite graph is not a complete graph**.

$K_{m,n}$ is a complete graph if m=n=1.

The maximum number of edges in a bipartite graph with n vertices is −
$[n^2/4]$

If n=10, k5, 5= [n2/4] = $[10^2/4]$ = 25.

Similarly, K6, 4=24

K7, 3=21

K8, 2=16

K9, 1=9

If n=9, k5, 4 = [n2/4] = [92/4] = 20

Similarly, K6, 3=18

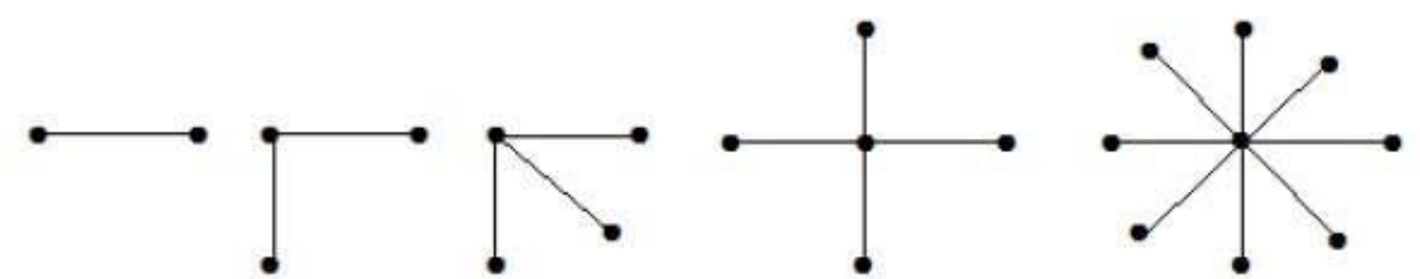
K7, 2=14

K8, 1=8

‘G’ is a bipartite graph if ‘G’ has no cycles of odd length. A special case of bipartite graph is a star graph.

21) ***Star Graph.*** A complete bipartite graph of the form K^1, n^{-1} is a star graph with n-vertices. A star graph is a complete bipartite graph if a single vertex belongs to one set and all the remaining vertices belong to the other set.

Example



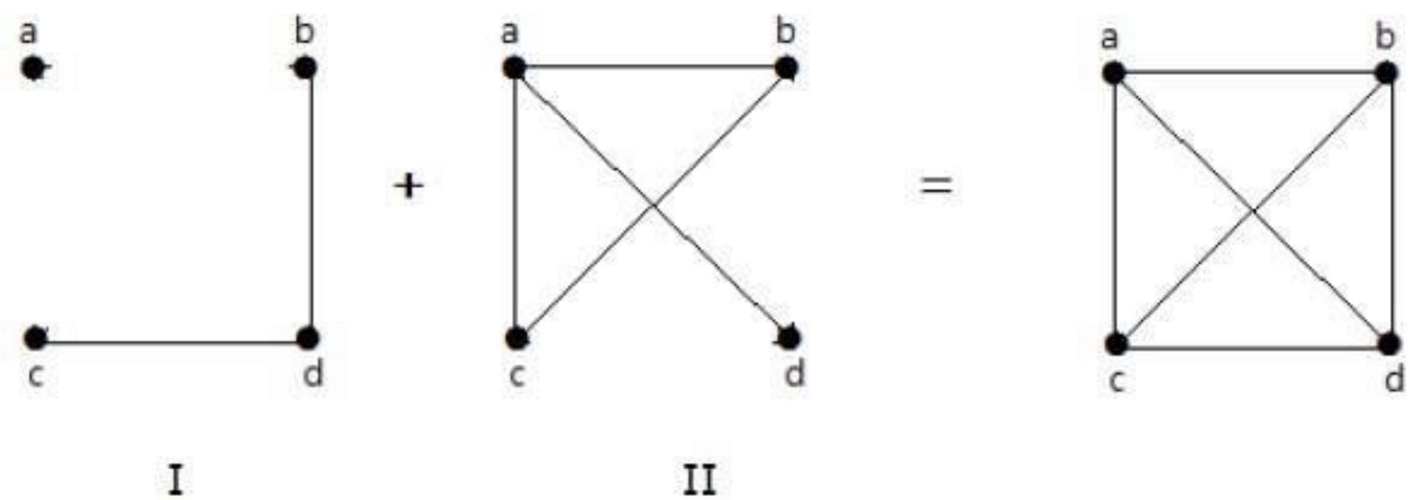
In the above graphs, out of ‘n’ vertices, all the ‘n–1’ vertices are connected to a single vertex. Hence it is in the form of K^1, $^{n-1}$ which are star graphs.

22) Complement of a Graph. Let 'G–' be a simple graph with some vertices as that of 'G' and an edge {U, V} is present in 'G–', if the edge is not present in G. It means, two vertices are adjacent in 'G–' if the two vertices are not adjacent in G.

If the edges that exist in graph I are absent in another graph II, and if both graph I and graph II are combined together to form a complete graph, then graph I and graph II are called complements of each other.

Example

In the following example, graph-I has two edges 'cd' and 'bd'. Its complement graph-II has four edges.



Note that the edges in graph-I are not present in graph-II and vice versa. Hence, the combination of both the graphs gives a complete graph of 'n' vertices.

Note − A combination of two complementary graphs gives a complete graph.

If 'G' is any simple graph, then

|E(G)| + |E('G-')| = |E(Kn)|, where n = number of vertices in the graph.

Example

Let ‘G’ be a simple graph with nine vertices and twelve edges, find the number of edges in 'G-'.

You have, |E(G)| + |E('G-')| = |E(Kn)|

12 + |E('G-')| =

$9(9-1) / 2 = {}^{9}C_{2}$

12 + |E('G-')| = 36

|E('G-')| = 24

‘G’ is a simple graph with 40 edges and its complement 'G–' has 38 edges. Find the number of vertices in the graph G or 'G–'.

Let the number of vertices in the graph be ‘n’.

We have, |E(G)| + |E('G-')| = |E(Kn)|

40 + 38 = n(n-1)/2

156 = n(n-1)

13(12) = n(n-1)

n = 13

Important Points-

- Edge set of a graph can be empty, but vertex set of a graph cannot be empty.
- Every polygon is a 2-Regular Graph.
- Every complete graph of ‘n’ vertices is a (n-1)-regular graph.
- Every regular graph need not be a complete graph.

Remember-

The following table is useful to remember different types of graphs-

	Self-Loop(s)	Parallel Edge(s)
Graph	Yes	Yes
Simple Graph	No	No
Multi Graph	No	Yes
Pseudo Graph	Yes	No

MyOpenMath:
https://www.myopenmath.com/assessment/watchvid.php?url=http%3A%2F%2Fyoutu.be%2FgbPFyGP_EzU%3Ft%26start%3D328%26end%3D378 (connected and disconnected)

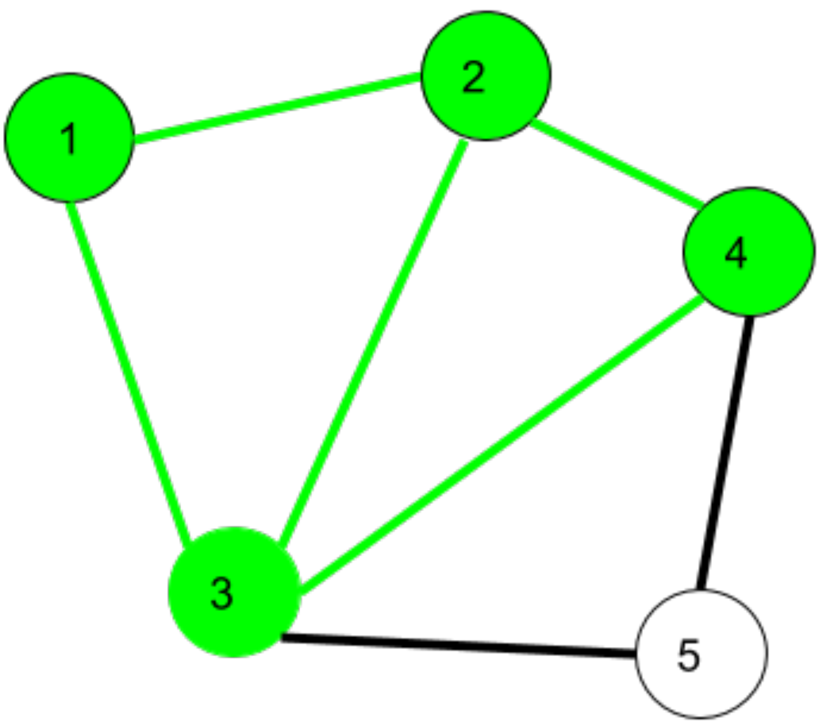
Walks, Trails, Paths, Cycles and Circuits in Graph

Source: https://www.geeksforgeeks.org/mathematics-walks-trails-paths-cycles-and-circuits-in-graph/
Video: https://www.youtube.com/watch?v=hlHWguJVAdU

1. 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.

Note: Vertices and Edges can be repeated.



Here, 1->2->3->4->2->1->3 is a walk.

Walk can be open or closed.

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.

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.

In the above diagram:

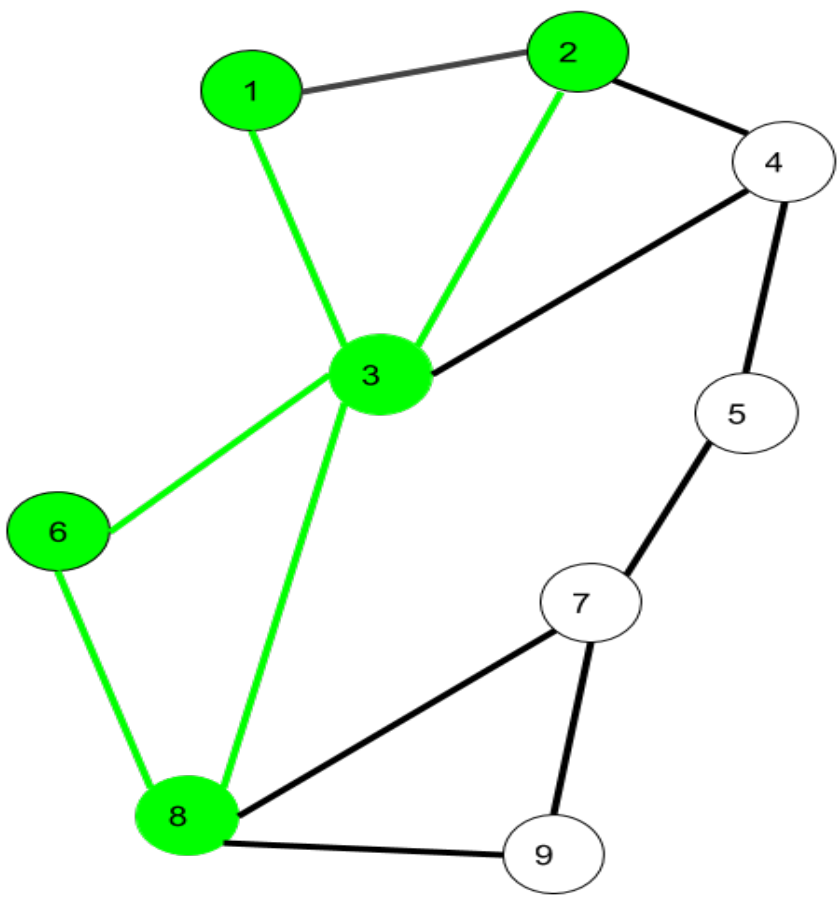
1->2->3->4->5->3 is an open walk.

1->2->3->4->5->3->1 is a closed walk.

2. Trail –

Trail is an open walk in which no edge is repeated.

Vertex can be repeated.



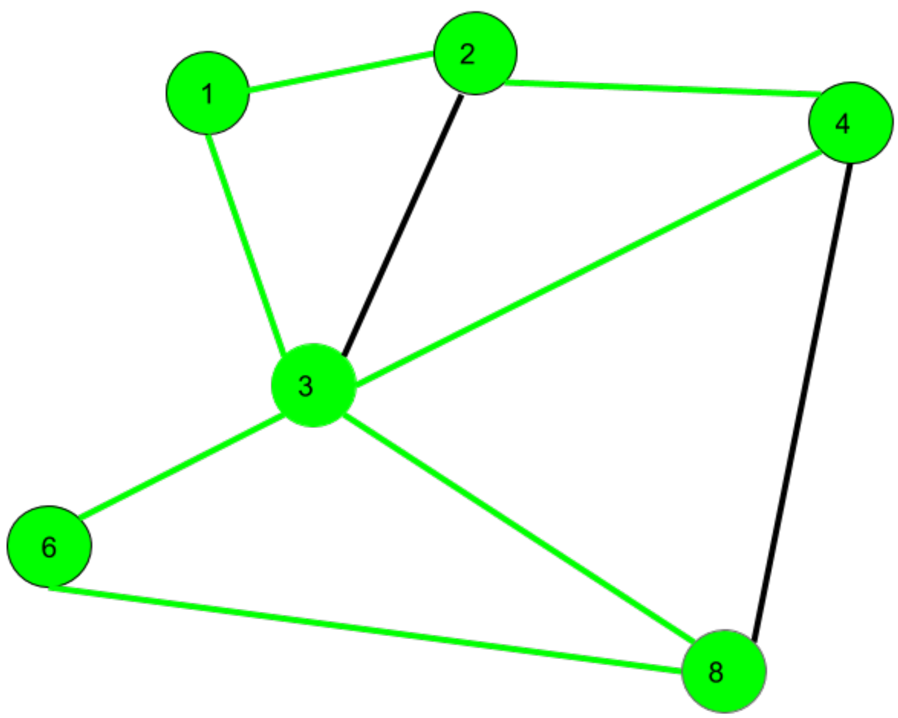
Here 1->3->8->6->3->2 is trail

Also 1->3->8->6->3->2->1 will be a closed trail

3. Circuit –

Traversing a graph such that not an edge is repeated but vertex can be repeated and it is closed also i.e. it is a closed trail.

Vertex can be repeated.
Edge can not be repeated.



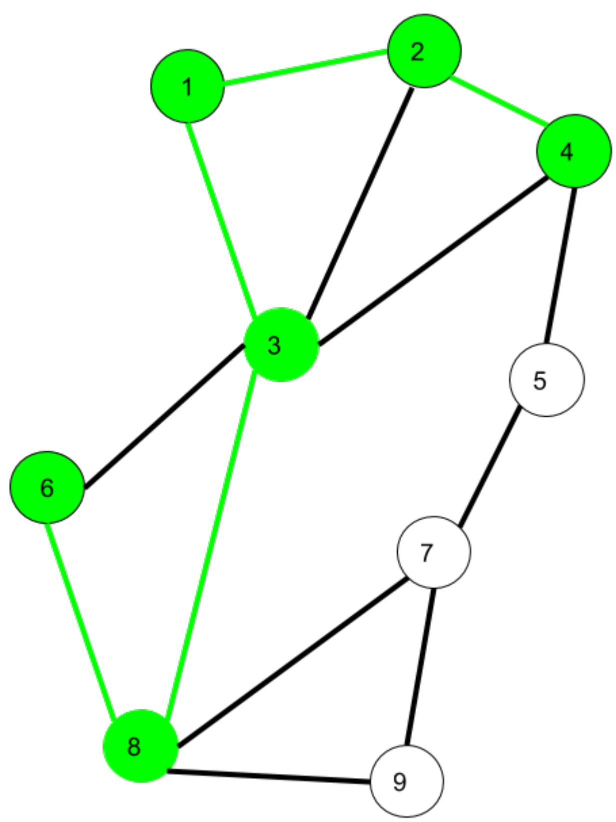
Here 1->2->4->3->6->8->3->1 is a circuit.

Circuit is a closed trail.

These can have repeated vertices only.

4. Path –

It is a trail in which neither vertices nor edges are repeated i.e. if we traverse a graph such that we do not repeat a vertex and nor we repeat an edge. As path is also a trail, thus it is also an open walk.
Another definition for path is a walk with no repeated vertex. This directly implies that no edges will ever be repeated and hence is redundant to write in the definition of path.



Vertex not repeated
Edge not repeated

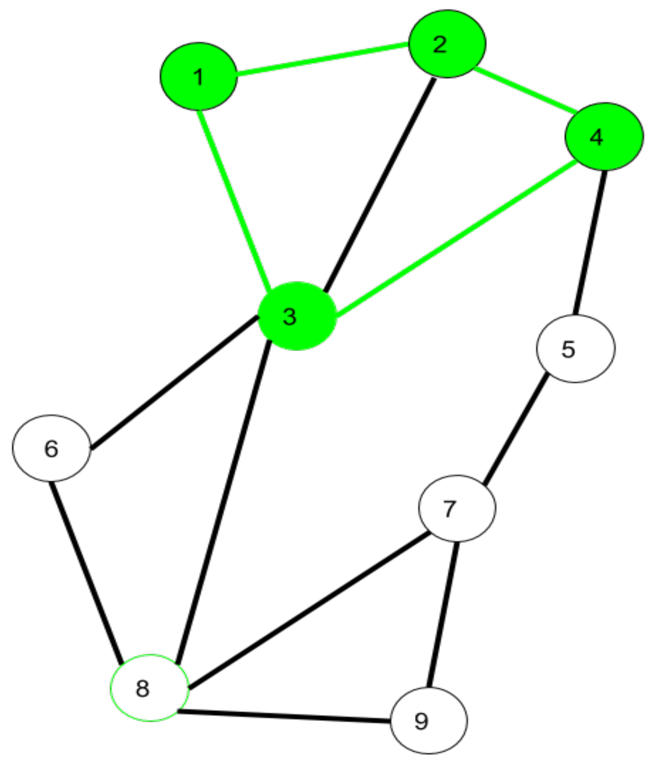
Here 6->8->3->1->2->4 is a Path

5. Cycle –

Traversing a graph such that we do not repeat a vertex nor we repeat a edge but the starting and ending vertex must be same i.e. we can repeat starting and ending vertex only then we get a cycle.

Vertex not repeated

Edge not repeated



Here 1->2->4->3->1 is a cycle.

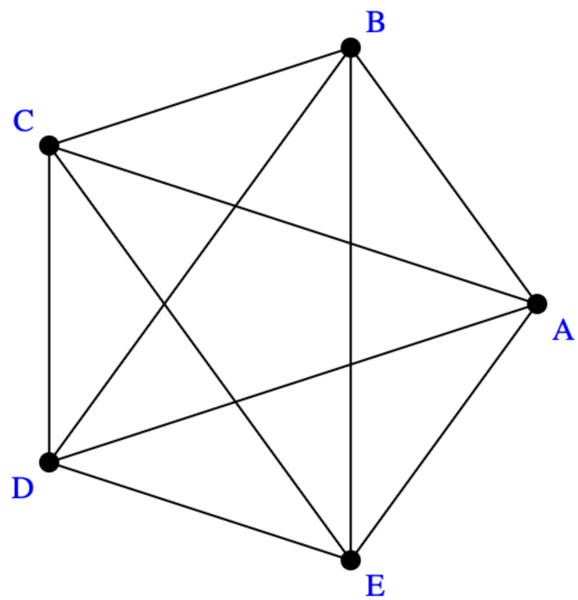
Cycle is a closed path.

These cannot have repeat anything (neither edges nor vertices).

Note that for closed sequences start and end vertices are the only ones that can repeat.

CHECK FOR UNDERSTANDING

Question 1



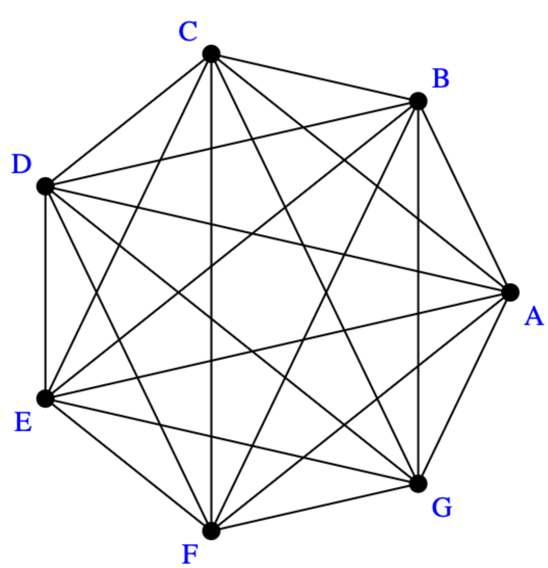
Using the given graph, determine if the following series of vertices gives you a walk, path, circuit, or none of these:

A→B→C→D→C→B.

A) a walk only
B) none of these
C) a walk, a path, and a circuit
D) a walk and a path only

Question 2

Using the given graph, determine if the following series of vertices gives you a walk, path, circuit, or none of these:

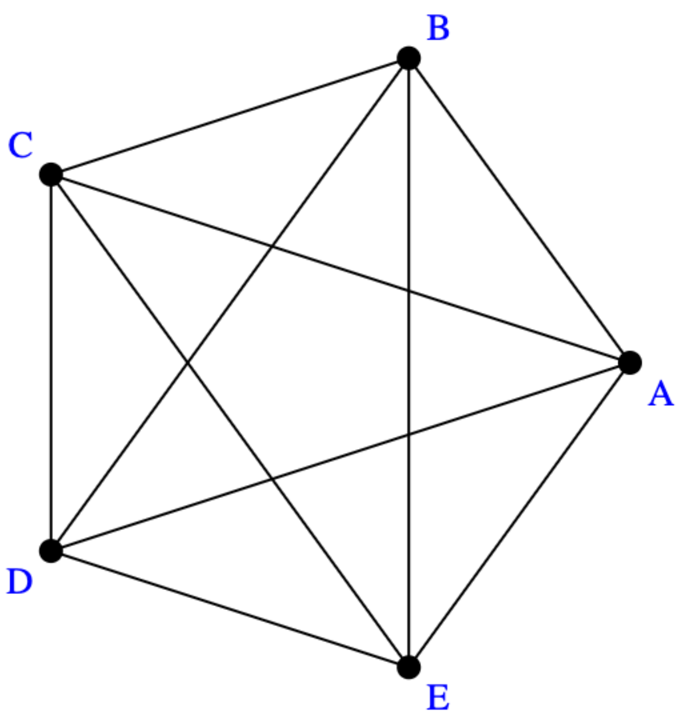


E→A→B→C→E→C→A.

A) a walk only
B) none of these
C) a walk, a path, and a circuit
D) a walk and a path only

3) Using the graph above, determine if the following series of vertices gives you a walk, path, circuit, or none of these:

B→D→E→A→B.



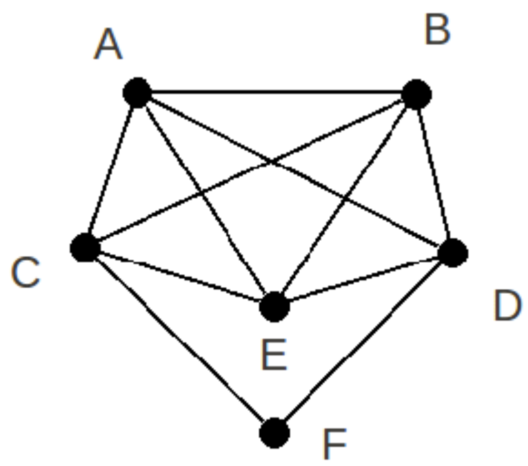
A) a walk only
B) none of these
C) a walk, a path, and a circuit
D) a walk and a path only

Graph Concepts and Terminology:

Order of a Network: the number of vertices in the entire network or graph
Adjacent Vertices: two vertices that are connected by an edge
Adjacent Edges: two edges that share a common vertex
Degree of a Vertex: the number of edges at that vertex
Path: a sequence of vertices with each vertex adjacent to the next one that starts and ends at different vertices and travels over any edge only once
Circuit: a path that starts and ends at the same vertex
Bridge: an edge such that if it were removed from a connected graph, the graph would become disconnected

Example 6.1.3: Graph Terminology

Figure 6.1.4: Graph 4



In the above graph the following is true:

Vertex A is adjacent to vertex B, vertex C, vertex D, and vertex E.

Vertex F is adjacent to vertex C, and vertex D.

Edge DF is adjacent to edge BD, edge AD, edge CF, and edge DE. The degrees of the vertices:

A	4
B	4
C	4
D	4
E	4
F	2

Here are some paths in the above graph: (there are many more than listed)
A, B, D
A, B, C, E
F, D, E, B, C

Here are some circuits in the above graph: (there are many more than listed)

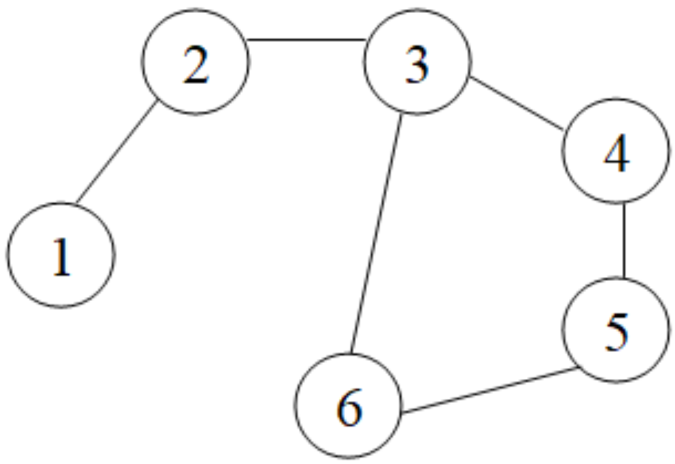
B, A, D, B
B, C, F, D, B
F, C, E, D, F

The above graph does not have any bridges.

Quick Idea.

In graph theory, a path that starts from a given vertex and ends at the same vertex is called **a cycle.**

In the example below, we can see that nodes 3-4-5-6-3 result in a cycle:



Quick Watch

Circuit: a path that starts and ends at the same vertex

https://www.youtube.com/watch?v=_YvlvSO7RBQ

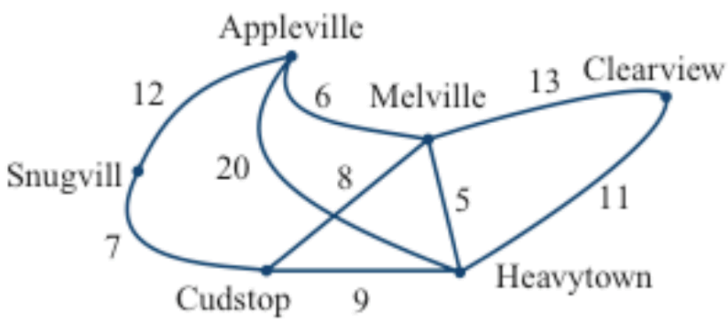
Networks

A network is a connection of vertices through edges. The internet is an example of a network with computers as the vertices and the connections between these computers as edges.

Spanning Subgraph: a graph that joins all of the vertices of a more complex graph, but does not create a circuit

Example 6.2.1: Spanning Subgraph

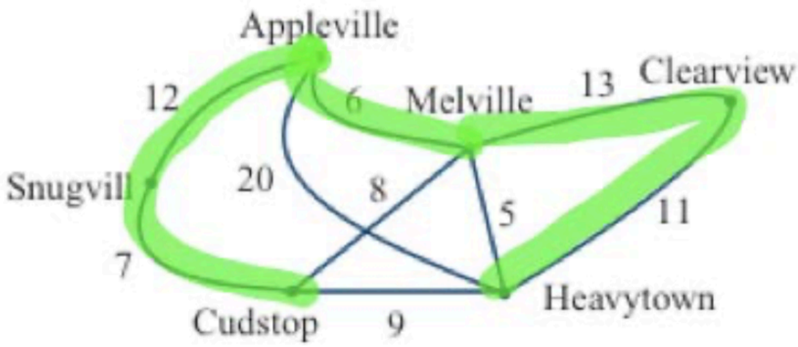
Figure 6.2.1: Map of Connecting Towns



This is a graph showing how six cities are linked by roads. This graph has many spanning subgraphs but two examples are shown below.

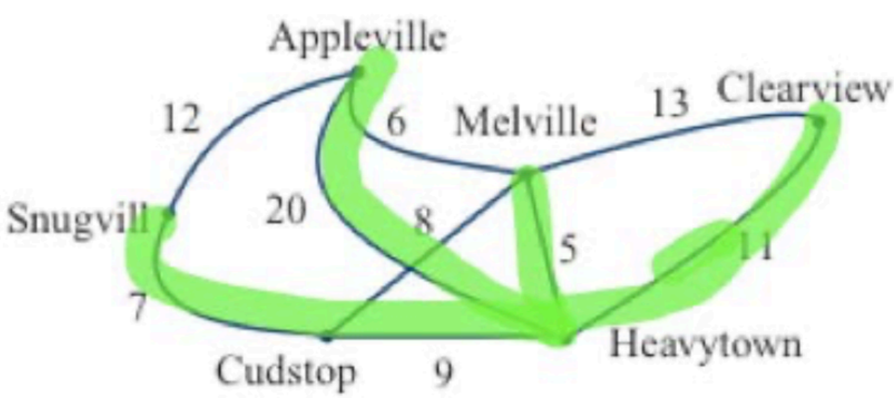
This graph spans all of the cities (vertices) of the original graph but does not contain any circuits.

Figure 6.2.2: Spanning Subgraph 1



This graph spans all of the cities (vertices) of the original graph, but does not contain any circuits.

Figure 6.2.3: Spanning Subgraph 2



TREES

Video: **https://www.youtube.com/watch?v=b233VKD6udo**

Trees are graphs that do not contain even a single cycle. They represent hierarchical structure in a graphical form. Trees belong to the simplest class of graphs. Despite their simplicity, they have a rich structure.

Trees provide a range of useful applications as simple as a family tree to as complex as trees in data structures of computer science.

Tree

A **connected acyclic graph** is called a tree. In other words, a connected graph with no cycles is called a tree.

The edges of a tree are known as **branches**. Elements of trees are called their nodes. The nodes without child nodes are called **leaf nodes**.

A tree with 'n' vertices has 'n-1' edges. If it has one more edge extra than 'n-1', then the extra edge should obviously has to pair up with two vertices which leads to form a cycle. Then, it becomes a cyclic graph which is a violation for the tree graph.

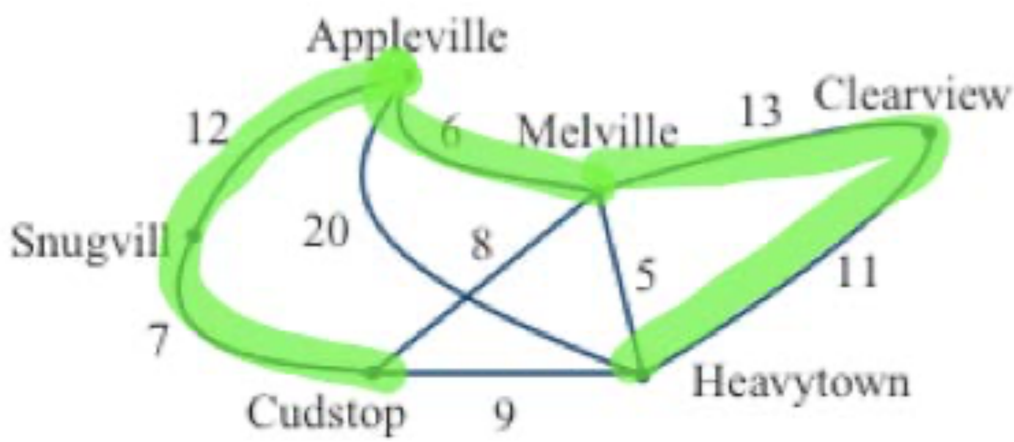
Key Idea. A tree is a graph that is connected and has no circuits. Therefore, a spanning subgraph is a tree and the examples of spanning subgraphs in Example 6.2.1 above are also trees.

Properties of Trees:

1. If a graph is a tree, there is one and only one path joining any two vertices. Conversely, if there is one and only one path joining any two vertices of a graph, the graph must be a tree.
2. In a tree, every edge is a bridge. Conversely, if every edge of a connected graph is a bridge, then the graph must be a tree.
3. A tree with N vertices must have N-1 edges.
4. A connected graph with N vertices and N-1 edges must be a tree.

Example 6.2.2: Tree Properties

Figure 6.2.2: Spanning Subgraph 1



Consider the spanning subgraph highlighted in green shown in Figure 6.2.2.

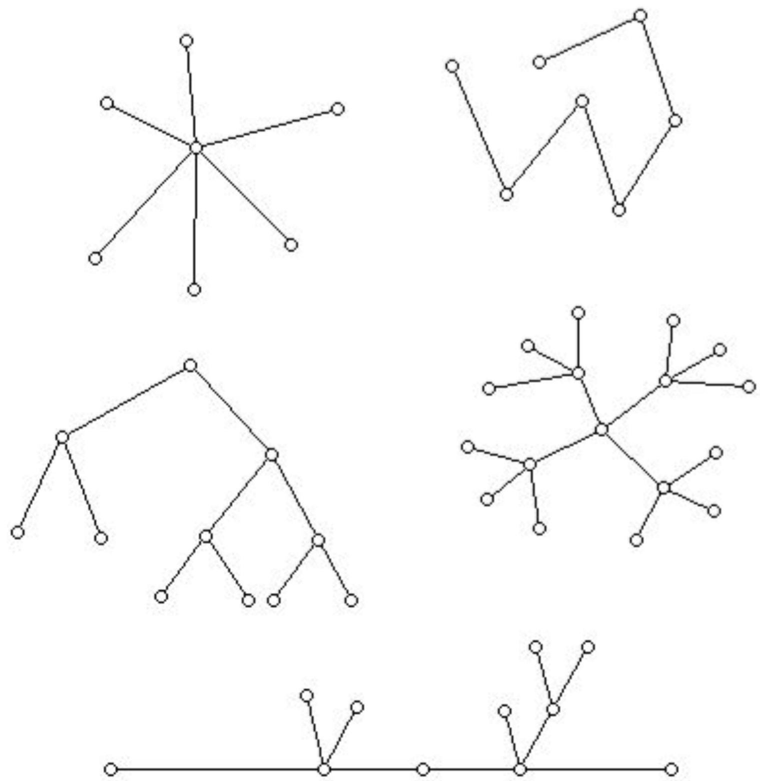
1. Tree Property 1
   Look at the vertices Appleville and Heavytown. Since the graph is a tree, there is only one path joining these two cities. Also, since there is only one path between any two cities on the whole graph, then the graph must be a tree.
2. Tree Property 2
   Since the graph is a tree, notice that every edge of the graph is a bridge, which is an edge such that if it were removed the graph would become disconnected.
3. Tree Property 3
   Since the graph is a tree and it has six vertices, it must have N – 1 or six – 1 = five edges.

4. Tree Property 4
   Since the graph is connected and has six vertices and five edges, it must be a tree.

Example 6.2.3: More Examples of Trees:

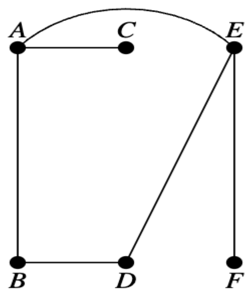
All of the graphs shown below are trees and they all satisfy the tree properties.

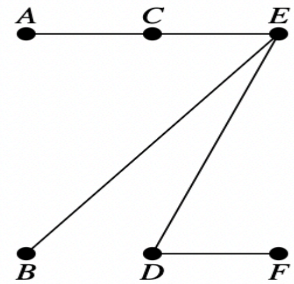
Figure 6.2.4: More Examples of Trees

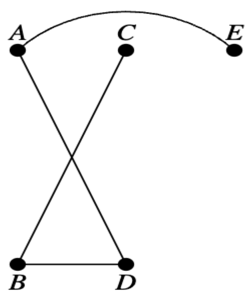


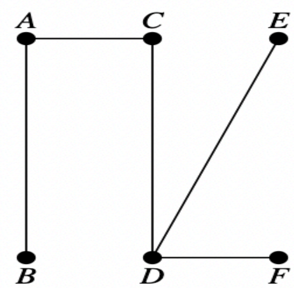
CHECK FOR UNDERSTANDING 2

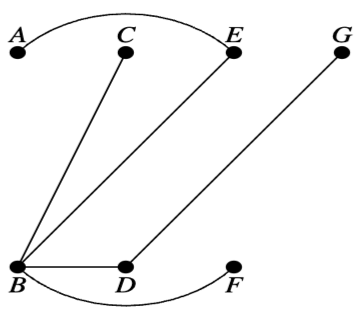
QUESTION #1 Which of the graphs is NOT a tree?

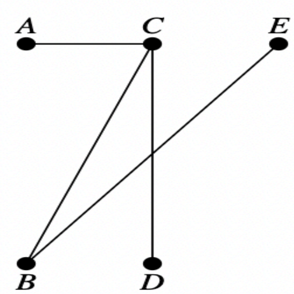
- ○ 

- ○ 

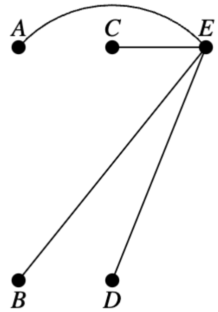
- ○ 

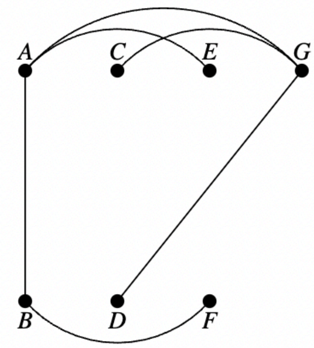
- ○ 

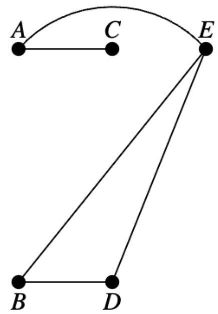
- ○ 

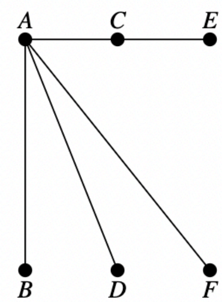
- ○ 

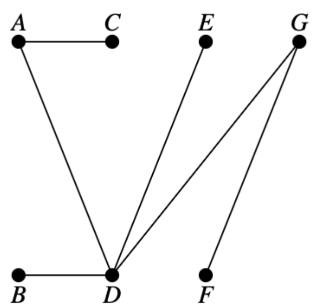
Question #2. Select all the graphs that are trees.

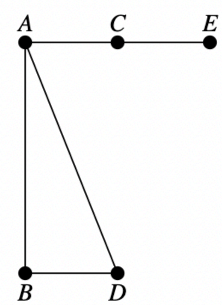
☐ 

☐ 

☐ 

☐ 

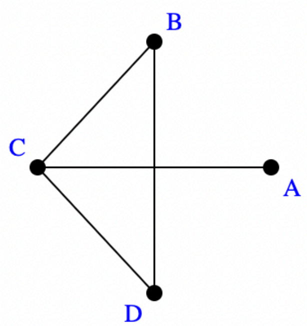
☐ 

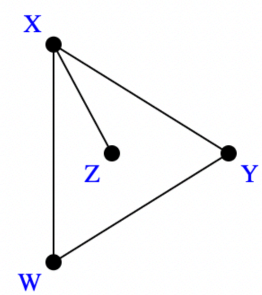
☐ 

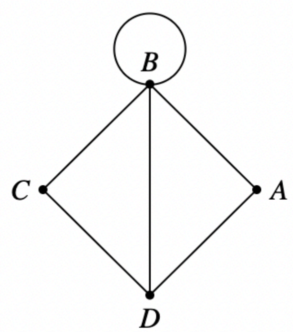
QUICK CHECK FOR UNDERSTANDING

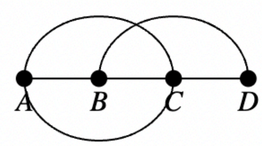
Question #1) Graphs are **equivalent** (or **isomorphic**) if they have the same number of vertices and the same edge connections. The vertices do *not* need to have the same labels, and they do not have to be drawn in the same positions.

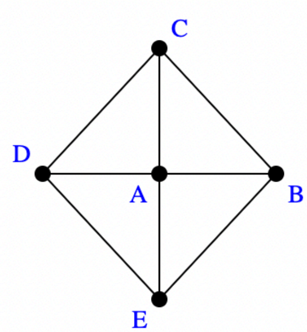
Select three equivalent graphs from the following.

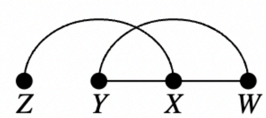




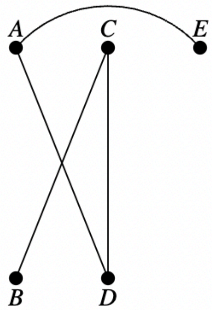


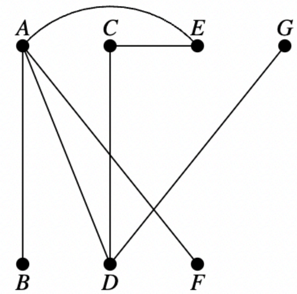


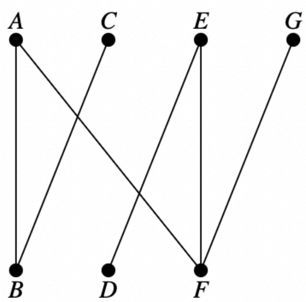


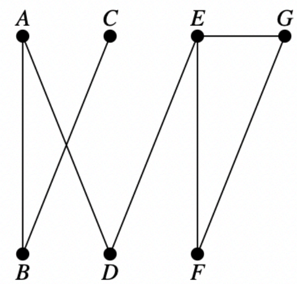


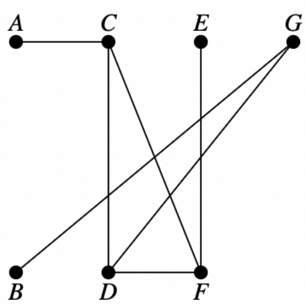
Question #2. Select all the graphs that are trees.

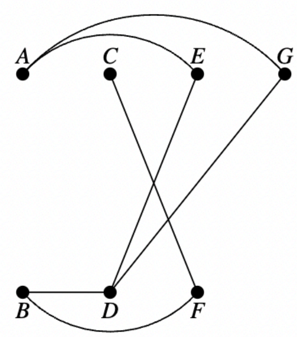




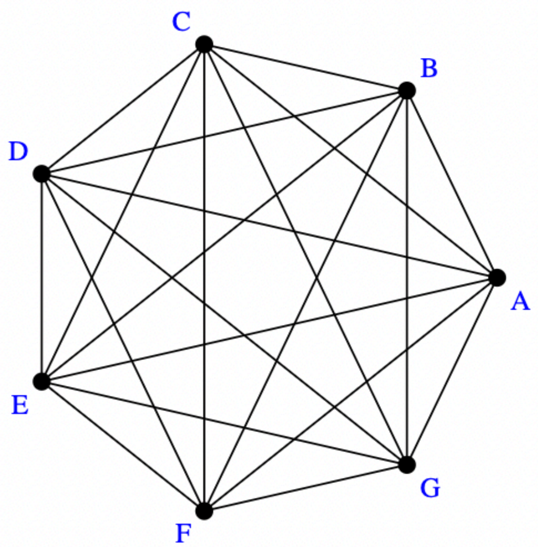








Question #3.

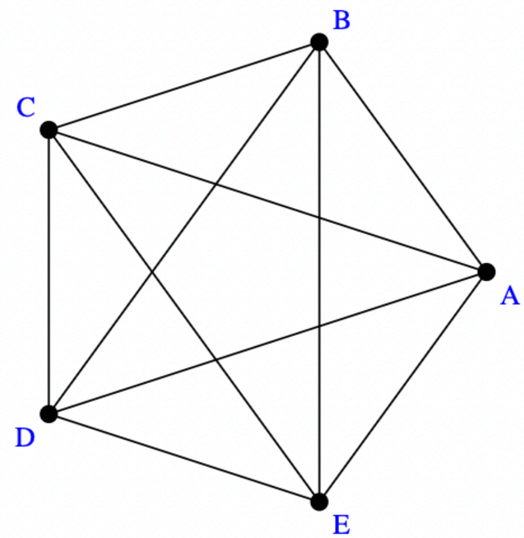


Using the graph above, determine if the following series of vertices gives you a walk, path, circuit, or none of these:

E→A→B→C→E→C→A.

- ○ a walk and a path only
- ○ a walk only
- ○ a walk, a path, and a circuit
- ○ none of these

Question #4.



Using the graph above, determine if the following series of vertices gives you a walk, path, circuit, or none of these:

B→D→E→A→B.

- a walk, a path, and a circuit
- a walk only
- none of these
- a walk and a path only

TREES

Video: **https://www.youtube.com/watch?v=b233VKD6udo**

Trees are graphs that do not contain even a single cycle. They represent hierarchical structure in a graphical form. Trees belong to the simplest class of graphs. Despite their simplicity, they have a rich structure.

Trees provide a range of useful applications as simple as a family tree to as complex as trees in data structures of computer science.

Tree

A **connected acyclic graph** is called a tree. In other words, a connected graph with no cycles is called a tree.

The edges of a tree are known as **branches**. Elements of trees are called their nodes. The nodes without child nodes are called **leaf nodes**.

A tree with 'n' vertices has 'n-1' edges. If it has one more edge extra than 'n-1', then the extra edge should obviously has to pair up with two vertices which leads to form a cycle. Then, it becomes a cyclic graph which is a violation for the tree graph.

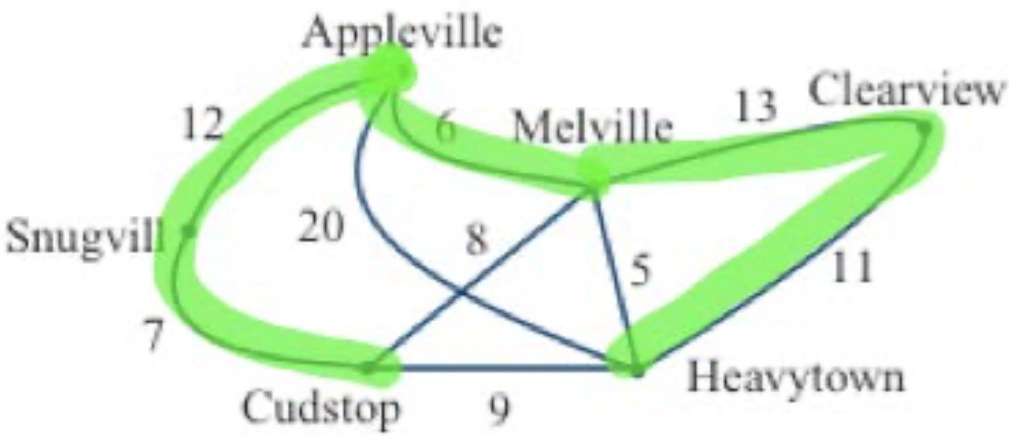
Key Idea. A tree is a graph that is connected and has no circuits. Therefore, a spanning subgraph is a tree and the examples of spanning subgraphs in Example 6.2.1 above are also trees.

Properties of Trees:

5. If a graph is a tree, there is one and only one path joining any two vertices. Conversely, if there is one and only one path joining any two vertices of a graph, the graph must be a tree.
6. In a tree, every edge is a bridge. Conversely, if every edge of a connected graph is a bridge, then the graph must be a tree.
7. A tree with N vertices must have N-1 edges.
8. A connected graph with N vertices and N-1 edges must be a tree.

Example 6.2.2: Tree Properties

Figure 6.2.2: Spanning Subgraph 1



Consider the spanning subgraph highlighted in green shown in Figure 6.2.2.

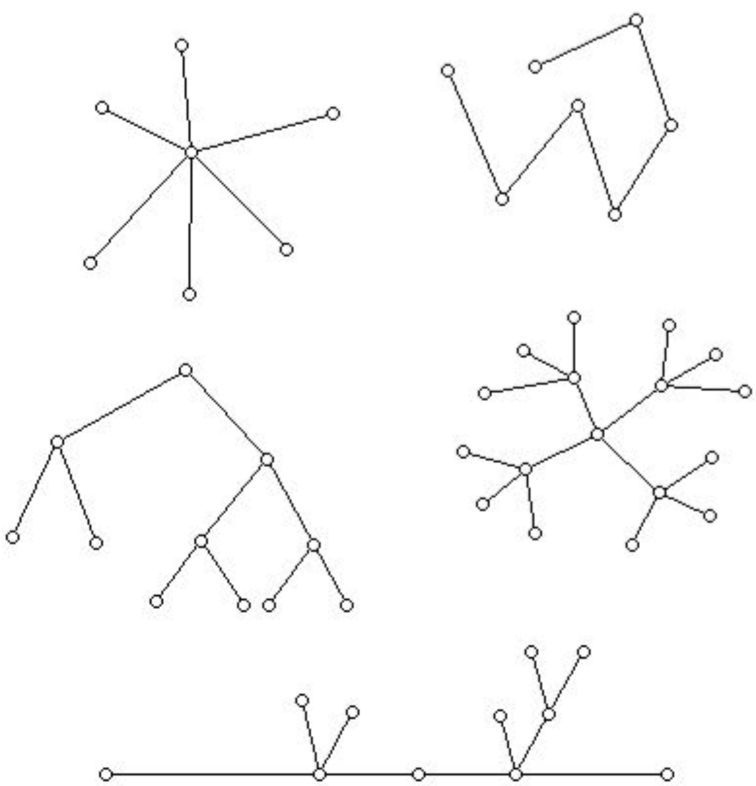
5. Tree Property 1
   Look at the vertices Appleville and Heavytown. Since the graph is a tree, there is only one path joining these two cities. Also, since there is only one path between any two cities on the whole graph, then the graph must be a tree.
6. Tree Property 2
   Since the graph is a tree, notice that every edge of the graph is a bridge, which is an edge such that if it were removed the graph would become disconnected.
7. Tree Property 3
   Since the graph is a tree and it has six vertices, it must have N – 1 or six – 1 = five edges.

8. Tree Property 4
   Since the graph is connected and has six vertices and five edges, it must be a tree.

Example 6.2.3: More Examples of Trees:

All of the graphs shown below are trees and they all satisfy the tree properties.

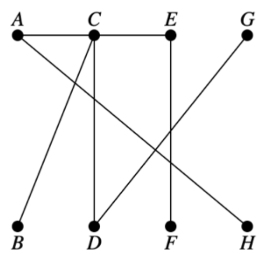
Figure 6.2.4: More Examples of Trees



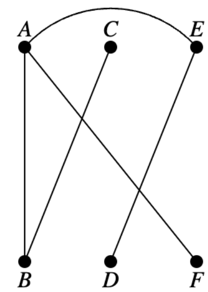
CHECK FOR UNDERSTANDING 1

Which of the graphs is NOT a tree?

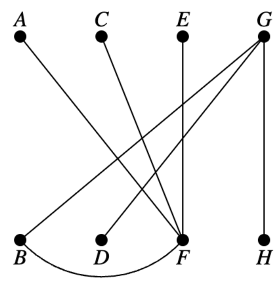
A)



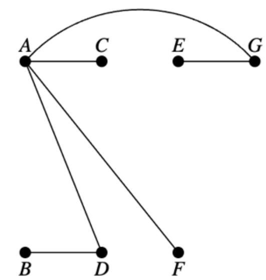
D)



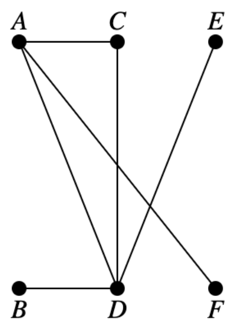
B)



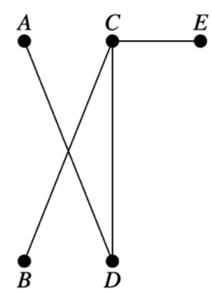
E)



C)



F)



2) T or F. A tree with 100 vertices has 101 edges

A) True

B) False

Minimum Cost Spanning Tree (India)

https://www.youtube.com/watch?v=4ZlRH0eK-qQ

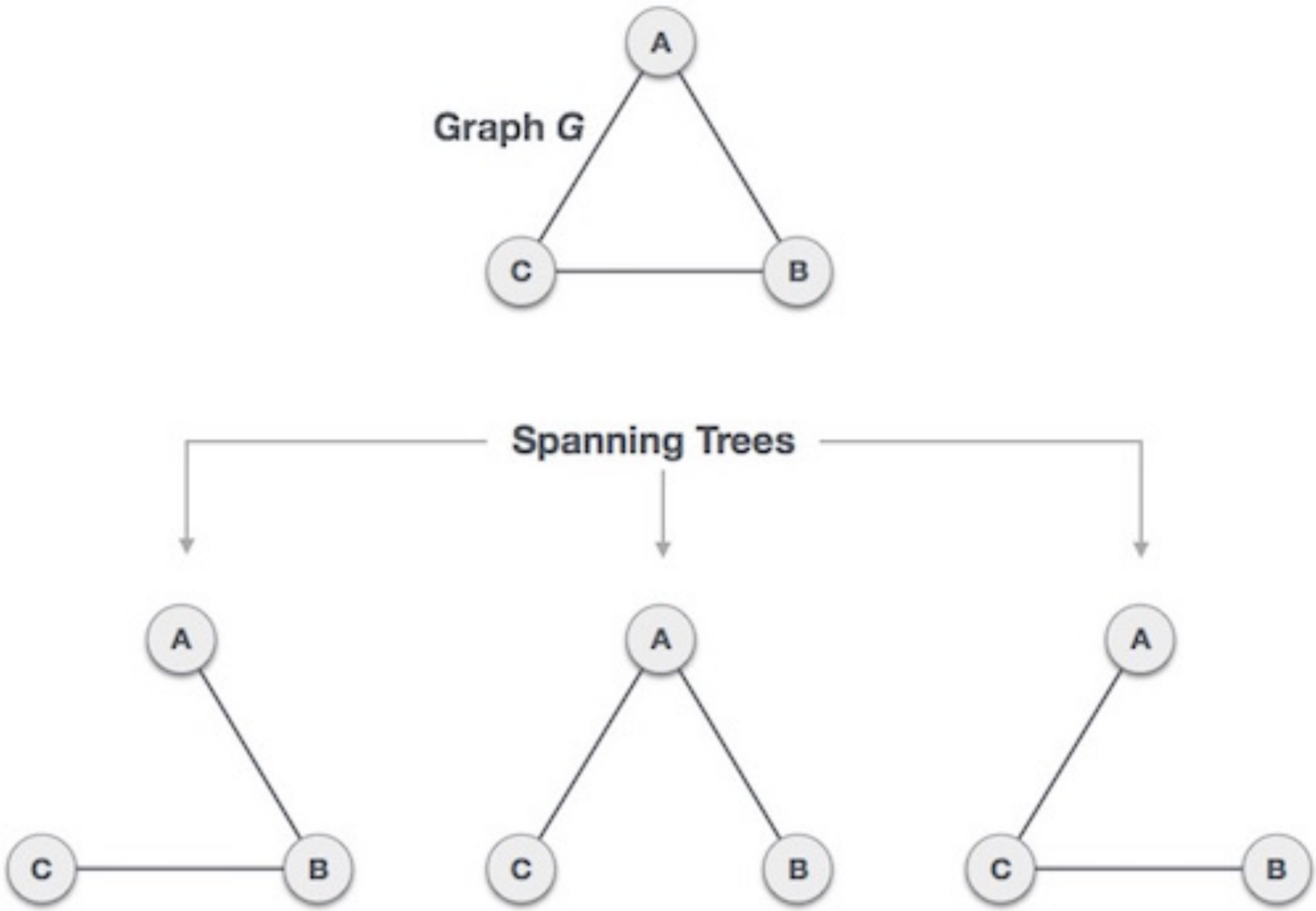
Spanning Tree

Source: https://www.tutorialspoint.com/data_structures_algorithms/spanning_tree.htm

Video: https://www.youtube.com/watch?v=0ljjRM8hWjU

A spanning tree is a subset of Graph G, which has all the vertices covered with minimum possible number of edges. Hence, a spanning tree does not have cycles and it cannot be disconnected.

By this definition, we can draw a conclusion that every connected and undirected Graph G has at least one spanning tree. A disconnected graph does not have any spanning tree, as it cannot be spanned to all its vertices.



Note: We found three spanning trees off one complete graph. A complete undirected graph can have maximum $\mathbf{n^{n-2}}$ number of spanning trees, where **n** is the number of nodes. In the above addressed example, **n is 3,** hence $\mathbf{3^{3-2} = 3}$ spanning trees are possible.

General Properties of Spanning Tree

We now understand that one graph can have more than one spanning tree. Following are a few properties of the spanning tree connected to graph G –

- A connected graph G can have more than one spanning tree.
- All possible spanning trees of graph G, have the same number of edges and vertices.

- The spanning tree does not have any cycle (loops).
- Removing one edge from the spanning tree will make the graph disconnected, i.e., the spanning tree is **minimally connected**.
- Adding one edge to the spanning tree will create a circuit or loop, i.e. the spanning tree is **maximally acyclic**.

Mathematical Properties of Spanning Tree

- Spanning tree has **n-1** edges, where **n** is the number of nodes (vertices).
- From a complete graph, by removing maximum **e – n + 1** edges, we can construct a spanning tree.
- A complete graph can have maximum $\mathbf{n^{n-2}}$ number of spanning trees.

Thus, we can conclude that spanning trees are a subset of connected Graph G and disconnected graphs do not have spanning tree.

Application of Spanning Tree

Spanning tree is basically used to find a minimum path to connect all nodes in a graph. Common applications of spanning trees are –

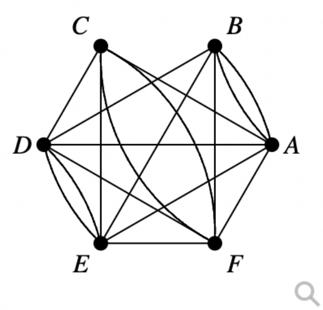
- **Civil Network Planning**
- **Computer Network Routing Protocol**
- **Cluster Analysis**

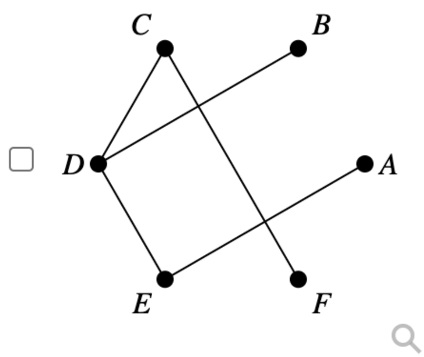
Let us understand this through a small example.

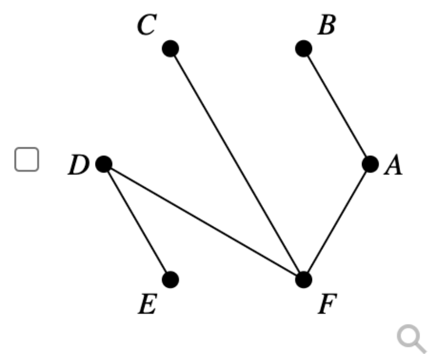
Example: Consider, city network as a huge graph and now plans to deploy telephone lines in such a way that in minimum lines we can connect to all city nodes. This is where the spanning tree comes into picture.

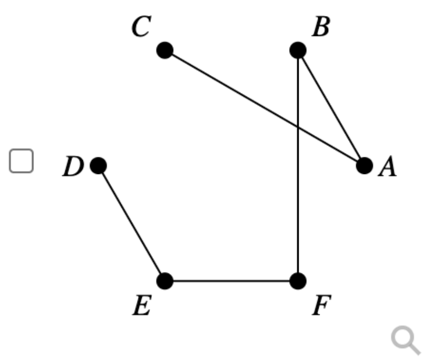
CHECK FOR UNDERSTANDING 2

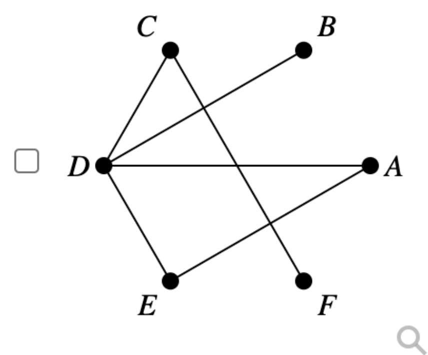
Question #1) Select the graphs that are spanning trees for the given graph.

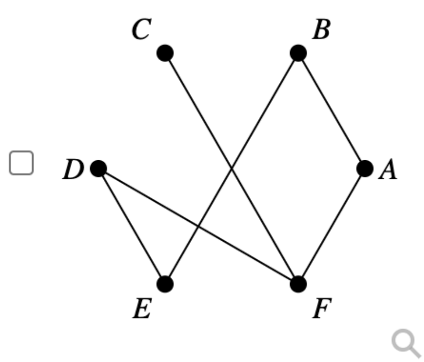


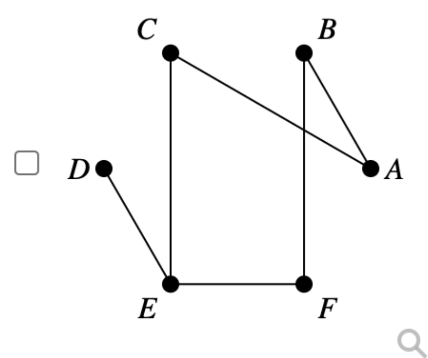




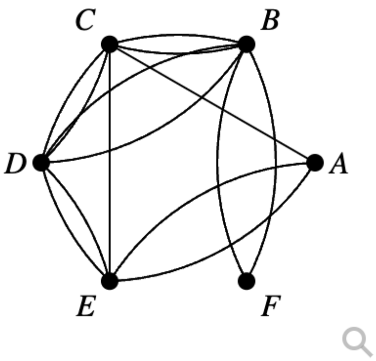


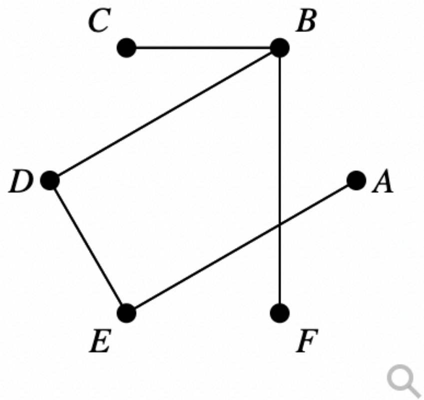


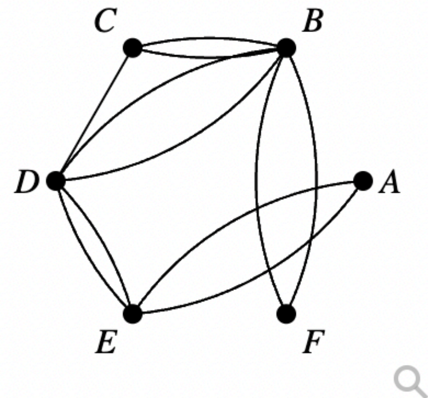


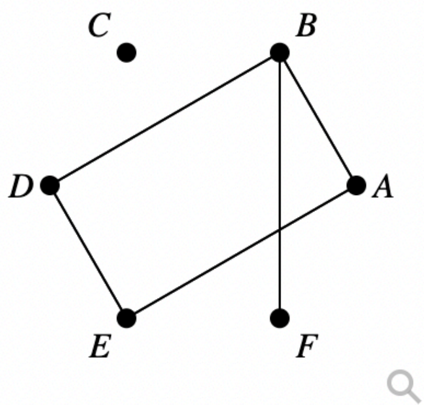


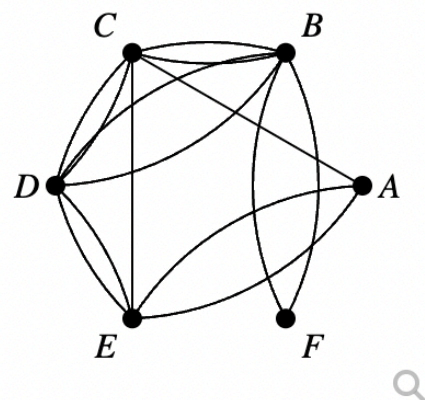
Question #2) Select the graphs that are spanning trees for the given graph.











Minimum Spanning Tree

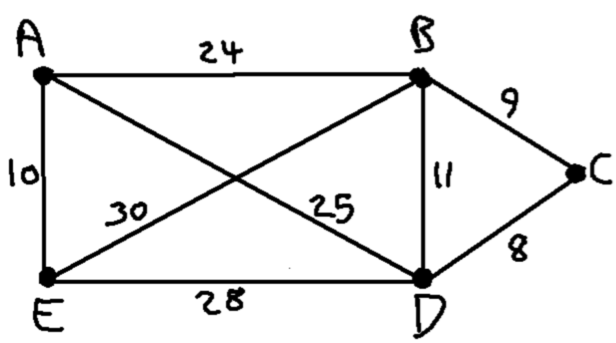
A minimum spanning tree is the tree that spans all of the vertices in a problem with the least cost (or time, or distance).

A tree that is created from a weighted graph such that the tree is of minimum possible total weight is called a/an ***minimum spanning tree.***

Key Idea. Spanning is **the amount of area or the amount of time that something encompasses**. An example of span is how long you live. An example of span is a house on three acres.

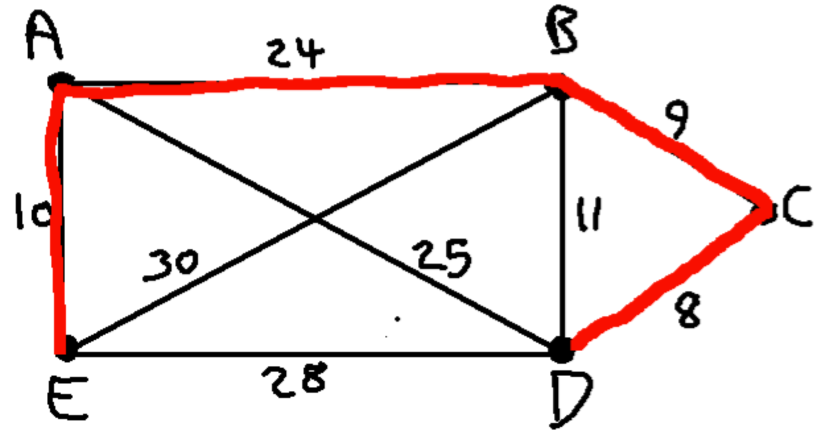
Additional Idea. In a weighted graph, a minimum spanning tree is a spanning tree that has minimum weight than all other spanning trees of the same graph. In real-world situations, this weight can be measured as distance, congestion, traffic load or any arbitrary value denoted to the edges.

Example 6.2.4: Minimum Spanning Tree



The above is a weighted graph where the numbers on each edge represent the cost of each edge. We want to find the minimum spanning tree of this graph so that we can find a network that will reach all vertices for the least total cost.

Figure 6.2.6: Minimum Spanning Tree for Weighted Graph 1



This is the minimum spanning tree for the graph with a total cost of 51.

Minimum Spanning-Tree Algorithm

We shall learn about two most important spanning tree algorithms here –

- Kruskal's Algorithm
- Prim's Algorithm

Kruskal’s Algorithm

Videos: https://www.youtube.com/watch?v=JZBQLXgSGfs

https://www.youtube.com/watch?v=Yo7sddEVONg

https://www.youtube.com/watch?v=d4BEgzK08JE&t=184s

Kruskal's algorithm to find the minimum cost spanning tree uses the greedy approach. This algorithm treats the graph as a forest and every node it has as an individual tree. A tree connects to another only and only if, it has the least cost among all available options and does not violate MST properties.

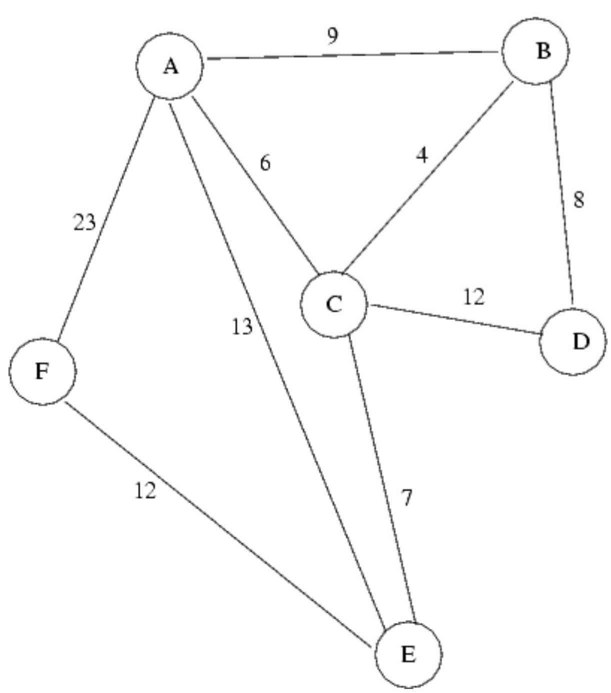
Key Idea. A procedure that produces the minimum spanning tree of a graph is called *Kruskal's* Algorithm. The algorithm specifies that one is always to choose an edge with the least possible *weight* while avoiding the creation of any *circuits*.

Kruskal's Algorithm: Since some graphs are much more complicated than the previous example, we can use Kruskal's Algorithm to always be able to find the minimum spanning tree for any graph.

1. Find the cheapest link in the graph. If there is more than one, pick one at random. Mark it in red.
2. Find the next cheapest link in the graph. If there is more than one, pick one at random. Mark it in red.
3. Continue doing this as long as the next cheapest link does not create a red circuit.
4. You are done when the red edges span every vertex of the graph without any circuits. The red edges are the MST (minimum spanning tree).

Example 6.2.5: Using Kruskal's Algorithm Figure

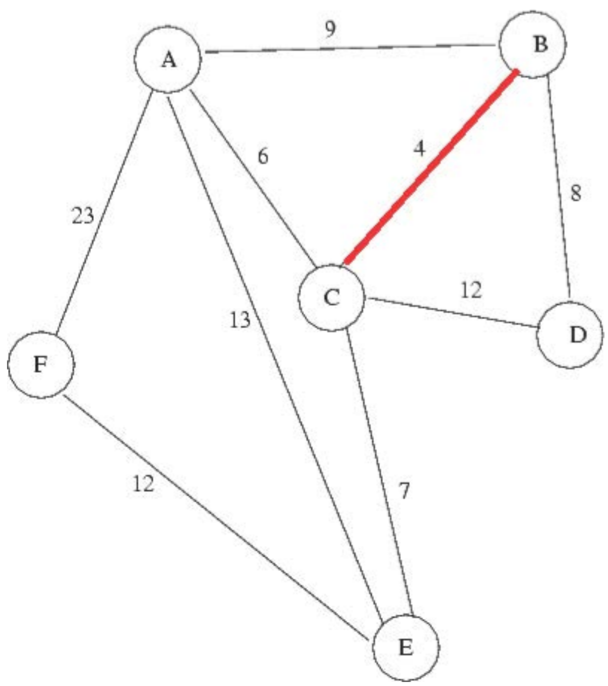
6.2.7: Weighted Graph 2



Suppose that it is desired to install a new fiber optic cable network between the six cities (A, B, C, D, E, and F) shown above for the least total cost. Also, suppose that the fiber optic cable can only be installed along the roadways shown above. The weighted graph above shows the cost (in millions of dollars) of installing the fiber optic cable along each roadway. We want to find the minimum spanning tree for this graph using Kruskal's Algorithm.

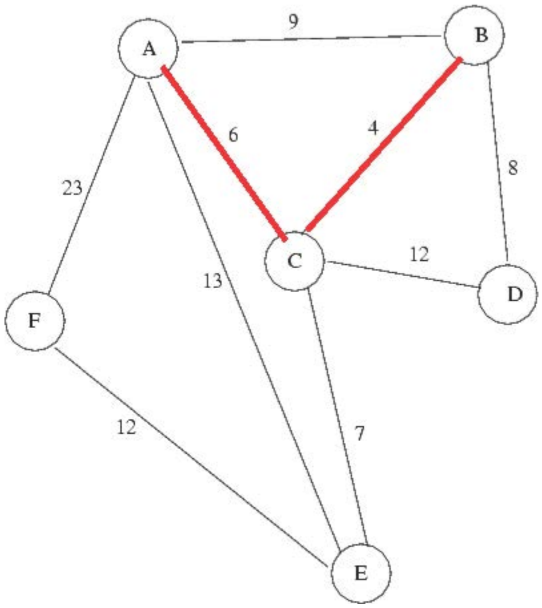
Step 1: Find the cheapest link of the whole graph and mark it in red. The cheapest link is between B and C with a cost of four million dollars.

Figure 6.2.8: Kruskal's Algorithm Step 1



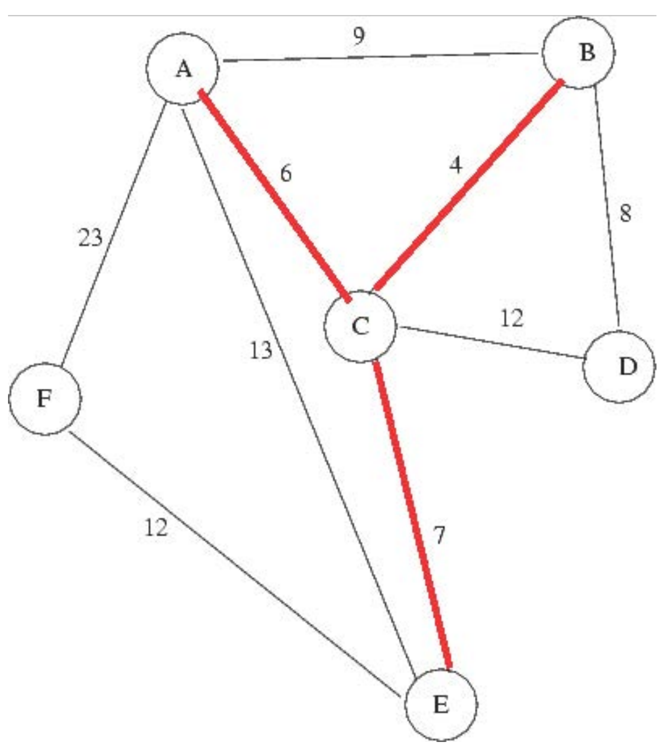
Step 2: Find the next cheapest link of the whole graph and mark it in red. The next cheapest link is between A and C with a cost of six million dollars.

Figure 6.2.9: Kruskal's Algorithm Step 2



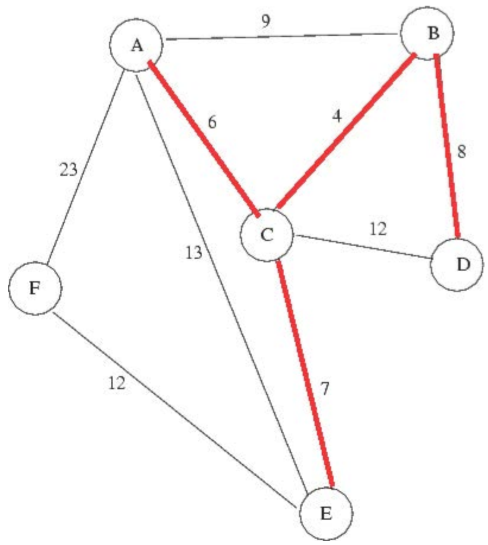
Step 3: Find the next cheapest link of the whole graph and mark it in red as long as it does not create a red circuit. The next cheapest link is between C and E with a cost of seven million dollars.

Figure 6.2.10: Kruskal's Algorithm Step 3



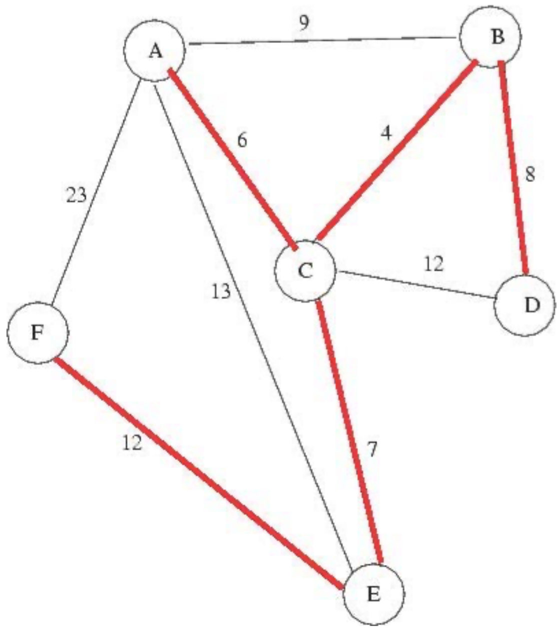
Step 4: Find the next cheapest link of the whole graph and mark it in red as long as it does not create a red circuit. The next cheapest link is between B and D with a cost of eight million dollars.

Figure 6.2.11: Kruskal's Algorithm Step 4



Step 5: Find the next cheapest link of the whole graph and mark it in red as long as it does not create a red circuit. The next cheapest link is between A and B with a cost of nine million dollars, but that would create a red circuit so we cannot use it. Therefore, the next cheapest link after that is between E and F with a cost of 12 million dollars, which we are able to use. We cannot use the link between C and D which also has a cost of 12 million dollars because it would create a red circuit.

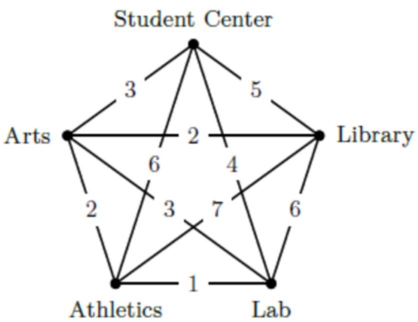
Figure 6.2.12: Kruskal's Algorithm Step 5



This was the last step, and we now have the minimum spanning tree for the weighted graph with a total cost of $37,000,000.

CHECK FOR UNDERSTANDING 3

1) A maintenance team is responsible for a group of five buildings on campus. These buildings are shown in the graph below, with the distance given between each pair of buildings. After a blizzard, the team is tasked with clearing the snow, but there is not enough time to clear all the walkways.

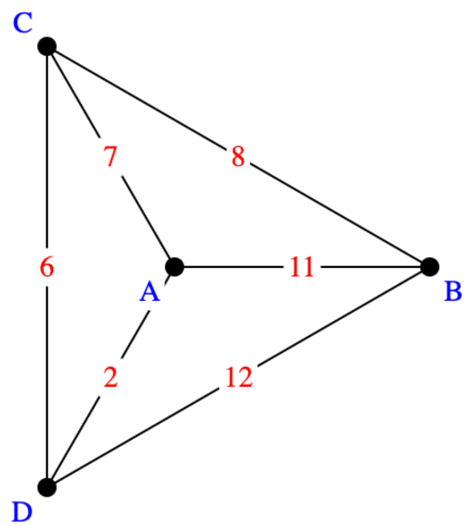


Which walkways should the maintenance team plow in order to connect all the buildings, while minimizing the time needed to do so (really, by minimizing the distance)?

- ☐ Student Center, Library
- ☐ Student Center, Lab
- ☐ Student Center, Athletics
- ☐ Student Center, Arts
- ☐ Library, Lab
- ☐ Library, Athletics
- ☐ Library, Arts
- ☐ Lab, Athletics
- ☐ Lab, Arts
- ☐ Athletics, Arts

What is the total length of walkway that will be plowed?

2) Find the minimum cost spanning tree on the graph above using Kruskal's algorithm. Which of the edges below are included in the minimum cost tree?



A) CD

B) AB

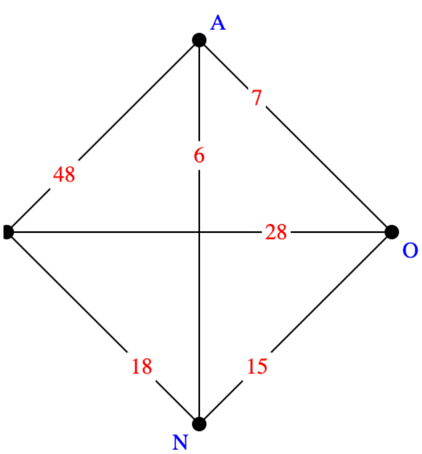
C) AD

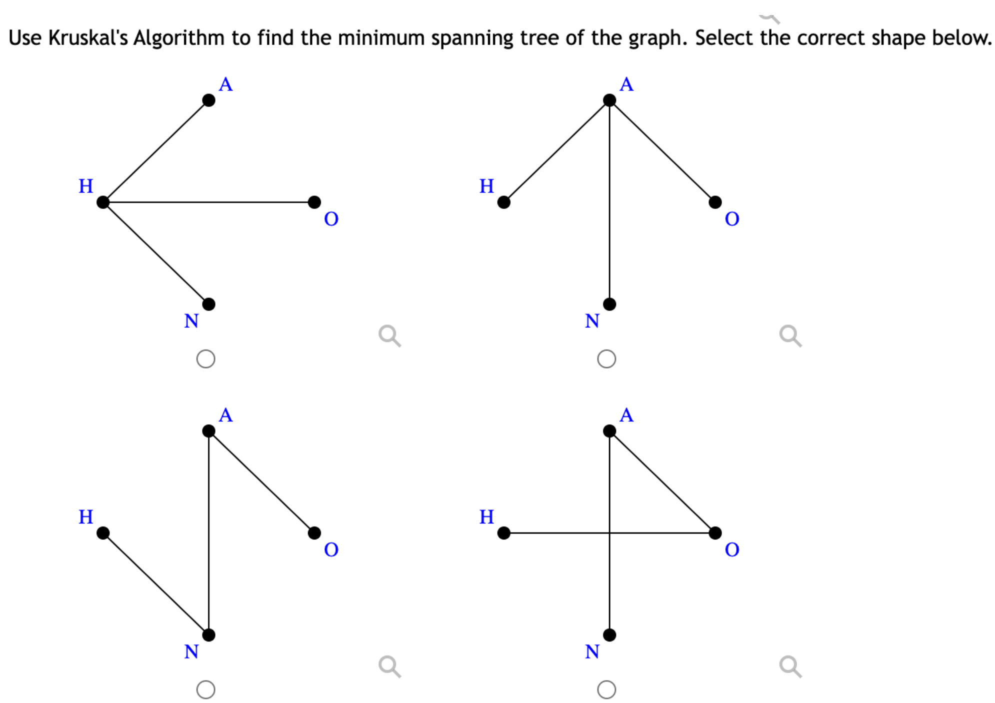
D) BD

E) AC

F) BC

3) Consider the connected, weighted graph below.





Determine the total weight of the minimum spanning tree.

The total weight of the minimum spanning tree is ________________.

4) A power company needs to lay updated distribution lines connecting eight cities in Virginia to the power grid. The distances between these cities are given in the table below. Design a network that will minimize the amount of new line.

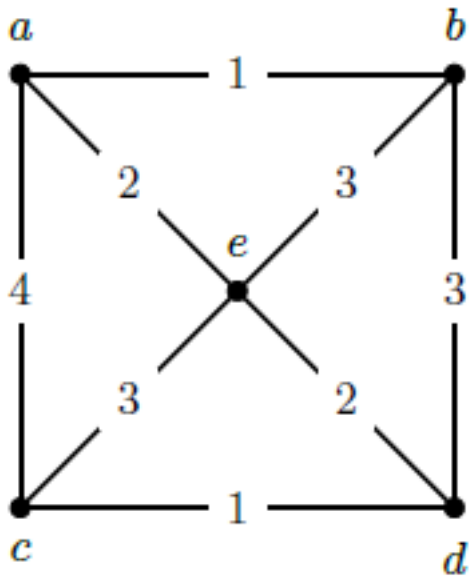
	Purcellville	Leesburg	Middleburg	Chantilly	Sterling	McLean	Arlington	Annandale
Purcellville	--	8	11	23	19	32	37	35
Leesburg	8	--	14	17	10	24	29	27
Middleburg	11	14	--	18	16	30	34	31
Chantilly	23	17	18	--	8	13	18	13
Sterling	19	10	16	8	--	15	20	17
McLean	32	24	30	13	15	--	5	7
Arlington	37	29	34	18	20	5	--	6
Annandale	35	27	31	13	17	7	6	--

What connections should be built between cities and

What is the total required length of line that must be laid?

- ☐ Purcellville, Leesburg
- ☐ Purcellville, Middleburg
- ☐ Purcellville, Chantilly
- ☐ Purcellville, Sterling
- ☐ Purcellville, McLean
- ☐ Purcellville, Arlington
- ☐ Purcellville, Annandale
- ☐ Leesburg, Middleburg
- ☐ Leesburg, Chantilly
- ☐ Leesburg, Sterling
- ☐ Leesburg, McLean
- ☐ Leesburg, Arlington
- ☐ Leesburg, Annandale
- ☐ Middleburg, Chantilly
- ☐ Middleburg, Sterling
- ☐ Middleburg, McLean
- ☐ Middleburg, Arlington
- ☐ Middleburg, Annandale
- ☐ Chantilly, Sterling
- ☐ Chantilly, McLean
- ☐ Chantilly, Arlington
- ☐ Chantilly, Annandale
- ☐ Sterling, McLean
- ☐ Sterling, Arlington
- ☐ Sterling, Annandale
- ☐ McLean, Arlington
- ☐ McLean, Annandale
- ☐ Arlington, Annandale

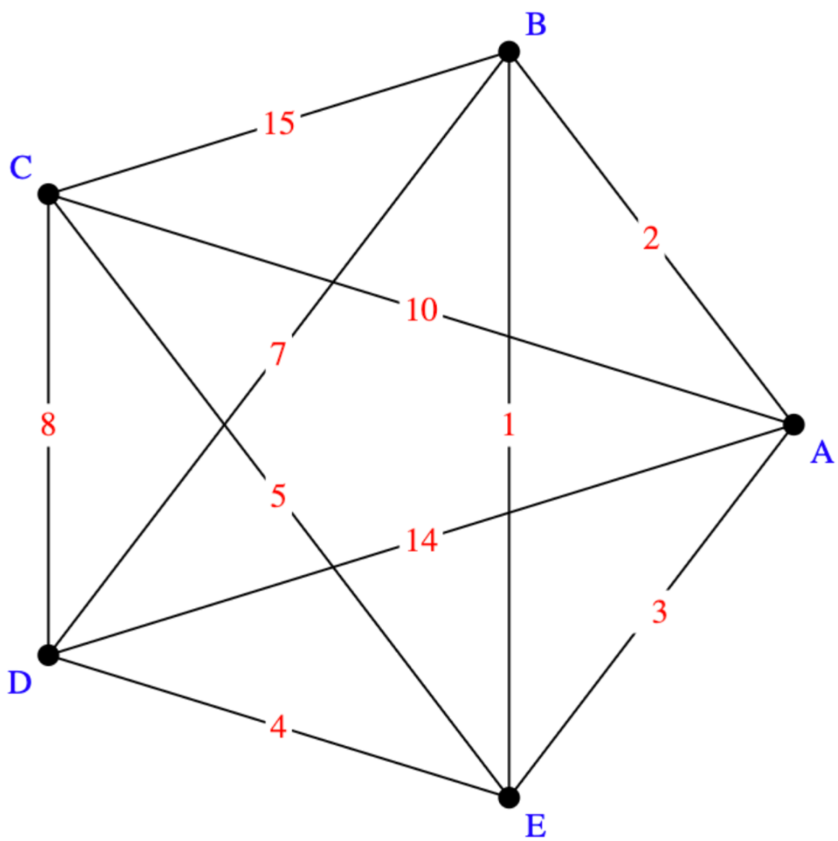
5) Use Kruskal's algorithm to find a minimum spanning tree for the graph below.



Select the edges that belong to this tree.

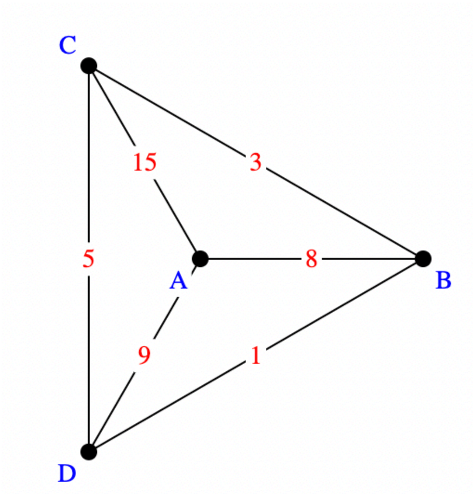
E) AB
F) AC
G) AE
H) BD
I) BE
J) CD
K) CE
L) DE

6) Use **Kruskal's Algorithm** to find a **Minimum Spanning Tree** for the graph below.



The weight of the minimum spanning tree is: _____

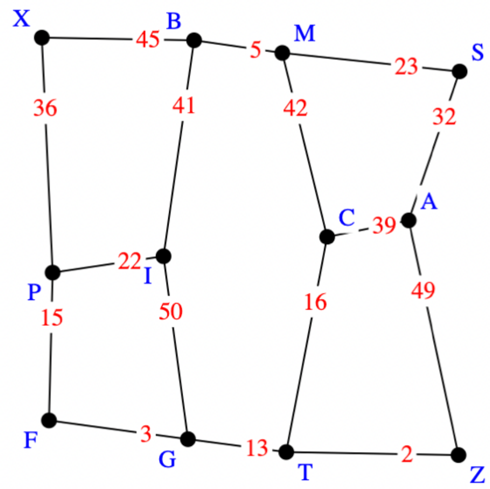
7)



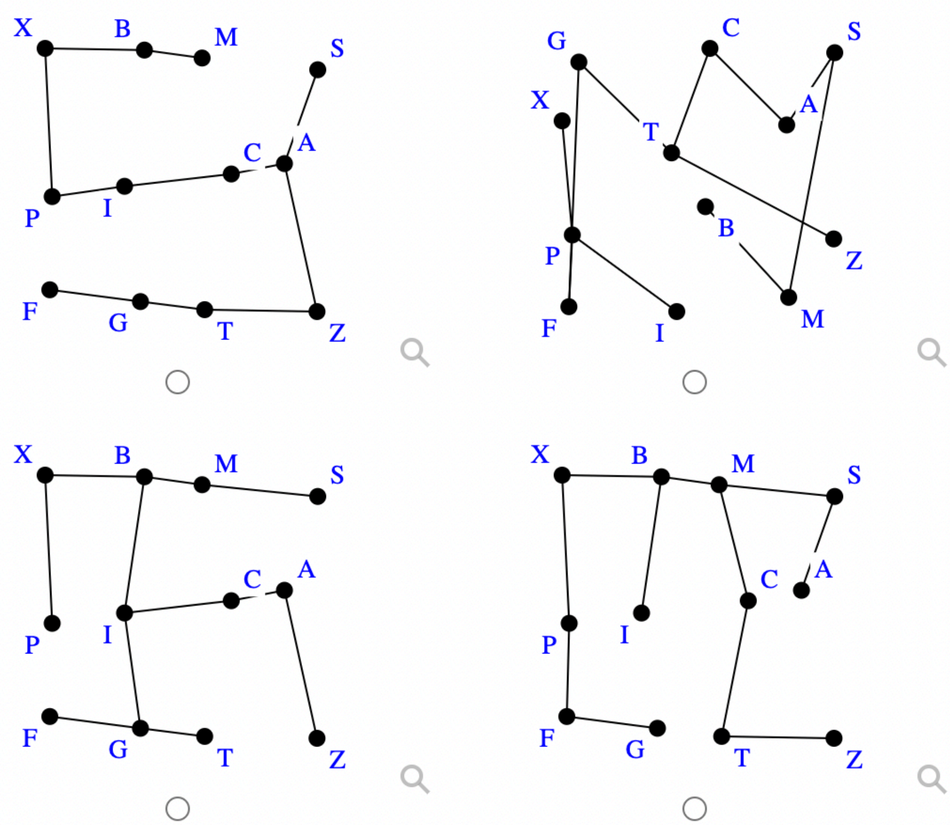
Find the minimum cost spanning tree on the graph above using Kruskal's algorithm. Which of the edges below are included in the minimum cost tree?

- ☐ BD
- ☐ BC
- ☐ AD
- ☐ CD
- ☐ AC
- ☐ AB

8) Consider the connected, weighted graph below.



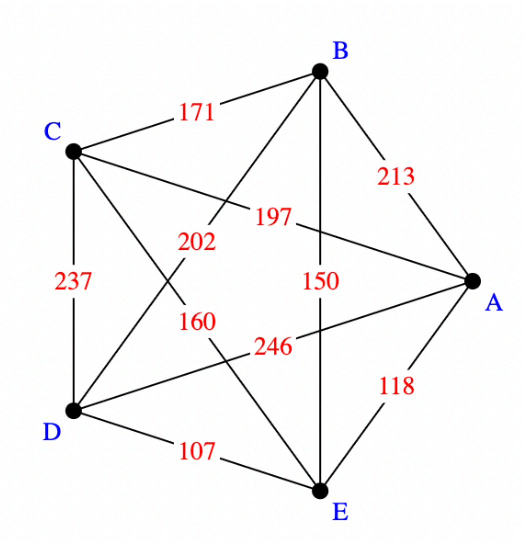
Use Kruskal's Algorithm to find the minimum spanning tree of the graph. Select the correct shape below.



Determine the total weight of the minimum spanning tree.

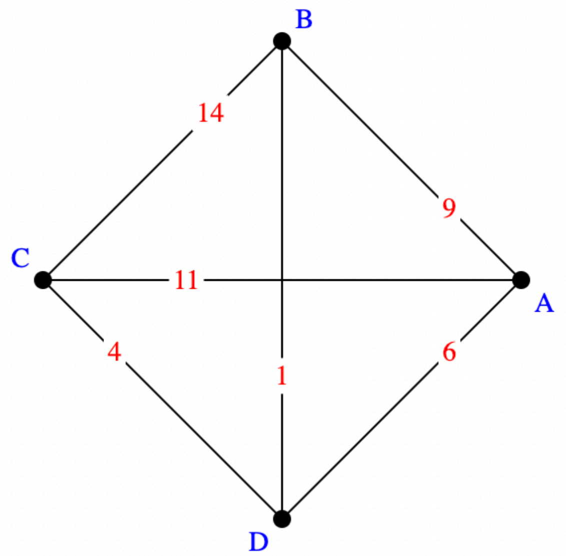
The total weight of the minimum spanning tree is ______.

9) A city planner wants to work with community leaders to develop a system of walker-friendly paths between the city's five parks. Since cost is an issue, the planner collects data on the cost of each park-to-park route. The planner uses the Nearest Neighbor method to determine an estimate of a least cost system of sidewalks that would allow a walker to visit each park once and return to the park where the walker has parked his/her car. The weighted graph below shows the cost of each possible path in thousands of dollars.



Starting and ending at vertex ***A***, what is the minimum cost to construct paths as estimated by the Nearest-Neighbor method? $ ___________.

10)



Find the minimum cost spanning tree on the graph above using Kruskal's algorithm. What is the **total cost** of the tree?

Sources
https://www.geeksforgeeks.org/graph-measurements-length-distance-diameter-eccentricity-radius-center/

https://www.gatevidyalay.com/tag/cycle-graph-definition/

Thank you for listening!
Do the assigned activities!