IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 18, Issue 1 Ser. I (Jan. – Feb. 2022), PP 08-27
www.iosrjournals.org

Matrix Representation of Graph Theory with Different Operations

Md. Ashraful Islam[1], Omar Faruk[2*] , Suman Kar[3], Mohammad Shahidul Islam[4]
[1](*Department of Computer Science and Engineering, Dhaka International University, Dhaka-1205, Bangladesh)*
[2](*Department of Civil Engineering, Dhaka International University, Dhaka-1212, Bangladesh)*
[3](*Basic Science Division, World University of Bangladesh, Dhaka-1205, Bangladesh)*
[4](*Department of Mathematics, Tejgaon College, Dhaka-1215, Bangladesh)*

Abstract: *Graph theory is one of the most important and basic topics of discrete mathematics in Mathematics. In all sectors of science graph theory has a great impact. The most common use of graphs occurs in Physics and Chemistry except Mathematics. It is also used in the modeling of Biology, Finance, and Computer science. Basic concepts of graphs are discussed here with classification and figures. In this paper Historical background of graphs, classification, matrix representation of graphs, different types of graph operations, isomorphism, and some important theorems are briefly reviewed. Our main objective is to represent the graph theory in terms of matrix. Future researchers will get a clear and visual concept on graph theory of discrete mathematics by this paper.*

Key Word*: Graphs, Matrix Representation, Incidence matrix, Adjacency Matrix, Cut-Set Matrix, Circuit Matrix, Graph operations.*

Date of Submission: 01-01-2022 Date of Acceptance: 12-01-2022

I. Introduction

Graphs are mathematical structures used to model pair-wise relations between objects from a certain collection. A graph $G = (V, E)$ consists of V, a nonempty set of vertices, and E, a set of edges. Each edge has either one or two vertices associated with it, called its endpoints. An edge is said to connect its endpoints. Vertices can be any abstract data type and can be presented with the points in the plane. These abstract data types are also called nodes. A line or line segment connecting these nodes is called an edge. Again, more abstractly saying, the edge can be an abstract data type that shows the relation between the nodes.

Graphs were first used as a purely mathematical way to solve a fun problem by Leonhard Paul Euler (1707- 1783). Leonhard Paul Euler was a pioneering Swiss mathematician who spent most of his life in Russia and Germany. Euler solved the first problem using graph theory and thereby led to the foundation of a very vast and important field of graph theory. He created the first graph to simulate a real-time place and situation to solve a problem which was then considered one of the toughest problems. The problem was 'Konigsberg bridge problem'. The Konigsberg bridge problem was originated in the city of Konigsberg, formerly in Germany but now known as Kaliningrad and part of Russia, located on the river Preger. The city had seven bridges that connected two islands with the main-land via seven bridges. People staying there always wondered whether was there any way to walk over all the bridges once and only once. The below picture is the map of Konigsberg during Euler's time showing the actual layout of the seven bridges highlighting the river Preger and the bridges.

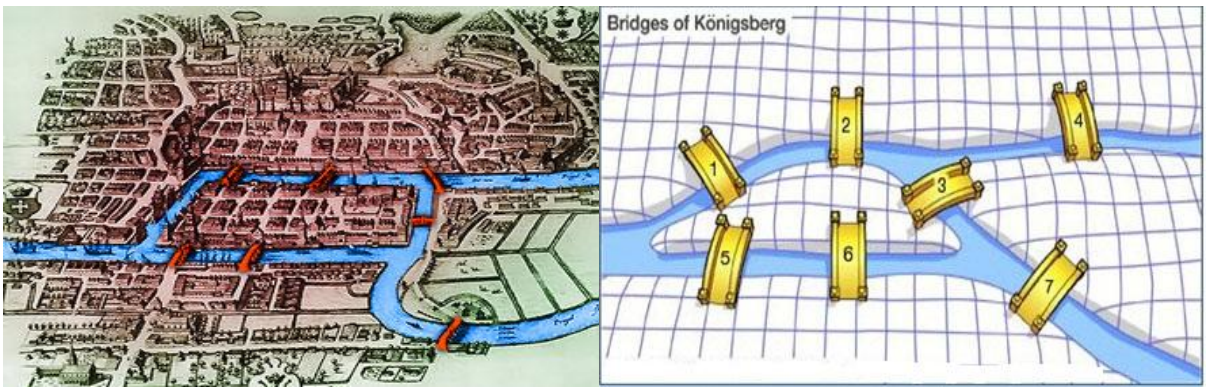


Figure 1: Konigsberg's seven bridges

In 1736 Euler came out with the solution in terms of graph theory. He proved that it was not possible to walk through the seven bridges exactly one time. In coming to this conclusion, Euler formulated the problem in terms of graph theory. He abstracted the case of Konigsberg by eliminating all unnecessary features. He drew a picture consisting of dots that represented the landmasses and the line segments representing the bridges that connected those landmasses. The resulting picture might have looked somewhat similar to the figure shown below.

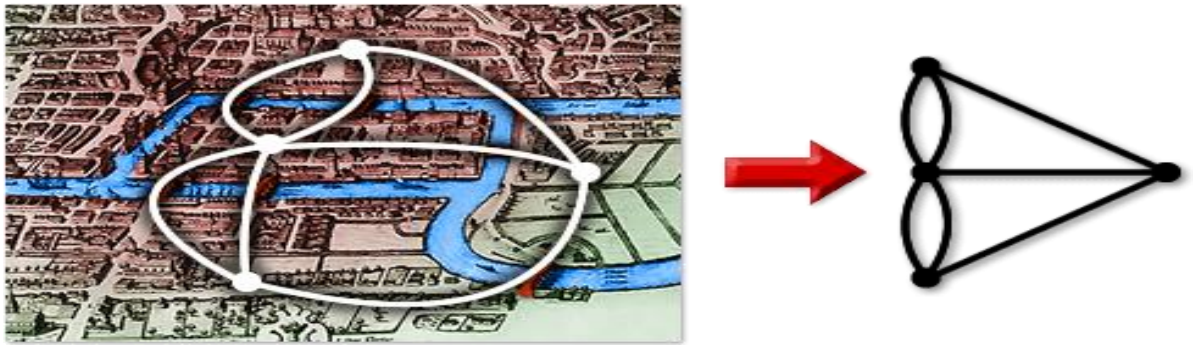
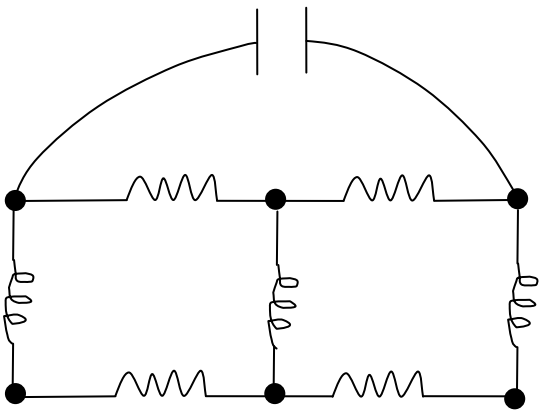


Figure 2: Euler's Graph of Konigsberg bridge

This simplifies the problem to a great extent. Now, the problem can be merely seen as the way of tracing the graph with a pencil without actually lifting it. One can try it in all possible ways but you will soon figure out that it is not possible. But Euler not only proved that it is not possible but also explained why it is not and what should be the characteristic of the graphs so that its edge could be traversed exactly once. He came out with the new concept of the degree of nodes. The Degree of Node can be defined as the number of edges touching a given node. Euler proposed that any given graph can be traversed with each edge traversed exactly once if and only if it had zero or exactly two nodes with odd degrees. The graph following this condition is called the Eulerian circuit or path. We can easily infer this theorem. Exactly two nodes are starting and ending of your trip. If it has even nodes then we can easily come and leave the node without repeating the edge twice or more. In the actual case of seven bridges of Konigsberg, once the situation was presented in terms of the graph the case was simplified as the graph had just 4 nodes with each node having an odd degree. So, Euler concluded that these bridges cannot be traversed exactly once. In 1840, August Ferdinand Mobius (1790–1868) gave the idea of the complete graph and bipartite graph, and Kuratowski (1896 –1980) proved that they are planar by means of recreational problems. Gustav Robert Kirchhoff (1824 –1887) developed the theory of trees in 1847 in order to solve the system of simultaneous linear equations which give the current in each branch and around each circuit of an electric network. Although a physicist, he thought like a mathematician when he abstracted an electric network with its resistances, condensers, inductances, etc., and replaced it by its corresponding combinatorial structure consisting only of points and lines without any indication of the type of electrical element represented by individual lines. Thus, in effect, Kirchhoff replaced each electrical network with its underlying graph and showed that it is not necessary to consider every cycle in the graph of an electric network separately in order to solve the system of equations. Instead, he pointed out by a simple but powerful construction which has since become a standard procedure that the independent cycles of a graph determined by any of its "spanning trees" will suffice. A contrived electrical network, its underlying graph, and a spanning tree are shown in the figure below.



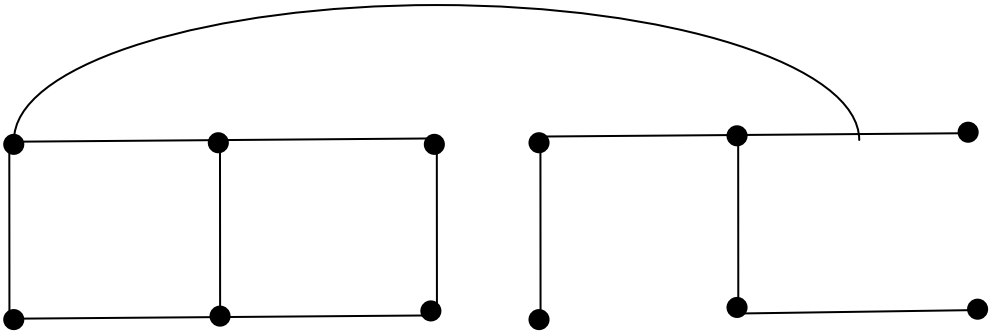


Figure 3: Kirchhoff's circuit

In 1852 Thomas Guthrie (1803–1873) found the famous four-color problem. In 1857 Arthur Cayley (1821–1895) discovered the important class of graphs called trees in the very natural setting of organic chemistry. He was engaged in enumerating the isomers of the saturated hydrocarbons C_nH_{2n+2}, with a given number n of carbon atoms, as shown in the figure below.

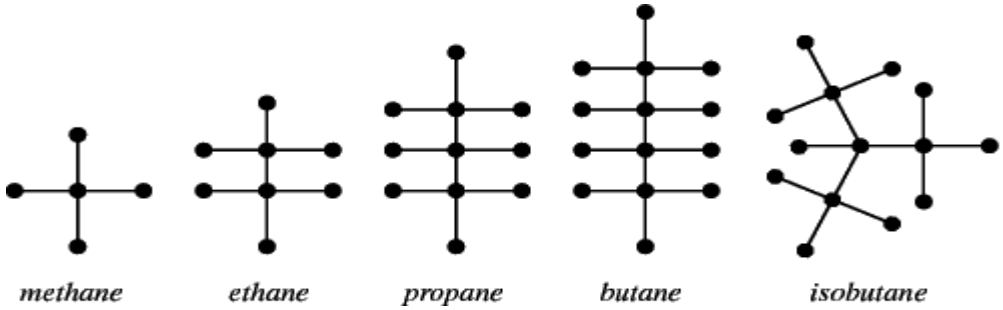


Figure 4: Cayley's isomers

Of course, Cayley restated the problem abstractly: find the number of trees with p points in which every point has degree 1 or 4. He did not immediately succeed in solving this and so he altered the problem until he was able to enumerate: rooted trees (in which one point is distinguished from the others), trees with points of degree at most 4, and finally the chemical problem of trees in which every point has degree 1 or 4. Jordan later (1869) independently discovered trees as a purely mathematical discipline. A game invented by Sir William Rowan Hamilton (1805-1865) in 1859 used a regular solid dodecahedron whose 20 vertices are labeled with the names of famous cities. The player is challenged to travel "around the world" by finding a closed circuit along the edges which passes through each vertex exactly once. Hamilton sold his idea to a toymaker in Dublin for 25 guineas. This was a shrewd move since the game was not a financial success. The figure of the graph is shown below.

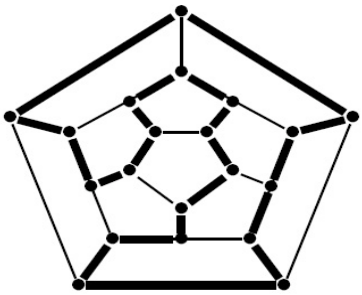


Figure 5: Hamilton's dodecahedron toy

It took 200 years after Euler before the first book on graph theory was written. In 1936 a Jewish Hungarian mathematician Denes Konig (1884 –1944) wrote the first book on graph theory which was originally published in Leipzig, Germany. The name of the book was 'Theorie der endlichen und unendlichen Graphen (Theory of finite and infinite graphs)'. Another book named 'Graph Theory' was written by Frank Harary (1921–2005) which was published in 1969. This book was considered the definitive textbook on the subject worldwide and enabled mathematicians, chemists, electrical engineers, and social scientists to talk to each other.

Since then graph theory has developed into an extensive and popular branch of mathematics that has been applied to many problems in mathematics, computer science, and other scientific areas.

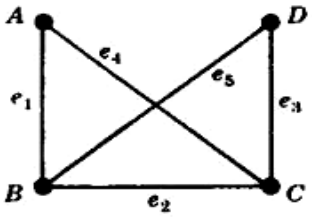
II. Preliminaries

In this section, some definitions related to the graph theory have been discussed which are important for representing our main objective in the later sections.

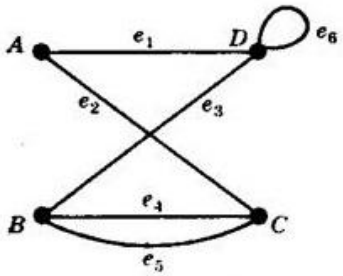
(1) Vertex: A vertex is a point where multiple lines meet. It is also called a **node**. Similar to points, a vertex is also denoted by an alphabet.
Example: Here, the vertex is named as 'a'.

(2) Edge: An edge is a 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: In the below graph 'a' and 'b' are the two vertices and the link between them is called an edge.

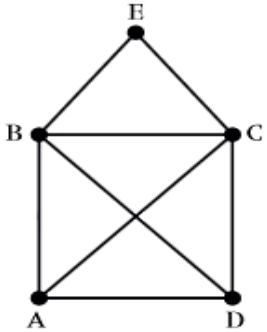
(3) Degree of a Vertex: Degree is defined for a vertex. It is the number of edges connected (coming in or leaving out) to a vertex.
Example: Let us name the vertices in Graph, the vertices A and D have degree 2, the vertices B and C have degree 3.



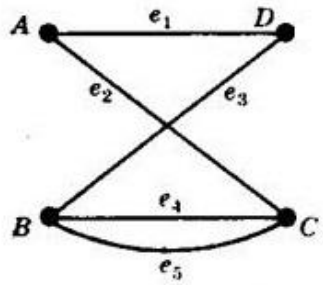
(4) Loops: In a graph, if an edge starts and ends on the same vertex, it is called a loop.
Example: In the Graph below vertex D has a loop.



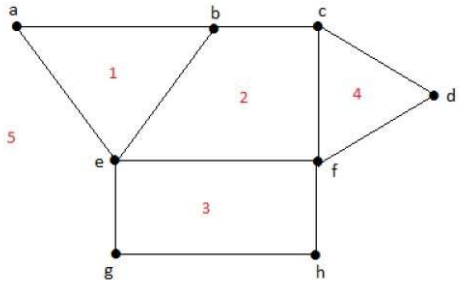
(5) Order of a Graph: The order of a graph is defined as the number of vertices present in the graph.
Example: The order of the graph below is 5.



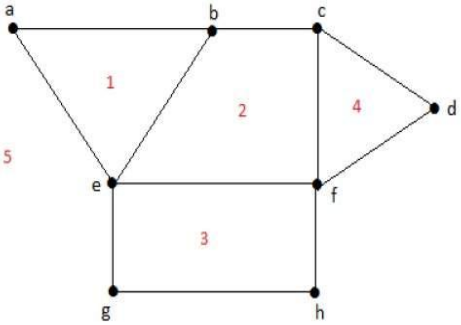
(6) Multiple Edges: In a graph, if two vertices are connected with more than 1 edge, it is called multiple edges.
Example: In the Graph below the vertices B and C have multiple edges.



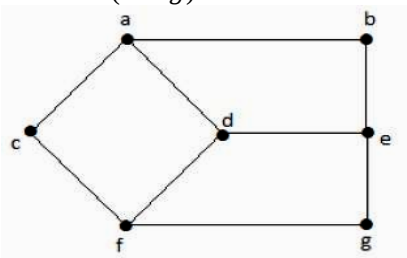
(7) Region: Every planar graph divides the plane into connected areas called regions.
Example: The regions are shown in the graph below.



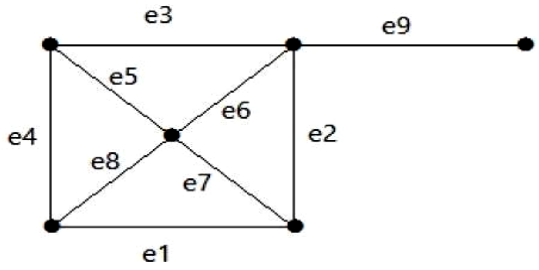
(8) Degree of the region: The degree of a bounded region is the number of edges enclosing the regions r.
Example: In the graph below $deg(1) = 3, deg(2) = 4, deg(3) = 4, deg(4) = 3, deg(5) = 8$.



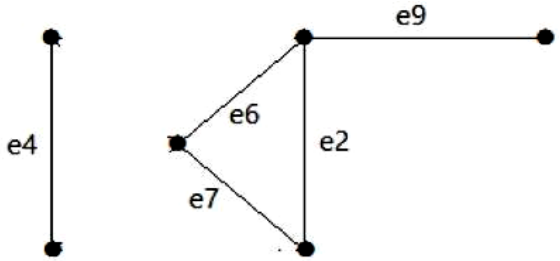
(9) Diameter: The diameter of a graph is the length of the longest shortest path between any pairs of nodes in the graph. It is denoted by $d(G)$.
Example: In the graph below the diameter is 3(a to g).



(10) Cut Set of a Graph: Let $G = (V, E)$ be a connected graph. A subset E' of E is called a cut set of G if deletion of all the edges of E' from G makes G disconnect. If deleting a certain number of edges from a graph makes it disconnected, then those deleted edges are called the cut set of the graph.
Example: Take a look at the following graph. Its cut set is $E_1 = \{e_1, e_3, e_5, e_8\}$.



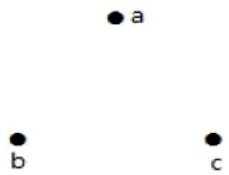
After removing the cut set E_1 from the graph, it would appear as follows –



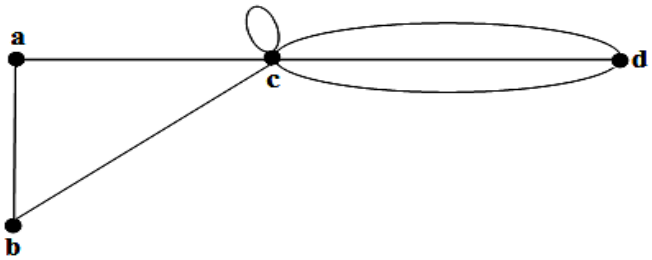
Similarly, other cut sets can disconnect the graph −

- $E_3 = \{e_9\}$ – Smallest cut set of the graph.
- $E_4 = \{e_3, e_4, e_5\}$

(11) Null Graph: A graph that has no edges is called a null graph. The null graph of n vertices is denoted by N_n.
Example:

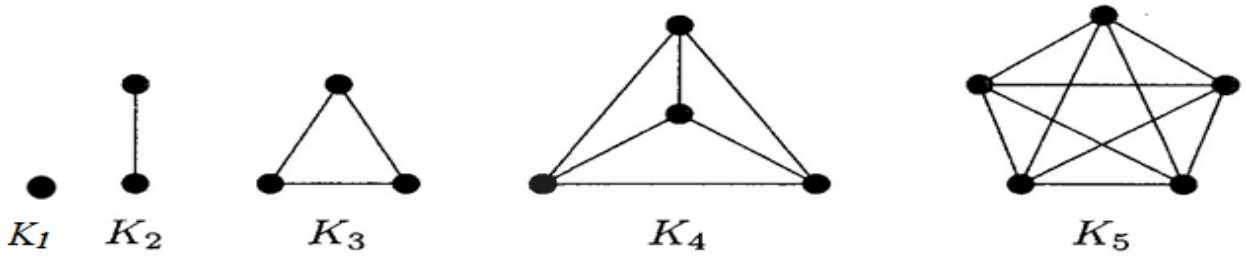


(12) Pseudograph: A graph that contains loops or multiple edges or both is called a pseudograph. Pseudograph = simple graph + multi-edge + loop.
Example:



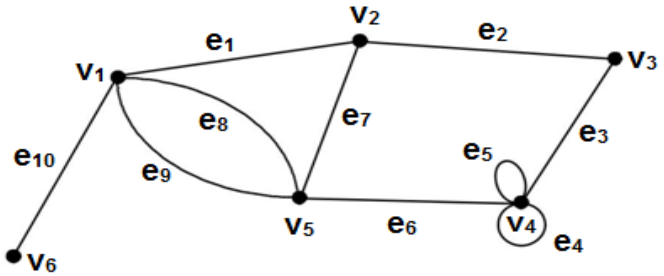
(13) Complete graph: A simple graph that contains exactly one edge between each pair of distinct vertices is called a complete graph. The complete graph with n vertices is denoted by K_n.

Example:



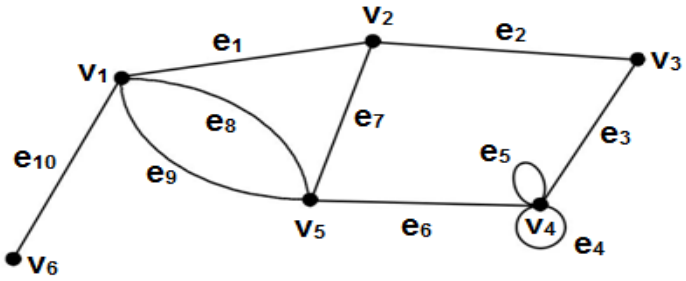
(14) Walk: A walk in a pseudograph is an alternating sequence of vertices and edges, beginning and ending with a vertex, in which each vertex (except the last) is incident with the edge which follows and the last edge is incident with the edge which precedes it. A walk is closed if the first vertex is the same as the last vertex otherwise it is open.

Example: In the graph $\mathbf{v_2, e_7, v_5, e_8, v_1, e_8, v_5, e_6,\ v_4, e_5, v_4, e_5, v_4}$ is an open walk. On the other hand, the walk $\mathbf{v_4, e_5, v_4, e_3, v_3, e_2, v_2, e_7, v_5, e_6, v_4}$ is closed.



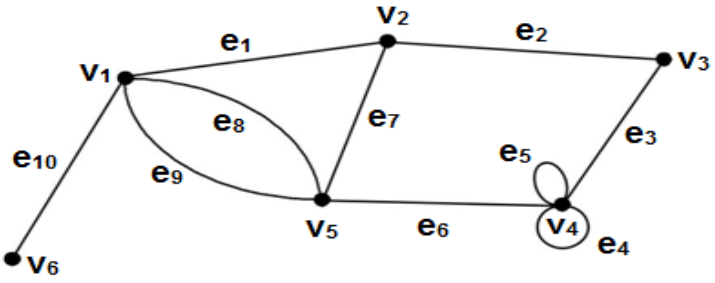
(15) Path: A walk in which all vertices are distinct is called a path.

Example: In the graph the walk $\mathbf{v_2, e_7, v_5, e_6, v_4, e_3, v_3}$ is a path.



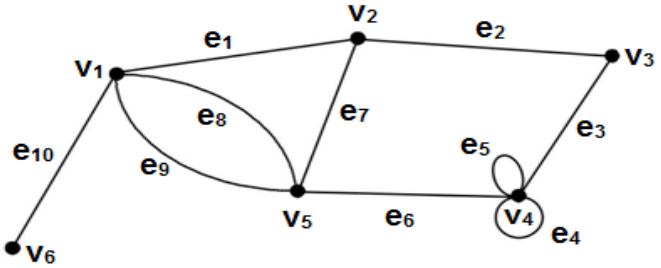
(16) Trail: A walk in which all edges are distinct is called a trail.

Example: The walk in the graph $\mathbf{v_1, e_8, v_5, e_9, v_1, e_1, v_2, e_7, v_5, e_6, v_4, e_5, v_4, e_4, v_4}$ is a trail.

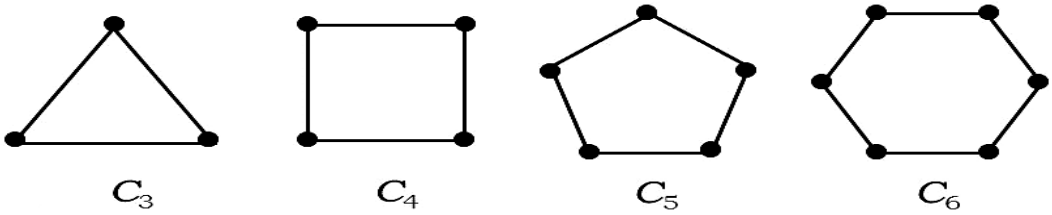


(17) Circuit: A closed trail is called a circuit.

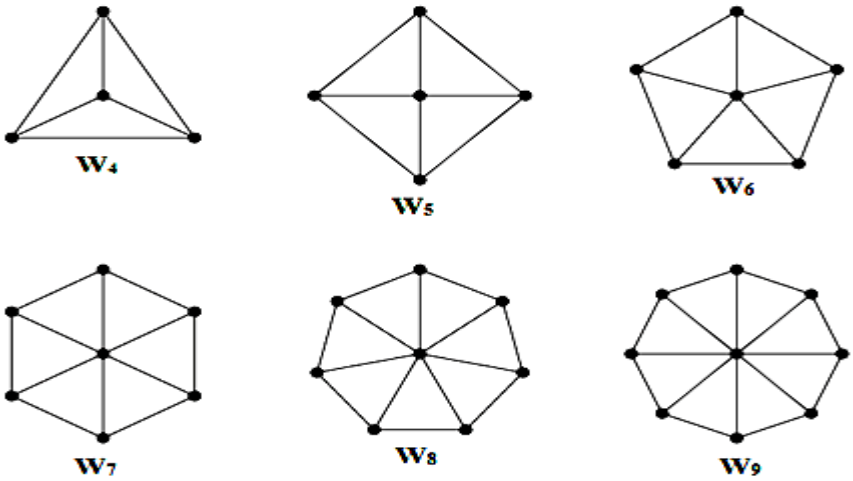
Example: In the graph the walk $\mathbf{v_2, e_7, v_5, e_6, v_4, e_3, v_3, e_2, v_2}$ is a circuit.



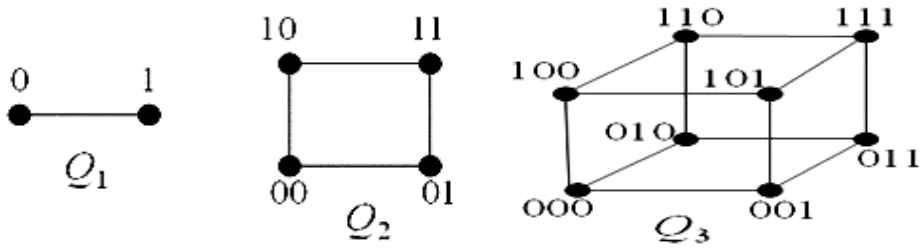
(18) Cycle: A cycle is a closed walk (path) in which all vertices are distinct except first (v_1) and last (v_n). If a graph consists of a single cycle, it is called a cycle graph. The cycle graph with n vertices is denoted by C_n.
Example:



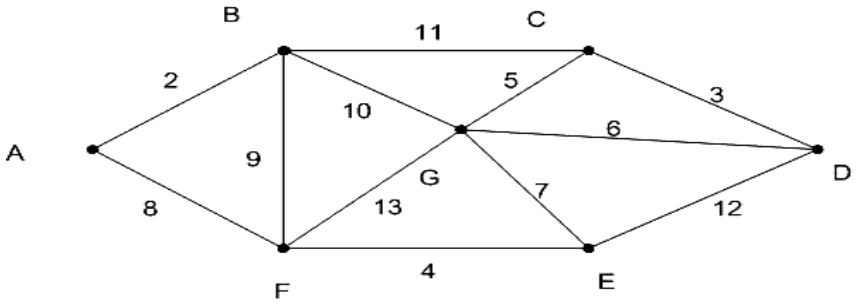
(19) Wheel: By adding an additional vertex to the cycle C_n, for $n \geq 3$ and connecting this new vertex to each of the n vertices in C_n by new edges, we obtain a graph which is called wheel. It is denoted by W_n.
Example:



(20) n-cube: The graph that has vertices representing the 2^n bit strings of length n are called n-dimensional cube or n-cube. It is denoted by Q_n.
Example:

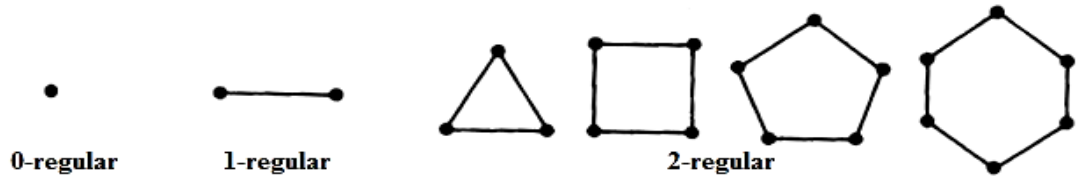


(21) Weighted graph: A graph G is said to be a weighted graph if each edge of G is assigned a non-negative number (weight, length, distance, cost, delay, probability). A weighted graph is therefore a special type of labeled graph.
Example:



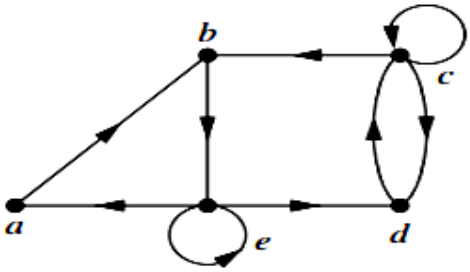
(22) Regular graph: A graph is regular if all the vertices of the graph have the same degree. In a regular graph G of degree n, the degree of each vertex of G is n.

Example:



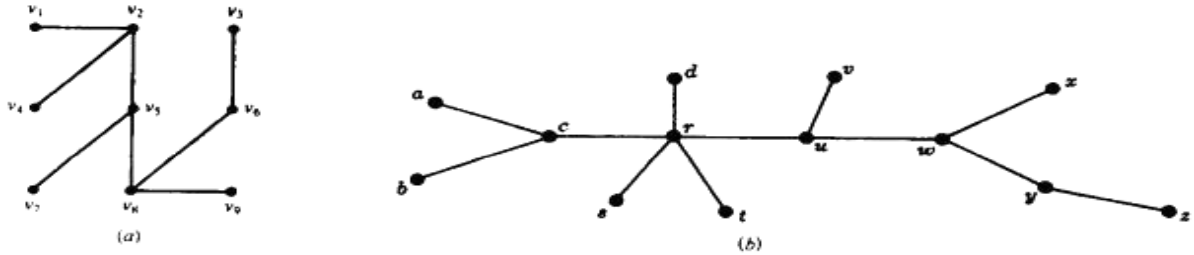
(23) Directed graph: A digraph or directed graph is a graph in which each edge of the graph has a direction. Such an edge is known as the directed edge.

Example:



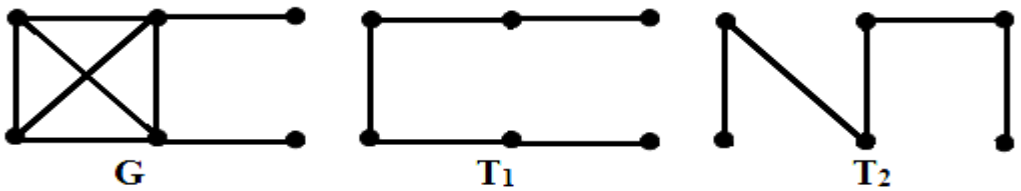
(24) Tree: A tree is a connected graph that contains no cycles. In a tree, every pair of points is connected by a unique path.

Example:



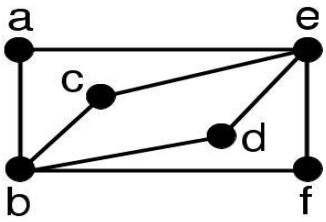
(25) Spanning tree: A spanning tree for a graph G is a sub-graph of G which is a tree that includes every vertex of G.

Example:



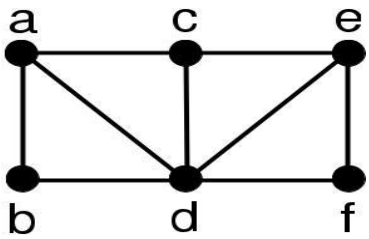
(26) Euler graph: A connected graph G is called an Euler graph if there is a closed trail that includes every edge of the graph G. An Euler path is a path that uses every edge of a graph exactly once. An Euler path starts and ends at different vertices.

Example:



(27) Hamiltonian Graphs: A connected graph G is called a Hamiltonian graph if there is a cycle that includes every vertex of G and the cycle is called the Hamiltonian cycle. Hamiltonian walk in graph G is a walk that passes through each vertex exactly once.

Example:



(28) Isomorphism: Let $G_1 = G_1\ (V_1, E_1)$ and $G_2 = G_2\ (V_2, E_2)$ be two graphs. Then G_1 is said to be isomorphic to G_2 if there is a one-to-one function φ from V_1 into V_2 such that

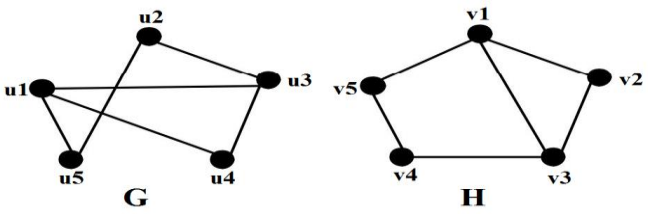
1) If uv is an edge in E_1 then $\varphi\ (u)\ \varphi\ (v)$ is an edge in E_2,
2) Every edge in E_2 has the form $\varphi\ (u)\ \varphi\ (v)$ for some edge in $uv \in E_1$

If G_1 is isomorphic to G_2 then we denote this by $G1 \cong G2$.

If two graphs are isomorphic they must have:

i. the same number of vertices
ii. the same number of edges
iii. the same degrees for corresponding vertices
iv. the same number of connected components
v. the same number of loops
vi. the same number of parallel edges.

Example:



The number of vertices and edges of G and H are the same. Hence there may be an isomorphism between G and H.

Let us define a mapping $\varphi: G \to H$ such that

$$\varphi(u_1) = v_1, \varphi(u_2) = v_4, \varphi(u_3) = v_3, \varphi(u_4) = v_2, \varphi(u_5) = v_5.$$

Now

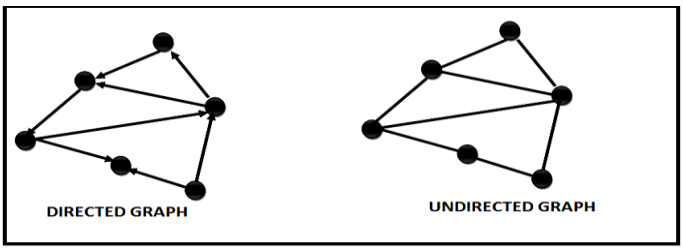
i. u_1u_3 is an edge of G and $\varphi(u_1)\ \varphi(u_3) = v_1v_3$ is an edge of H.
ii. u_1u_4 is an edge of G and $\varphi(u_1)\ \varphi(u_4) = v_1v_2$ is an edge of H.
iii. u_1u_5 is an edge of G and $\varphi(u_1)\ \varphi(u_5) = v_1v_5$ is an edge of H.
iv. u_2u_3 is an edge of G and $\varphi(u_2)\ \varphi(u_3) = v_4v_3$ is an edge of H.
v. u_2u_5 is an edge of G and $\varphi(u_2)\ \varphi(u_5) = v_4v_5$ is an edge of H.
vi. u_3u_4 is an edge of G and $\varphi(u_3)\ \varphi(u_4) = v_3v_2$ is an edge of H.

Thus if u_iu_j is an edge of G then $\varphi(u_i)\ \varphi(u_j)$ is an edge of H. Also, every edge in H has the form $\varphi(u_i)\ \varphi(u_j)$ for some $u_iu_j \in G$. Hence by the definition of isomorphism, the two graphs are isomorphic.

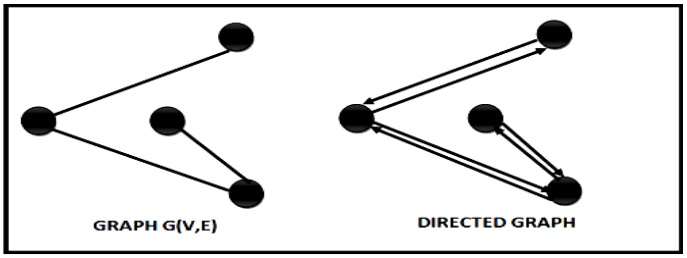
III. Main Results

Graph operations

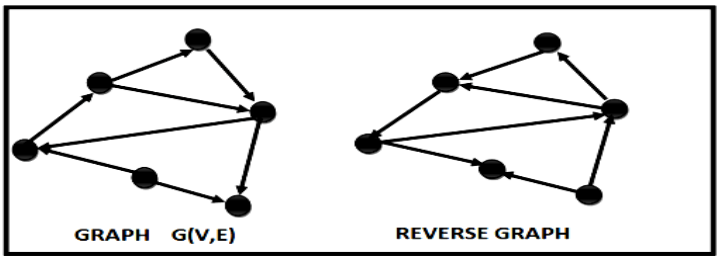
(1) Conversion from Directed Graph to Undirected graph: This is the simplest conversion. A directed graph has directions represented by arrows, in this conversion we just remove all the arrows and do not store the direction information. Also, the graph remains unchanged in terms of its structure.
Example: The below image shows a conversion from a directed graph to an undirected graph.



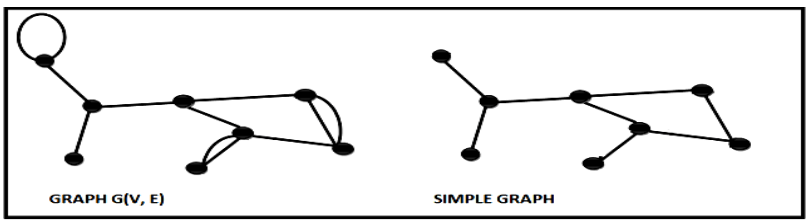
(2) Conversion from Undirected Graph to Directed graph: This conversion gives a directed graph given an undirected graph $G\ (V, E)$. It is the exact reverse of the above. The trick to achieving this is to add one edge for each existing edge in the edge family E. Once the extra edges are added, we just assign the opposite direction to each pair of edges between connecting vertices.
Example: The below image shows a conversion from an undirected graph to a directed graph.



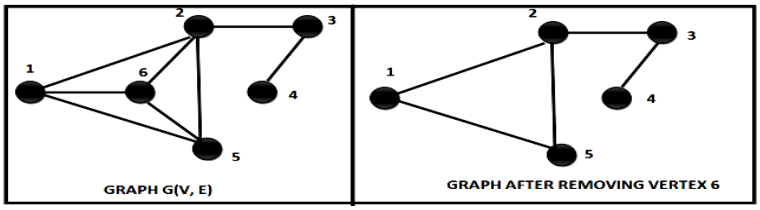
(3) Reversing a graph: Reversing a graph is an operation meant for directed graphs. To reverse a graph we reverse the direction of the edges. The reverse graph has the same vertex set as the original graph.
Example: The below image shows reversing a graph.



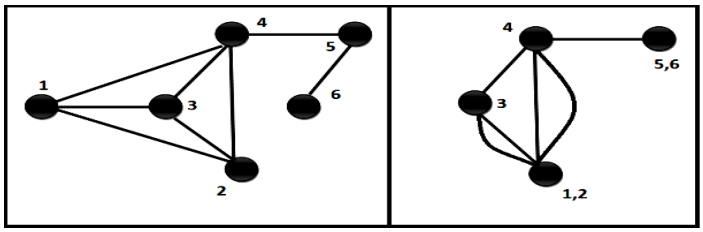
(4) Deriving a Simple graph: The operation is to derive a simple graph out of any given graph. A simple graph by definition must not contain any self-loops or multi edges. As we understand that a graph can contain loops and multiple edges, this operation shall remove the loops and multiple edges from the graph $G\ (V, E)$ to obtain a simple graph.
Example: The below image shows deriving a simple graph from a graph G.



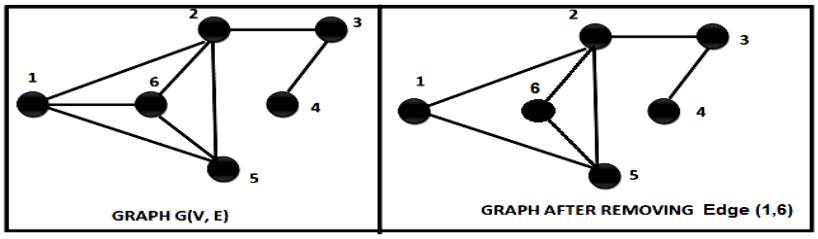
(6) Delete Vertex: This operation changes the vertex set and the edge family of the graph.
Example: The below image shows deleting vertex of a graph G.



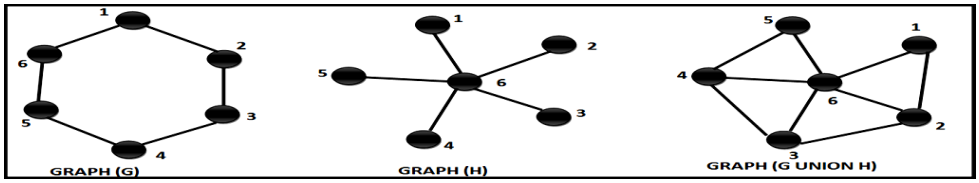
(7) Contract Vertex: Contract vertex can be done by contracting two vertices into one. Also, it can be done by contracting an edge. We cannot contract one vertex, for contraction we need a set of vertices, a minimum of two. Contraction can only be done when there is an edge between the two vertices. The operation basically removes all the edges between the two vertices.
Example: In the below illustration vertices $1, 2$ are contracted and $5, 6$ are also contracted.



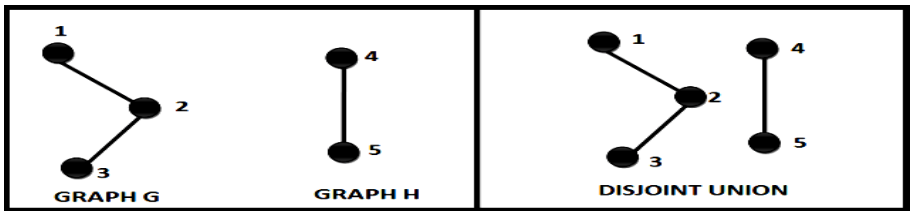
(8) Delete Edge: Deleting an edge can be done by removing the connection between the given vertices.
Example: The below image shows deleting edge from a graph G.



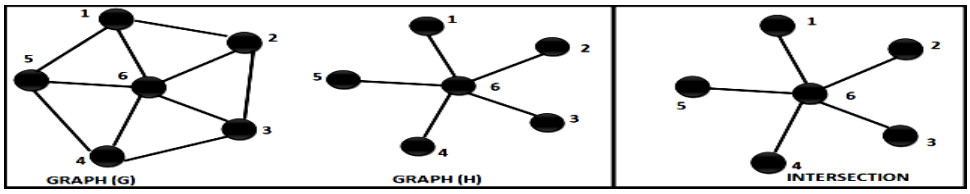
(9) Union of Graphs: The union of two graphs $G\ (V_G, E_G)$ and $H\ (V_H, E_H)$ is the union of their vertex sets and their edge families. That means $G \cup H\ =\ (V_G \cup V_H, E_G \cup E_H)$.
Example: The below image shows a union of graph G and graph H.



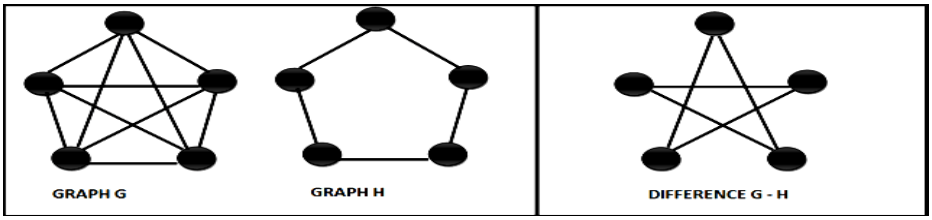
(10) Disjoint union: When V_G and V_H are disjoint then the Union is referred to as disjoint Union and it is denoted by $G + H$.
Example: The vertex sets are disjoint in the two graphs.



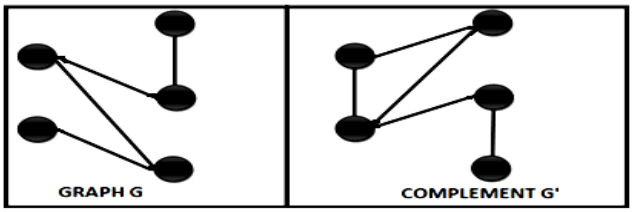
(11) Intersection of Graphs: The intersection of two graphs $G\ (V_G, E_G)$ and $H\ (V_H, E_H)$ is the union of their vertex sets and the intersection of their edge families. That means $G \cap H = (V_G \cup V_H, E_G \cap E_H)$.
Example: The below image shows an intersection of graph G and graph H.



(12) Difference of Graphs: The difference of two graphs $G(V_G, E_G)$ and $H\ (V_H, E_H)$is the union of the vertices of two graphs G and H and the edges in the difference are the difference of edges which is $E_G - E_H$. It is denoted by $G - H$. The Graph Difference of any graph and itself is an empty graph which means $G - G =$ Empty Graph.
Example: The below image shows a difference between graph G and graph H.



(13) Graph Complement: The complement of graph $G\ (V, E)$ is a graph that has the same vertices as G but the edges defined by two vertices in the complement are adjacent only if they are not adjacent in $G\ (V, E)$. It is denoted by G'**.** The Graph complement G' is also called edge complement.
Example: The below image shows a complement of graph G.



Matrix Representation of Graph theory

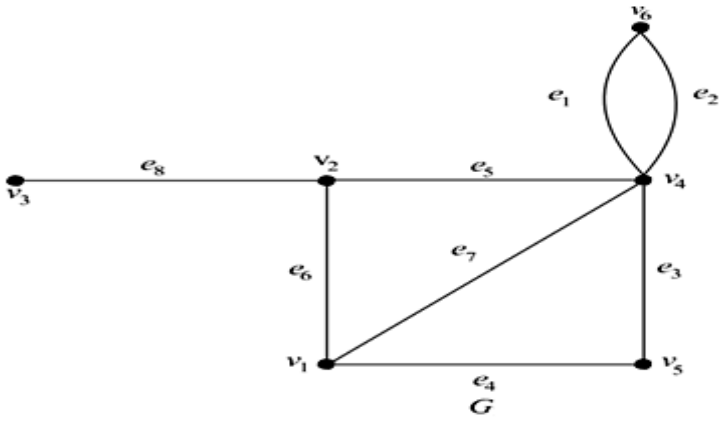
The graph can be represented in the form of the matrix. The matrices that can be formed by a graph are given below.

1. Incidence Matrix
2. Adjacency Matrix
3. Cut-Set Matrix
4. Circuit Matrix
5. Path Matrix

(1) Incidence Matrix: An edge connected to a vertex is known as the incidence edge to that vertex. Let G be a graph with n vertices, m edges, and without self-loops. The incidence matrix A of G is an $n \times m$ matrix $A = [a_{ij}]$ whose n rows correspond to the n vertices and the m columns correspond to m edges such that

$$a_{ij} = \begin{cases} 1, \text{if jth edge } m_j \text{ is incident on the ith vertex} \\ 0, \text{otherwise} \end{cases}$$

It is also called vertex-edge incidence matrix and is denoted by $A(G)$.
Example: Consider the Graph G below



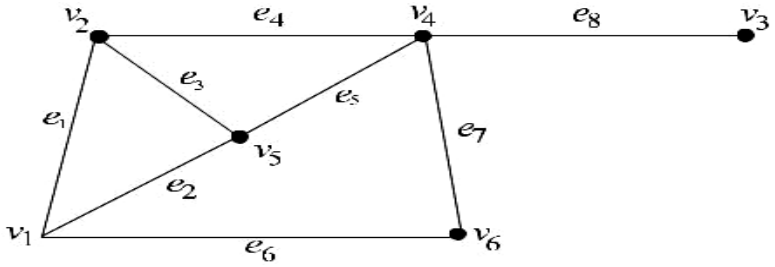
The incidence matrix of G is

$$A(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

(2) Adjacency Matrix: When two vertices are connected by single path then they are known as adjacent vertices. If a vertex is connected to itself then the vertex is said to be adjacent to itself. Let G be a graph with n vertices, m edges. The adjacency matrix A of G is an $n \times m$ matrix $A = [a_{ij}]$ whose n rows correspond to the n vertices and the m columns correspond to m edges such that

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

Example: Consider the Graph G below



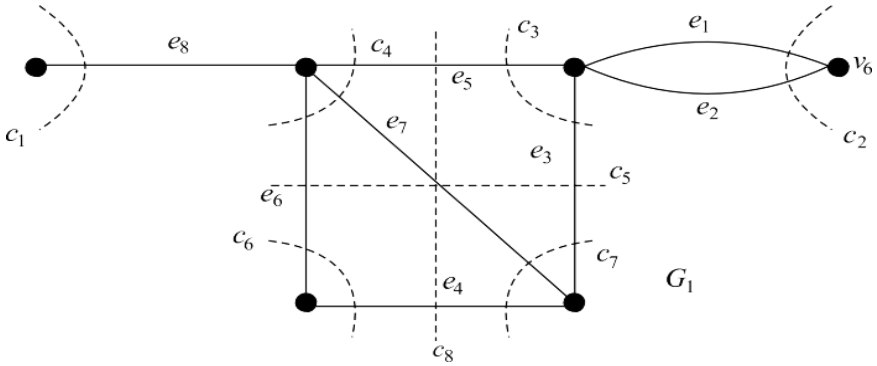
The adjacency matrix of G is

$$A(G) = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

(3) Cut-Set Matrix: Cut set is a set of edges in a graph whose removal leaves the graph disconnected. Let G be a graph with m edges and q cutsets. The cut-set matrix $C = [c_{ij}]_{q \times m}$ of G is a matrix with

$$c_{ij} = \begin{cases} 1, \text{if ith cutset contains jth edge} \\ 0, \text{otherwise} \end{cases}$$

Example: Consider the graph shown in the figure below



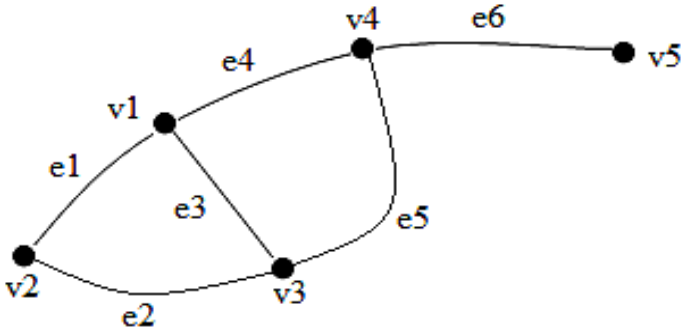
In the graph $G_1, E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$.
The cut-sets are $c_1 = \{e_8\},\ c_2 = \{e_1, e_2\},\ c_3 = \{e_3, e_5\},\ c_4 = \{e_5, e_6, e_7\},\ c_5 = \{e_3, e_6, e_7\},$ $c_6 = \{e_4, e_6\},\ c_7 = \{e_3, e_4, e_7\}\ and\ c_8 = \{e_4, e_5, e_7\}$.
Thus the cut-set matrices are given by

$$C\ (G_1) = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

2.4 Circuit Matrix: Circuit is a close walk in which no vertex/edge can appear twice. Consider a loopless graph $G = (V, E)$ which contains circuits. We enumerate the circuits of G: $C_1, C_2, \ldots\ldots\ldots\ldots\ldots\ldots, C_l$. The circuit matrix of G is an $\ell \times m$ matrix $C = [c_{ij}]$ where

$$c_{ij} = \begin{cases} 1, \text{if ith circuit includes jth edge} \\ 0, \text{otherwise} \end{cases}$$

Example: Consider the graph shown in the figure below



In the graph the circuits are $c_1 = \{e_1, e_2, e_3\},\ c_2 = \{e_3, e_4, e_5\},\ c_3 = \{e_1, e_2, e_5, e_4\}$. Hence the circuit matrix is given by

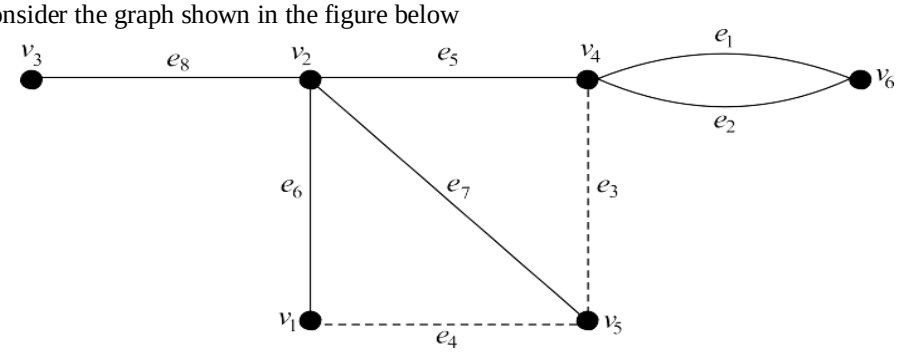
$$C\ (G) = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{cccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

(5) Path Matrix: Path is an open walk in which no vertex /edge can appear twice. Let G be a graph with m edges, and u and v be any two vertices in G. The path matrix for vertices u and v denoted by $P\ (u, v) = [p_{ij}]_{q \times m}$ where q is the number of different paths between u and v, is defined as

$$p_{ij} = \begin{cases} 1, \text{if jth edge lies in theith path} \\ 0, \text{otherwise} \end{cases}$$

Clearly, a path matrix is defined for a particular pair of vertices, the rows in $P(u,v)$ correspond to different paths between u and v, and the columns correspond to different edges in G.

Example: Consider the graph shown in the figure below



The different paths between the vertices v_3 and v_4 are $p_1 = \{e_8, e_5\}$, $p_2 = \{e_8, e_7, e_3\}$ and $p_3 = \{e_8, e_6, e_4, e_3\}$. The path matrix for v_3, v_4 is given by

$$P(v_3, v_4) = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \end{array} \begin{array}{c} \begin{array}{cccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

Theorem 1 (HANDSHAKING THEOREM): If $G = (V, E)$ is an undirected graph with $\boldsymbol{e}$ edges, then

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

Proof: Consider two vertices v_1 and v_2 in V. If $e = \{v_1, v_2\}$ then $a + 1$ is contributed to $\sum_{v \in V} \deg(v)$ for both v_1 and v_2. Thus every non self-loop edge contributes $+2$ to the vertex degree sum. On the other hand, if $e = \{v_1\}$ is a self-loop, then this edge contributes $+2$ to the degree of v_1. Therefore, each edge contributes exactly $+2$ to the vertex degree sum. This completes the proof.

Theorem 2 (EULER'S FORMULA): Let G be a connected planar simple graph with e edges and v vertices. Let r be the number of regions in a planar representation of G. Then $r = e - v + 2$.

Proof: First we specify a planar representation of G. We will prove the theorem by constructing a sequence of subgraphs $G_1, G_2, \ldots\ldots\ldots, G_e = G$, successively adding an edge at each stage. This is done using the following inductive definition. Arbitrarily pick one edge of G to obtain G_1. Obtain G_n from G_{n-1} by arbitrarily adding an edge that is incident with a vertex already in G_{n-1}, adding the other vertex incident with this edge if it is not already in G_{n-1}. This construction is possible because *G* is connected. *G* is obtained after *e* edges are added. Let r_n, e_n and v_n represent the number of regions, edges, and vertices of the planar representation of G_n induced by the planar representation of G, respectively. The proof will now proceed by induction. The relationship $r_1 = e_1 - v_1 + 2$ is true for G_1 because $e_1 = 1, v_1 = 2$ and $r_1 = 1$.

Now assume that $r_k = e_k - v_k + 2$. Let $\{a_{k+1}, b_{k+1}\}$ be the edge that is added to G_k to obtain G_{k+1}. There are two possibilities to consider. In the first case, both a_{k+1} and b_{k+1} are already in G_k. These two vertices must be on the boundary of a common region R, or else it would be impossible to add the edge $\{a_{k+1}, b_{k+1}\}$ to G_k without two edges crossing (and G_{k+1} is planar). The addition of this new edge splits R into two regions. Consequently, in this case, $r_{k+1} = r_{k+1}, e_{k+1} = e_{k+1}$ and $v_{k+1} = v_k$. Thus, each side of the formula relating the number of regions, edges, and vertices increases by exactly one, so this formula is still true. In other words, $r_{k+1} = e_{k+1} - v_{k+1} + 2$.

In the second case, one of the two vertices of the new edge is not already in G_k. Suppose that a_{k+1} is in G_k but that b_{k+1} is not. Adding this new edge does not produce any new regions, because b_{k+1} must be in a region that has a_{k+1} on its boundary. Consequently, $r_{k+1} = r_k$. Moreover, $e_{k+1} = e_{k+1}$ and $v_{k+1} = v_{k+1}$. Each side of the formula relating the number of regions, edges, and vertices remains the same, so the formula is still true. In other words, $r_{k+1} = e_{k+1} - v_{k+1} + 2$. We have completed the induction argument. Hence, $r_n = e_n - v_n + 2$ for all n. Because the original graph is the graph G_e obtained after e edges have been added. This proves the theorem.

Theorem 3 (HALL'S MARRIAGE THEOREM): The bipartite graph $G = (V, E)$ with bipartition (V_1, V_2) has a complete matching from V_1 to V_2 if and only if $|N(A)| \geq |A|$ for all subsets A of V_1.

Proof: We first prove the only if part of the theorem. To do so, suppose that there is a complete matching M from V_1 to V_2. Then, if $A \subseteq V_1$, for every vertex $v \in A$, there is an edge in M connecting v to a vertex in V_2. Consequently, there are at least as many vertices in V_2 that are neighbors of vertices in V_1 as there are vertices in V_1. It follows that $|N(A)| \geq |A|$. To prove the if part of the theorem, the more difficult part, we need to show that if $|N(A)| \geq |A|$ for all $A \subseteq V_1$, then there is a complete matching M from V_1 to V_2. We will use strong induction on $|V_1|$ to prove this.
Basis step: If $|V_1| = 1$, then V_1 contains a single vertex v_0. Because $|N(\{v_0\})| \geq |\{v_0\}| = 1$, there is at least one edge connecting v_0 and a vertex $w_0 \in V_2$. Any such edge forms a complete matching from V_1 to V_2.
Inductive step: We first state the inductive hypothesis.
Inductive hypothesis: Let k be a positive integer. If $G = (V, E)$ is a bipartite graph with bipartition (V_1, V_2), and $|V_1| = j \leq k$, then there is a complete matching M from V_1 to V_2 whenever the condition that $|N(A)| \geq |A|$ for all $A \subseteq V_1$ is met.
Now suppose that $H = (W, F)$ is a bipartite graph with bipartition (W_1, W_2) and $|W_1| = k + 1$. We will prove that the inductive holds using a proof by cases, using two cases. Case (i) applies when for all integers j with $1 \leq j \leq k$, the vertices in every set of j elements from W_1 are adjacent to at least $j + 1$ elements of W_2. Case (ii) applies when for some j with $1 \leq j \leq k$ there is a subset W_1' of j vertices such that there are exactly j neighbors of these vertices in W_2. Because either Case (i) or Case (ii) holds, we need only consider these cases to complete the inductive step.

Case (i): Suppose that for all integers j with $1 \leq j \leq k$, the vertices in every subset of j elements from W_1 are adjacent to at least $j + 1$ elements of W_2. Then, we select a vertex $v \in W_1$ and an element $w \in N(\{v\})$, which must exist by our assumption that $|N(\{v\})| \geq |\{v\}| = 1$. We delete v and w and all edges incident to them from H. This produces a bipartite graph H' with bipartition $(W_1 - \{v\}, W_2 - \{w\})$. Because $|W_1 - \{v\}| = k$, the inductive hypothesis tells us there is a complete matching from $W_1 - \{v\}$ to $W_2 - \{w\}$. Adding the edge from v to w to this complete matching produces a complete matching from W_1 to W_2.

Case (ii): Suppose that for some j with $1 \leq j \leq k$, there is a subset W_1' of j vertices such that there are exactly j neighbors of these vertices in W_2. Let W_2' be the set of these neighbors. Then, by the inductive hypothesis, there is a complete matching from W_1' to W_2'. Remove these 2j vertices from W_1 and W_2 and all incident edges to produce a bipartite graph K with bipartition $(W_1 - W_1', W_2 - W_2')$. We will show that the graph K satisfies the condition $|N(A)| \geq |A|$ for all subsets A of $W_1 - W_1'$. If not, there would be a subset of t vertices of $W_1 - W_1'$ where $1 \leq t \leq k + 1 - j$ such that the vertices in this subset have fewer than t vertices of $W_2 - W_2'$ as neighbors. Then the set of $j + t$ vertices of W_1 consisting of these t vertices together with the j vertices we removed from W_1 has fewer than $j + t$ neighbors in W_2, contradicting the hypothesis that $|N(A)| \geq |A|$ for all $A \subseteq W_1$.
Hence, by the inductive hypothesis, the graph K has a complete matching. Combining this complete matching with the complete matching from W_1' to W_2', we obtain a complete matching from W_1 to W_2.
We have shown that in both cases there is a complete matching from W_1 to W_2. This completes the inductive step and completes the proof.

Theorem 5: Let $T = (V, E)$ be a graph with $|V| = n$. Then the following are equivalent:

(1) T is a tree.
(2) T is acyclic and has exactly $n - 1$ edges.
(3) T is connected and has exactly $n - 1$ edges.
(4) T is connected and every edge is a cut-edge.
(5) Any two vertices of T are connected by exactly one path.
(6) T is acyclic and the addition of any new edge creates exactly one cycle in the resulting graph.

Proof: $(1 \Longrightarrow 2)$ Assume T is a tree. Then by definition, T is acyclic, and the fact that it has $n - 1$ edges.

$(2 \Longrightarrow 3)$ Since T is acyclic, it must be a forest and we know $|E| = n - c(T)$. Since we assumed that T has $n - 1$ edges, we must have $n - c(T) = n - 1$ and thus the number of components of T is 1 and thus T must be connected.

$(3 \Longrightarrow 4)$ The fact that T is connected is assumed from 3. Suppose we consider the graph $T' = (V, E')$ where $E' = E\backslash\{e\}$. Then the number of edges in T' is $n-2$. The graph T'contains n vertices and must still be acyclic (that is a forest) and therefore $n-2 = n-c\ (T')$. Thus $c\ (T') = 2$ and e was a cut-edge.

$(4 \Longrightarrow 5)$ Choose two vertices v and v' in V. The fact that there is a path between v and v' is guaranteed by our assumption that T is connected. By way of contradiction, suppose that there are at least two paths from v to v' in T. These two paths must diverge at some vertex $w \in V$ and recombine at some other vertex w'. (See below Figure) We can construct a cycle in T by beginning at vertex w following the first path to w' and the following the second path back to w from w'.

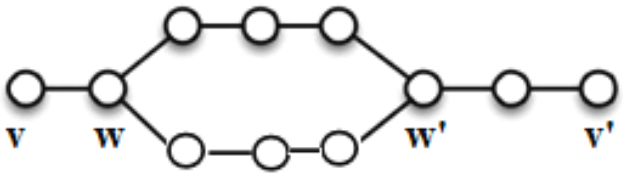


Figure 6: The proof of $4 \Longrightarrow 5$ requires us to assume the existence of two paths in graph T connecting vertex v to vertex v'. This assumption implies the existence of a cycle, contradicting our assumptions on T.

By Definition removing any edge in this cycle cannot result in a disconnected graph. Thus no edges in the constructed cycle in a cut-edge are contradicting our assumption on T. Thus, two paths connecting v and v' cannot exist.

$(5 \Longrightarrow 6)$ The fact that any pair of vertices is connected in T implies T is connected (i.e., has one component). Now suppose that T has a cycle (like the one illustrated in Figure above). Then it is easy to see there are (at least) two paths connecting w and w' contradicting our assumption. Therefore, T is acyclic. The fact that adding an edge creates exactly one cycle can be seen in the following way: Consider two vertices v and $v'$and suppose the edge $\{v, v'\}$ is not in E. We know there is a path

$$(v, \{v, u_1\}, u_1, \dots\dots\dots\dots, u_n, \{u_n, v'\}, v')$$

in T connecting v and v' and it is unique. Adding the edge $\{v, v'\}$ creates the cycle

$$c_1 = (v, \{v, u_1\}, u_1, \dots\dots\dots\dots, u_n, \{u_n, v'\}, v', \{v, v'\}, v)$$

so at least one cycle is created. To see that this cycle is unique, note that if there is another cycle present then it must contain the edge $\{v, v'\}$. Suppose that this cycle is

$$c_2 = (v, \{v, w_1\}, w_1, \dots\dots\dots\dots, w_n, \{w_n, v'\}, v', \{v, v'\}, v)$$

where there is at least one vertex w_i not present in the set $\{u_1, \dots\dots\dots\dots, u_n\}$ (otherwise, the two cycles are identical). We now see there must be two disjoint paths connecting v and v', namely

$$(v, \{v, u_1\}, u_1, \dots\dots\dots\dots, u_n, \{u_n, v'\}, v')$$

and

$$(v, \{v, w_1\}, w_1, \dots\dots\dots\dots, w_n, \{w_n, v'\}, v')$$

this contradicts our assumption on T. Thus the created cycle is unique.

$(6 \Longrightarrow 1)$ It suffices to show that T has a single component. Suppose not, there are at least two components of T. Chose two vertices v and v' in V so that these two vertices are not in the same component. Then the edge $e = \{v, v'\}$ is not in E, and adding it to E cannot create a cycle. To see why not that if T' is the graph that results from the addition of e then e is now a cut-edge. We see that e cannot lie on a cycle and thus the addition of this edge does not create a cycle, contradicting our assumption on T. Thus, T must have a single component. Since it is acyclic and connected hence T is a tree. This completes the proof.

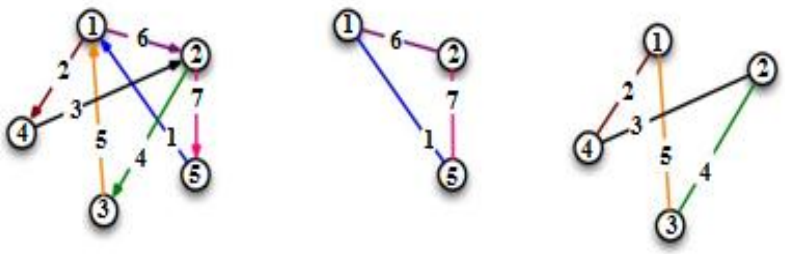
Theorem 6: Let $G = (V, E)$ be a non-empty, non-trivial connected graph G. Then the following are equivalent:

(1) G is Eulerian.
(2) The degree of every vertex in G is even.
(3) The set E is the union of the edge sets of a collection of edge-disjoint cycles in G.

Proof: $(1 \Longrightarrow 2)$ Assume G is Eulerian then there is an Eulerian tour t of G. Let v be a vertex in G. Each time v is traversed while following the tour; we must enter v by one edge and leave by another. Thus, v must have an even number of edges adjacent to it. If v is the initial (and final) vertex in the tour, then we leave v in the very first step of the tour and return in the last stage, thus the initial (and final) vertex of the tour must also have an even degree. Thus every vertex has an even degree.

$(2 \Longrightarrow 3)$ Since G is connected and every vertex has an even degree, it follows that the degree of each vertex is at least 2. By definition, G must contain a cycle C. If this cycle includes every edge in G, then (3) is established. Suppose otherwise. Consider the graph G' obtained by removing all edges in C. If we consider C as a subgraph of G, then each vertex in C has exactly two edges adjacent to it. Thus if v is a vertex in the cycle, then removing the edges in C that are adjacent to it will result in a vertex v having 2 fewer edges in G' than it did in G. Since we assumed that every vertex in G had an even degree, it follows that every vertex in G' must also have an even degree (since we removed) either 2 or 0 edges from each vertex in G to obtain G'. We can repeat the previous process of constructing a cycle in $G'$and if necessary forming G''. Since there are a finite number of edges in G, this process must stop at some point and we will be left with a collection of edge disjoint cycles $C = \{C, C', \dots\dots\dots\dots\}$ whose union is the entire edge set of G.

$(3 \Longrightarrow 1)$ Assume that G is connected and that its edge set is the union of a collection of edge-disjoint cycles. We proceed by induction on the number of cycles. If there is only one cycle, then we simply follow this cycle in either direction to obtain a tour of G. Now suppose that the statement is true for a graph whose edge set is the union of $\leq n$ edge disjoint cycles. We'll show that the statement is true for a graph whose edge set is composed of $n+1$ edge disjoint cycles. Denote the cycles $C_1, \dots\dots\dots\dots, C_{n+1}$. A subgraph G' of G composed of only cycles $C_1, \dots\dots\dots, C_n$ will have m components with $1 \leq m \leq n$. Each component is composed of at most n edge disjoint cycles and therefore applying the induction hypothesis, each has a tour. Denote the components $K_1, \dots\dots\dots\dots, K_m$. The cycle C_{n+1} shares one vertex in common with at least one of these components (and perhaps all of them). Without loss of generality, assume that K_1 is a component sharing a vertex in common with C_{n+1} (if not, reorder the components to make this true). Begin following the tour around K_1 until we encounter the vertex v_1 that component K_1 and C_{n+1} share. At this point, break the tour of K_1 and begin traversing C_{n+1} until (i) we return to v_1 or (ii) we encounter a vertex v_2 that is shared by another component (say K_2). In case (i), we complete the tour of K_1 and necessarily we must have completed a tour of the entire graph since it is connected. In case (ii) we follow the tour of K_2 until we return to v_2 and then continue following C_{n+1} until either case (i) occurs or case (ii) occurs again. In either case, we apply the same logic as before. Since there are a finite number of components, this process will eventually terminate with case (i), we complete the tour of K_1 and thus we will have constructed a tour of the entire graph. This theorem is illustrated in the figure below. This completes the proof.



IV. Conclusion

In this work, graphs are discussed with simple examples and theorems to explain easily. The historical background of graphs states that many important topics of science could not possible to explain without the help of graph theory. Many practical problems can be easily represented in terms of graph theory. This may help future researchers to proceed further.

References

[1]. Discrete Mathematics and its applications - Kenneth H. Rosen (7th edition)
[2]. Discrete Mathematics - Prof. Dr. Md. Ayub Ali and Prof. Dr. M.F Rahman
[3]. Graph theory - Harary

[4]. Graph Theory with Applications - J.A. Bondy and U.S.R. Murty
[5]. Graph Theory- Reinhard Diestel
[6]. Introduction to Graph Theory - Douglas West (2nd edition)
[7]. Modern Graph Theory- B. Bollobas, Springer-Verlag.
[8]. Graph theory - Keijo Ruohonen
[9]. Complex Graphs and Networks - Fan Cheung and Linyuan Lu.
[10]. Graphs, Networks and Algorithms - Dieter Jungnickel,Vol. 5, Springer Verlag, Berlin.
[11]. An Inside Guide To Algorithms - Siegel and Cole.
[12]. Graph Theory -Tero Harju, Department of Mathematics, University of Turku, Finland.
[13]. https://en.wikipedia.org/wiki/Graph_theory
[14]. https://www.tutorialspoint.com/graph_theory/index.htm
[15]. http://mathforum.org/isaac/problems/bridges2.html
[16]. http://www-history.mcs.st-andrews.ac.uk/Biographies/Euler.html