STUDENT NAME.

ROLL NO.

SUBJECT: DISCRETE MATHEMATICS

SEM 4

BRANCH: CE

UNIT 4 : GRAPH AND TREE

1. APLLICATION OF GRAPH THEORY

- Computer Science – Graph theory is used for the study of algorithms. For example,
  - Kruskal's Algorithm
  - Prim's Algorithm
  - Dijkstra's Algorithm
- Computer Network – The relationships among interconnected computers in the network follows the principles of graph theory.
- Science – The molecular structure and chemical structure of a substance, the DNA structure of an organism, etc., are represented by graphs .
- Electrical Engineering – The concepts of graph theory is used in designing circuit connections.
- General – Routes between the cities, different places, can be represented using graphs

2. HISTORY :

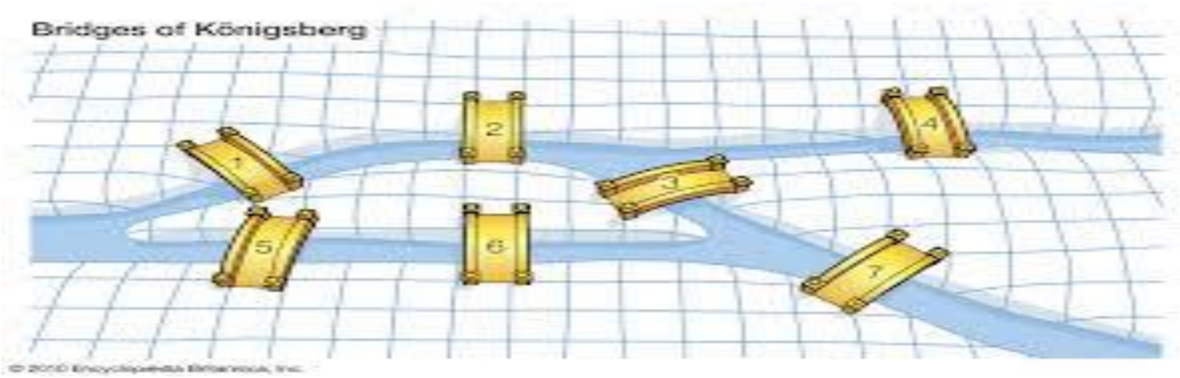
Leonhard Euler

The mathematician, astronomer, engineer (just a *few* of his titles),

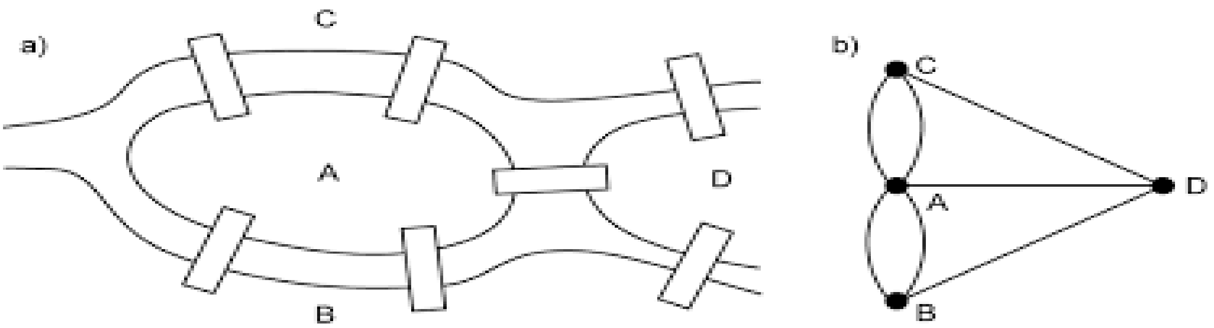
Who made significant contributions to calculus and the person who actually created graph theory that we depend on every single day.

Taking a walk through Konigsberg………

There were two large islands in the middle of the Pregel River, and they were each connected to one another, as well as to the two riverbanks on either side, which comprised the majority of the city. And how were they connected? By bridges, of course! Seven of them, in fact



The goal was to walk across all of the seven bridges crossing the islands only once, without ever repeating a single bridge in the course of one's walk.



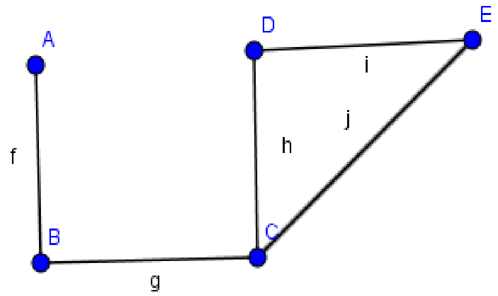
Euler was hooked. Euler was so entranced, in fact, that he ended up writing a paper later that year that would contain a solution to the bridge problem.

But before we understand how Euler solved this problem, we need to cover a few basic graph theory rules first...

3. DEFINITIONS :

1. GRAPH :

   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.

   In this graph

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

   E={AB,BC,CD,DE,CE}

   = {f , g , h , I , j}

   EXAMPLES

   1] 2]

2. NULL GRAPH

   A graph having no edges is called null graph

   Example:

3. TRIVIAL GRAPH
   A graph with only one vertex is called trivial graph.
   Example:

4. LOOP
   In a graph, if an edge is drawn from vertex to itself, it is called a loop.
   Here A is a vertex for which it has an edge (A, A) forming a loop.
   Example:

5. PARALLEL EDGES
   In graph, if a pair of vertices is connected by more than one edge, then those edges are called parallel edges.
   Here edges g and d are parallel edges
   A graph having parallel edges is known as a Multigraph.
   Example:

6. SIMPLE GRAPH
   A graph G is called simple graph if G does not have any loop and parallel edges
   Example:

7. ADJACENT VERTEX
   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.
8. ADJACENT EDGE
   In a graph, two nonparallel 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:

9. DEGREE OF VERTEX
   It is the number of edges incident with the vertex V.
   Notation: deg(V).
   Example:

10. PENDENT VERTEX
    A vertex with degree one is called a pendent vertex.

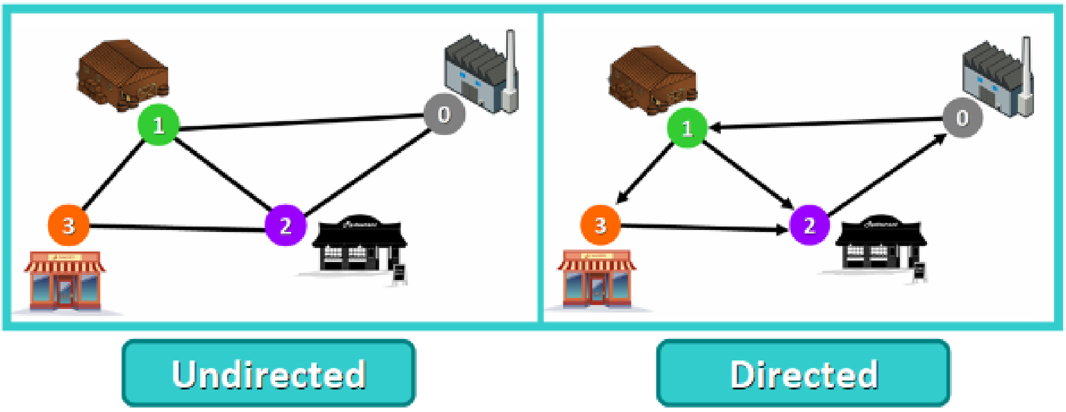
11. ISOLATED VERTEX
    A vertex with degree zero is called an isolated vertex.
    Example:

12. REGULAR GRAPH
    If degree of all vertex of graph is same then it is called regular graph
    Example:

13. DIRECTED GRAPH
    A graph is a directed graph, where all edges are directed from one vertex to another vertex

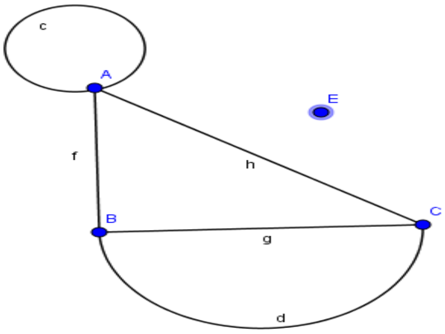
14. UNDIRECTED GRAPH
    A graph is an undirected graph ,where all edges are bidirectional

Example:

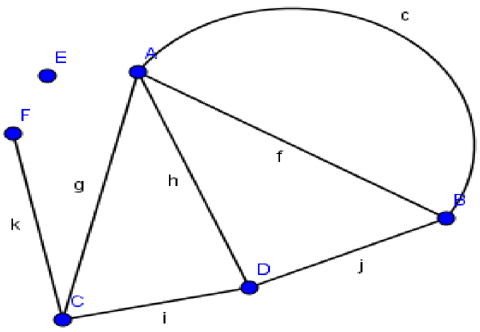


EXERCISE 1:

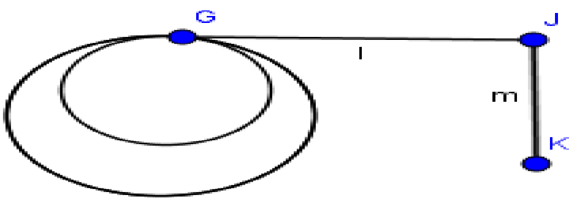
1. FIND THE DEGREE OF EACH VERTEX OF FOLLOWING GRAPHS



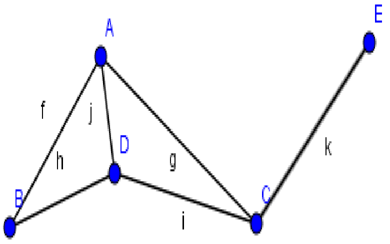
(1) ANS:



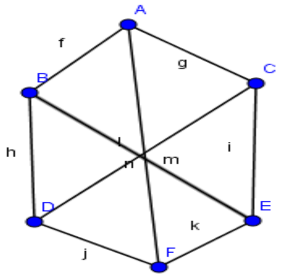
(2) ANS:



(3) ANS:



(4) ANS:



(5) ANS :

2. Draw a graph with five vertices {A,B,C,D,E} with deg(A)=2, deg(B)=3, deg(C)=2, deg(D)=2, deg(E)=1

3. Draw a graph with 4 four vertices and all vertices have degree five

4. Draw a graph with six vertices and degree of all vertices are five

5. Draw a connected graph with 5 vertices and 7 edges

Theorem 1: In a graph G, the sum of the degree of the all vertices of G is equal to the twice the number of edges of G.

In other words: If G is a graph with n vertices and e edges then

$\deg(V_1) + \deg(V_2) + \ldots\ldots\ldots + \deg(V_n) = 2e$ **[First theorem of Graph theory]**

Proof.

Let f be any edge of graph G.

Case-1

If f is a loop incident on vertex V_1 then f is count twice when we count degree of vertex V1.

Case-2

If f is incident on V_1 *and* V_2 then f is count in both deg (V_1) and deg (V_2). So f is count twice if we add deg (V_1) and deg (V_2)

So that adding the degree of all vertices involves counting twice of each edge of G

There for addition of degree of all vertexes is equal to twice of number of edges.

<u>Note : The sum of degree of all vertices in graph G is always even number</u>

Example:

<u>Theorem 2:</u> The number of vertices of odd degree in a graph is always even in quantity

Proof:

By the first theorem if G is a graph with n vertices and e edges then

$$\sum_{i=1}^{n} deg(Vi) = 2 * e \ = even\, number$$

---------------------(1)

The quantity in the left side of above equation can be expressed as the

sum of even degree vertices and odd degree vertices as follow

$$\sum_{i=1}^{n} deg(Vi) = \sum_{even} deg(Vi) + \sum_{odd} deg(Vi)$$

Since the left hand side of the above equation is even by (1) and the first

expression on the right hand side is also even as sum of even numbers is

even. so second expression must be even

$\sum_{odd} deg(Vi)$ = even number

In equation above equation *deg(Vi*) is odd so that the total number of the terms in the sum must be even to make a sum an even number.

Therefore the number of vertices of odd degree in a graph is always even.

Example :

<u>Theorem 3:</u> Show that the maximum number of edges in a simple graph with n vertices is $\frac{n(n-1)}{2}$

Proof:

Let G is a simple Graph with n vertices.

Since G is a simple Graph first vertex can be adjacent with maximum (n – 1) vertices. So on first vertex n – 1 edges incident.

Now first vertex adjacent with second vertex so second vertex can be adjacent with maximum (n – 2) vertices. So on second vertex maximum (n – 2) new edges incident.

Continuing....

On (n-1)th vertex maximum 1 new edges incident.

So that the maximum number of edges in G

= (n-1) +(n-2) +(n-3) + ...+1 + 0

= (n-1) +(n-2) +(n-3) + ...+1 + n - n

= 1+ 2+3+……+ (n-3)+(n-2)+(n-1) +n - n

= $\frac{n(n+1)}{2} - n$

= $\frac{n(n-1)}{2}$

Examples:

- **COMPLETE GRAPH**

  The simple graph is said to be complete graph if there exist an edge between each and every pair of vertices.

  A complete graph of n vertices is denoted by K_n.

  Example

- **BIPERTITE GRAPH**

A graph *G (V,E)* is called a bipartite graph if its vertex set *V* can be partitioned into two nonempty subsets V_1 and V_2 such that each edge of G has one end point in V_1 and the other end point in V_2. The partition $V = V_1 \cup V_2$ is called bipartition of G.

Example:

- **COMPLETE BIPERTITE 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.

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

Examples:

- **SUBGRAPH**

A graph $H=(V_1,E_1)$ is called sub graph of graph $G = (V, E)$ if the vertices and edges of H are contained in the vertices and edges of G.

i.e. V_1 is subset of V and E_1 is subset of E

Example:

EXERCISE:

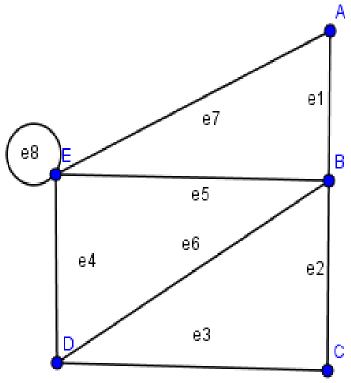
1. A graph G has 15 edges, 3 vertices of degree 4 and other vertices of degree 3. Find the number of vertices in G.

2. Find the number of edges in graph G, if it has 5 vertices each of degree 4.

3. Find the number of edges in graph G, if it has 3 vertices of degree 3, 2 vertices of degree 2 and 1 vertex of degree 5.

4. Find the number of edges in graph K_6 and $K_{2,3}$

- **WALK**
- A Walk is defined as a finite alternating sequence of vertices and edges beginning and ending with vertex, No edge appears more than once in walk, vertex, however, may appear more than once.
- The length of the walk is the number of edges in it.
- A walk is closed if the first vertex and last vertex are same.

- 

in this graph .

(1) A e1 B e2 C e3 D e4 E

(2) E e8 E e4 D e6 B e1 A are Walk

but

(3) A e1 B e5 E e8 E e7 A e1 B e2 C

is not a walk. (Because edge e1 is repeated)

Here Closed walk is A e1 B e2 C e3 D e4 E e7 A

- **PATH**
- An Open walk in which no vertex appears more than Once is called Path.
- The number of edges in path is called length of path
- **We can be including self-loop in walk but not in path.**
- In the figure
- (1) A e1 B e2 C e3 D e4 E is a path
  But
- (2) E e8 E e4 D e6 B e1 A
- (3)A e1 B e2 C e3 D e6 B are not path.

- **CIRCUIT**
- A Closed walk in which no vertex (except initial and final vertex) appears more than once is called circuit (cycle).
  Here
- A e1 B e5 E e7 A
- B e6 D e4 E e5 B are circuit.
- A circuit is also called a cycle

- **Clearly , every vertex in a circuit is of degree two.**

- **Connected Graph & Disconnected Graph**

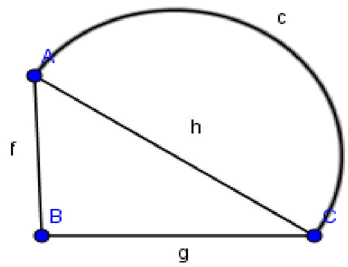
  A Graph G is said to be connected is there is at least one path between every pair of vertices in G . Otherwise it is disconnected.

  Example

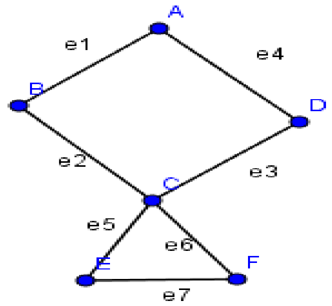
- **Cyclic graph.**
  In graph theory, a **cycle graph** or **circular graph** is a graph that consists of a single cycle, or in other words, some number of vertices (at least 3, if the graph is simple) connected in a closed chain. The cycle graph with *n* vertices is called C_n.
  Number of vertices in C_n equals the number of edges, and every vertex has degree 2; that is, every vertex has exactly two edges incident with it.
  Example.

Exercise:

1. Evaluate the at least three walks, paths and circuit (closed and open) for the following graph .
   1] ANS.

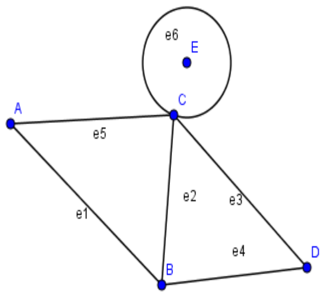


2] **ANS**



.

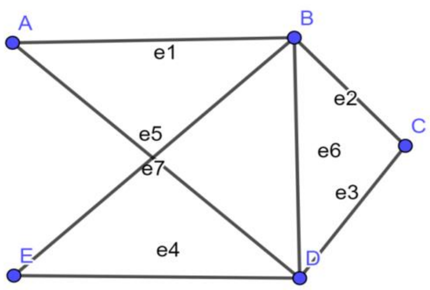
3] **ANS.**



- **Euler Graph**
- In a graph G it is possible to find a closed walk running through every edge of G exactly once , then such walk is called Euler line .
- A graph that consists of Euler line is called an Euler Graph .

  EXAMPLE

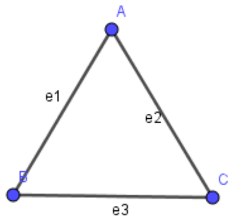
  1]



Here

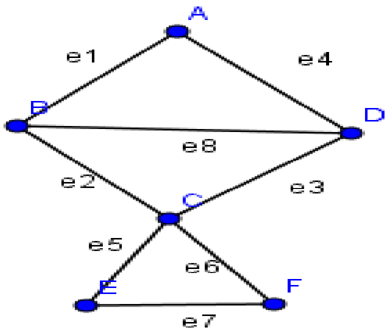
Closed walk is A e1 B e2 C e3 D e4 E e5 B e6 D e7 A is Euler line so this graph is Euler graph.

2]



It is Euler graph because we get Euler line is A e1 B e3 C e2 A

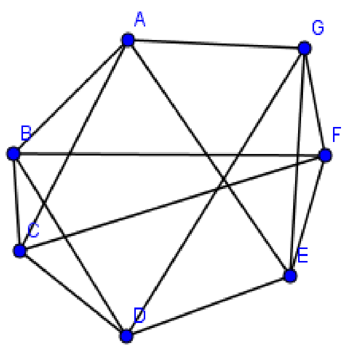
3]



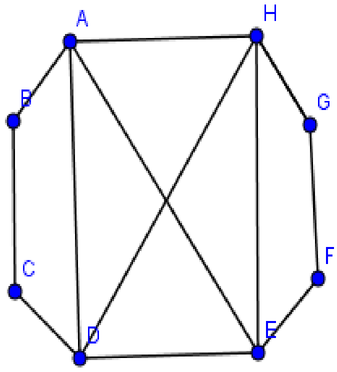
Is it Euler graph?
Justify ..

Note: **A given connected graph G is an Euler graph iff all Vertices of G are of even degree.**

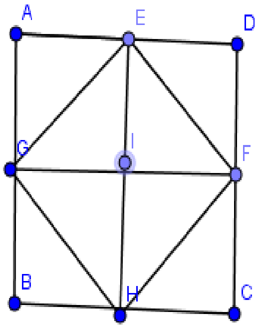
Exercise: Determine which of the following graph Euler graph is and write Euler line.
Give the reason if it is not Euler graph
1]



2]

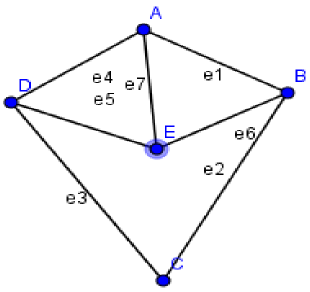


3]



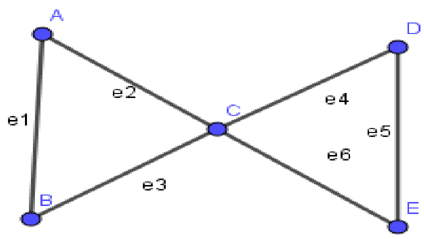
➢ **Hamiltonian Graph**

- A circuit in a connected graph G is said to be Hamiltonian circuit if it include every vertex of G.
- This types of graph is known as Hamiltonian Graph
- In following graph the circuit



- **A e1 B e2 C e3 D e5 E e7 A is Hamiltonian circuit so it is Hamiltonian graph.**

Example

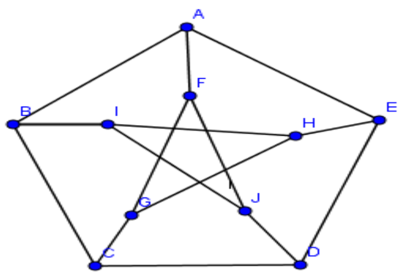


Is it Hamiltonian graph?
Justify

- If we remove any edge from a Hamiltonian circuit, then it called a Hamiltonian path.
- The length of Hamiltonian path (if it exists) in a connected graph of n vertices is n-1.
- It is easy to construct a Hamiltonian circuit in complete graph.

Example.

- Petersen Graph

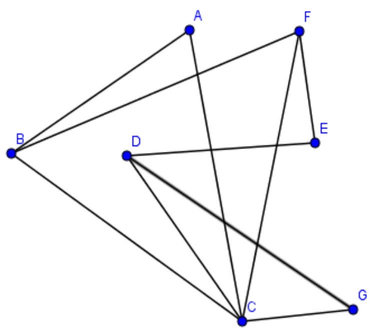


Is it Hamiltonian graph?
Is it Euler graph?
Justify.

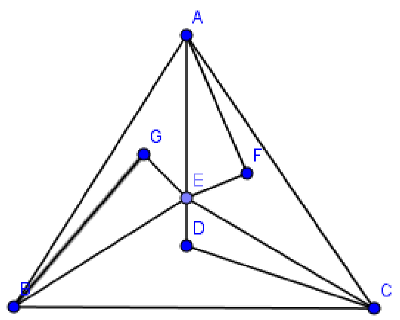
Exercise. Determine which of the following graph Hamiltonian graph is and write Hamiltonian circuit.

Give the reason if it is not Hamiltonian graph

1]

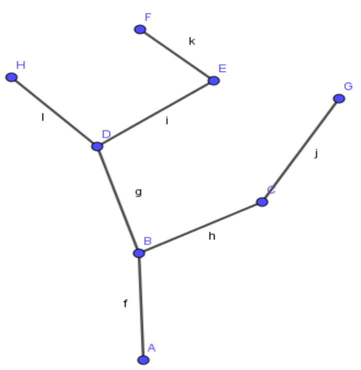


2]



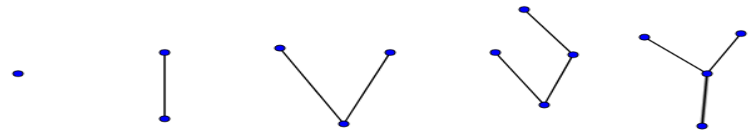
- **Tree**
- A tree is connected graph without any circuits.
- In other words a connected graph with no cycles.

The edges of the tree are known as branches and Elements of the tree is known as nodes.



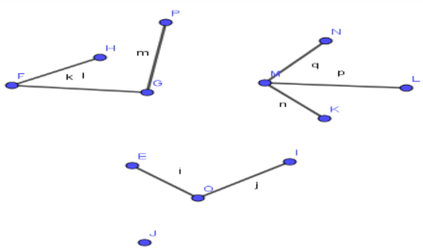
- Tree is a simple graph.
- i.e. having neither a self-loop nor parallel edges.
- For tree we should have at least one vertex.

- A vertex of degree one in a tree is called leaf.
- A tree with one ,two ,three ,four vertices



- Forest

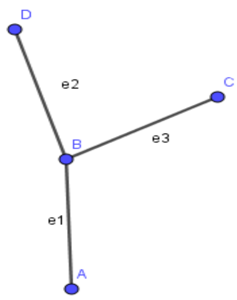
A disjoint collection of trees is known as forest .

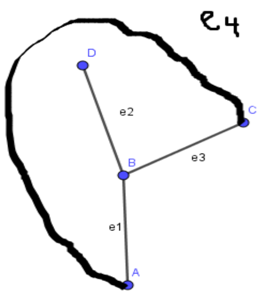


Theorem: There is one and only one path between every pair of vertices in a tree T.

Proof:

- Since T is connected graph ,there must exist at least one path between every pair of vertices in T ,
- Now suppose that between two vertices A and C of T there are two distinct paths .





Path 1: A e1 B e3 C

Path 2: A e4 C

The union of these paths will contain a circuit and T cannot be Tree.

U: A e1 B e3 C e4 A

Thus it is contradict with definition of tree

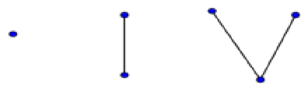
Therefor there is one and only one path between every pair of vertices in a tree T.

Theorem: A tree with n vertices has n-1 edges.

Proof:

By the induction rule

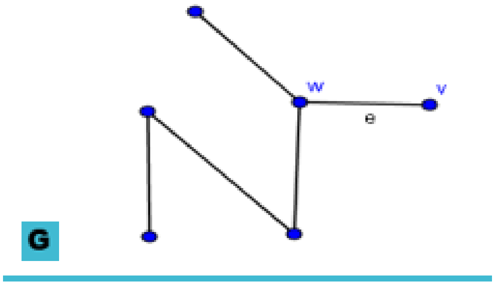
It is easy to see that the theorem is true for n =1, 2 and 3.



Assume that the theorem holds for all trees with n-1 vertices.

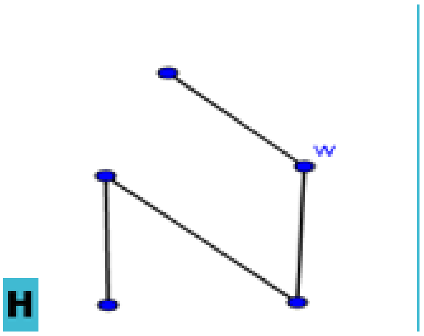
(I.e. A tree with n-1 vertices has n-2 edges)

Let G be tree with n vertices.



Pick up any leaf v and let e = {v, w} be a unique edge.

Now remove v and e form the graph G and get graph H



Now consider the graph H

It connected graph.

It has no circuits (cycles)

So H is tree with n-1 vertices.

And it has n-2 edges (Because of Assumption)

Therefore G has (n-2) + 1 = n-1 edges.

Hence it is proved

- **Binary Tree**

  A binary tree is defined as a tree in which there is exactly one vertex of degree two, and each of the remaining vertices is of degree one or three.

  Example

A vertex whose degree 2 is called root. A non-pendant vertex in a tree is called an internal vertex.

Properties

- The number of vertices in a binary tree is always odd, because there is exactly one vertex of even degree, and the remaining n-1 vertex are odd degree.
- Let p be the number of pendant vertices in a binary tree T. then n – p - 1 is the number of vertices of degree three.

Verify it in above example

Exercise.

Draw binary tree with 7 and 11 vertices.

- **Spanning Tree**

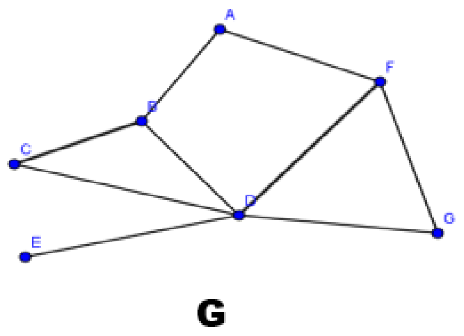
A tree T is said to be spanning tree of a connected graph G if T is a sub graph of G and T contains all vertices of G.

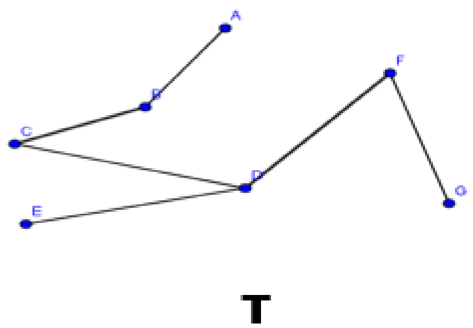
It is noted that spanning tree is defined only for a connected graph because a tree is always connected graph,

We cannot get spanning tree for disconnected graph

A graph may have many spanning trees.

Example.





Circuit Rank

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

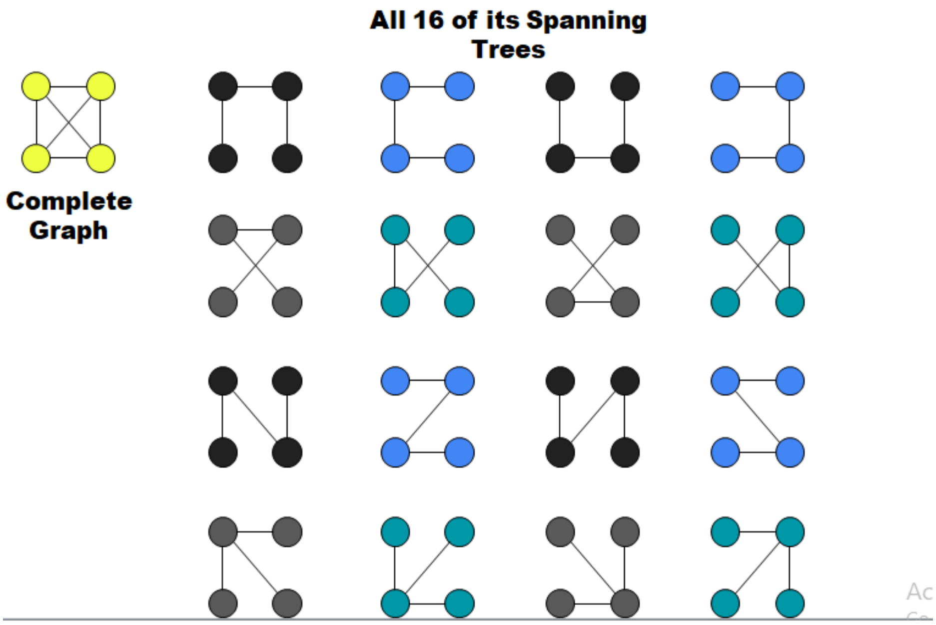
Therefore, the number of edges you need to delete from 'G' in order to get a spanning tree = m-n+1

This is known as Circuit Rank of Graph G.

Check in the above example.

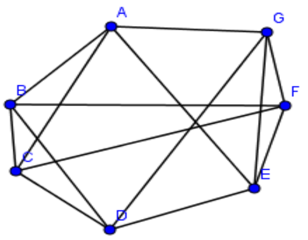
Spanning tree of a graph is just a sub graph that contains all the vertices and is a tree.

A graph may have many spanning trees.



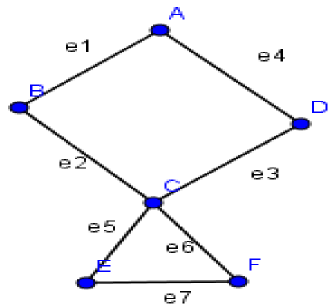
Exercise.

Check circuit rank and draw spanning tree of following graph.

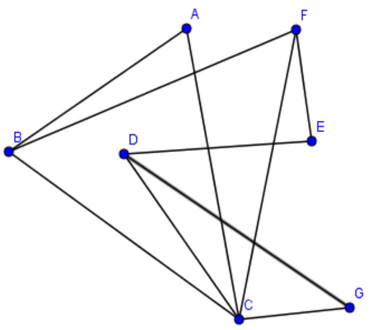


1]

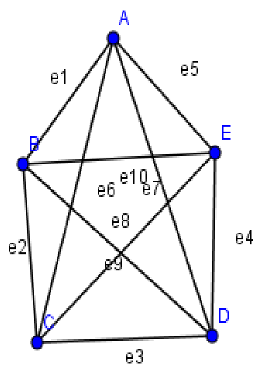
2]



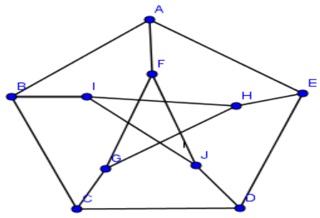
3]



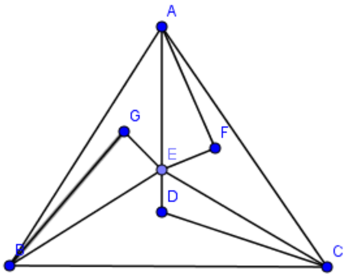
4]



5]



6]

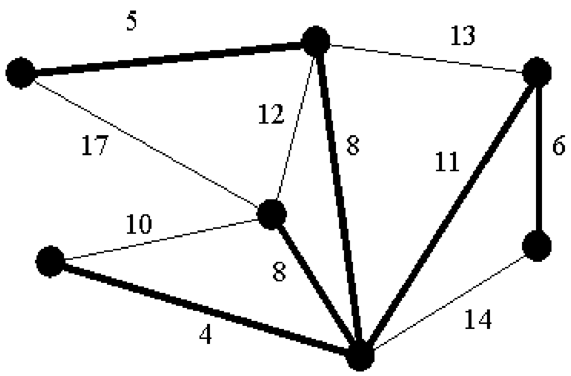


- **Minimum (weighted) Spanning tree:**

A tree T is said to be A minimum weighted spanning tree of a graph G if T is a sub graph of connected graph G with connects all vertices together, without any cycles and minimum possible total edge weight.

Thus it is spanning tree whose sum of edge weights is as small as possible. Then T is known as minimum spanning tree.

Example.



- **Kruskal's algorithm to minimum weighted spanning tree.**

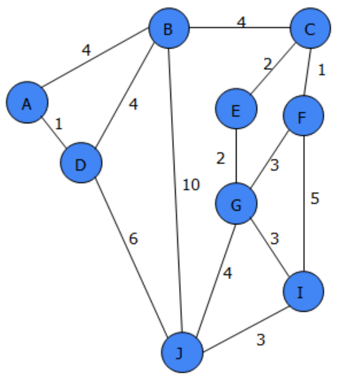
Step 1: Make list of weighted edges of graph.

Step 2: Sort edges as per increasing order of weight.

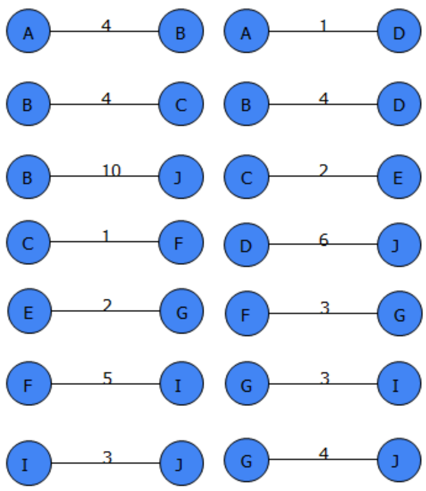
Step 3: Add edges in graph to make minimum weighted spanning tree.

Step 4: Derive total of edges weight of spanning tree.

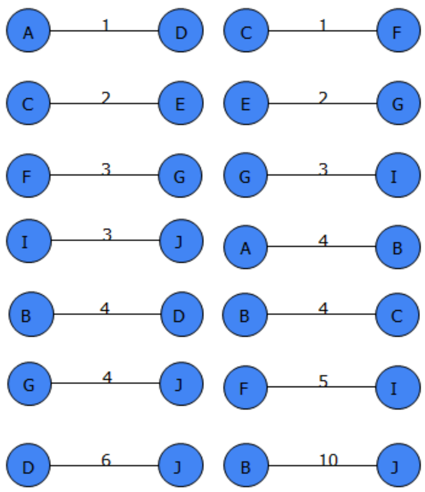
Example: Find minimum weighted spanning tree by Kruskal's Algorithm.



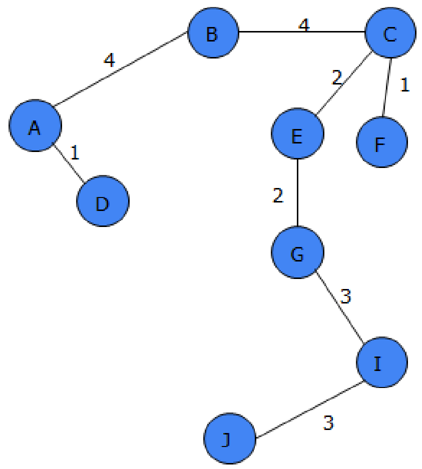
Step 1: List of edges



Step 2: Sort edges.



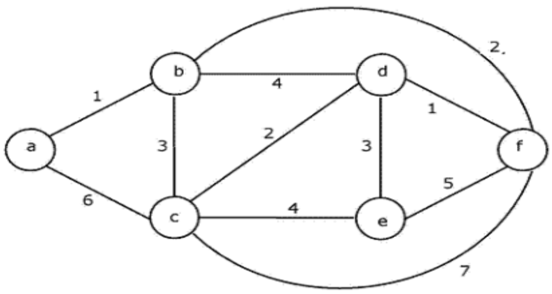
Step 3:



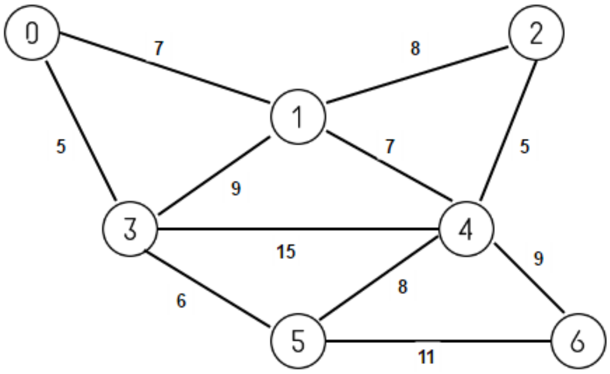
Step 4: Total edges weight = 1+1+2+2+3+3+4+4= 20

Exercise. Apply Kruskal's Algorithm to find the minimum weighted spanning tree.

1]



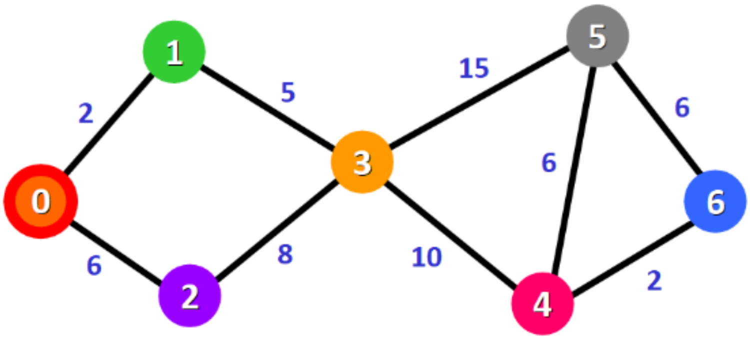
2]



- **Dijkstra's Algorithm**

- Dijkstra's Algorithm basically starts at the node that you choose (the source node) and it analyses the graph to find the shortest path between that node and all the other nodes in the graph.
- The algorithm keeps track of the currently known shortest distance from each node to the source node and it updates these values if it finds a shorter path.
- Once the algorithm has found the shortest path between the source node and another node, that node is marked as "visited" and added to the path.
- The process continues until all the nodes in the graph have been added to the path. This way, we have a path that connects the source node to all other nodes following the shortest path possible to reach each node.

Example. Find the shortest path of following graph



Unvisited Nodes= {0,1,2,3,4,5,6}

Distance:

0: 0

1: ∞

2: ∞

3: ∞

4: ∞

5: ∞

6: ∞

Start to visit other vertex from 0 vertex as per shortest path.

Distance:

0: 0

1: 2

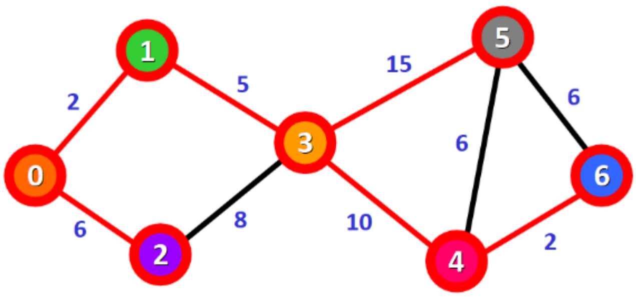
2: 6

3: 7

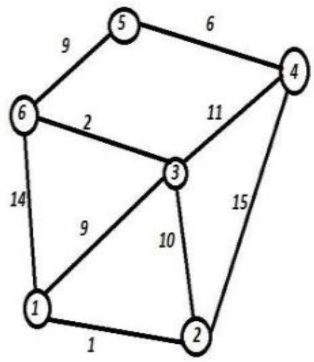
4: 17

5: 22

6: 19



Example: Apply Djakarta's algorithm to find shortest path from first vertex



- **Prim's Algorithm**

  Prim's algorithm is a minimum spanning tree algorithm that takes a graph as input and finds the subset of the edges of that graph which form a tree that includes every vertex has the minimum sum of weights among all the trees that can be formed from the graph

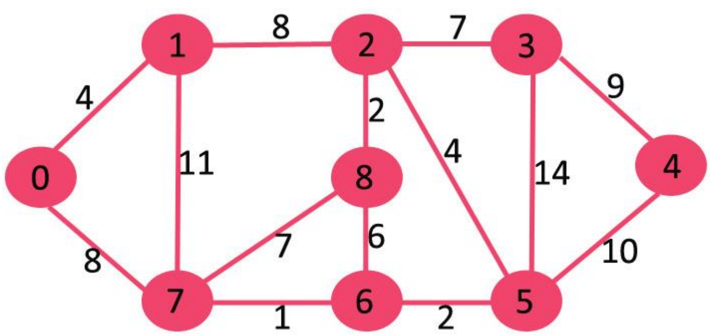
How Prim's algorithm works

We start from one vertex and keep adding edges with the lowest weight until we reach our goal.

The steps for implementing Prim's algorithm are as follows:

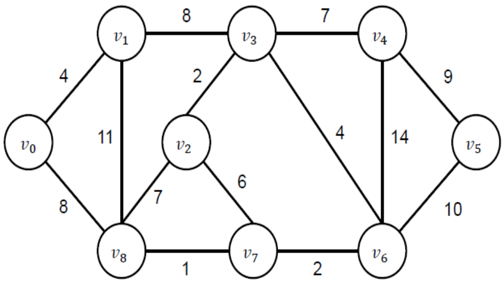
- Initialize the minimum spanning tree with a vertex chosen at random.
- Find all the edges that connect the tree to new vertices, find the minimum and add it to the tree
- Keep repeating step 2 until we get a minimum spanning tree

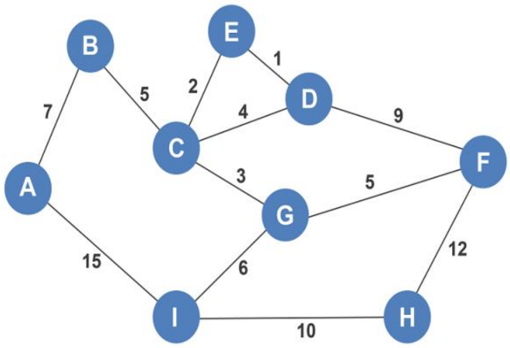
Example: Apply prim's algorithm to find the MST.



Solution.

Example.


v_1
v_3
v_4
v_0
v_2
v_5
v_8
v_7
v_6
8
7
4
2
9
11
4
14
6
7
8
10
1
2


B
E
D
C
F
A
G
I
H
7
5
2
1
4
9
3
5
15
6
12
10