GRAPH THEORY
A PROJECT REPORT

Project Internship Report Submitted by

MELVIN MATHEW

KH.SC.I5MAT17021

Under the Supervision and Guidance of

Mr.PARAMESWARAN.R

Assistant Professor

Department of Mathematics

in partial fulfillment of the requirement of

AMRITA VISHWA VIDYAPEETHAM

for the award of the degree of

BACHELOR OF MATHEMATICS

AMRITA VISHWA VIDYAPEETHAM

DEPARTMENT OF MATHEMATICS

AMRITA SCHOOL OF ARTS AND SCIENCES, KOCHI

MAY 2020

A PROJECT REPORT

Submitted by

MELVIN MATHEW
KH.SC.I5MAT17021

Under the guidance of

Mr.PARAMESWARAN.R
Assistant Professor,
Department of Mathematics

in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF SCIENCE

IN

MATHEMATICS

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ARTS AND SCIENCES, KOCHI

MAY 2020

AMRITA SCHOOL OF ARTS AND SCIENCES
KOCHI

DEPARTMENT OF MATHEMATICS

PROJECT WORK

MAY 2020

This is to certify that the project Work / entitled

GRAPH THEORY

is the bonafide record of project work done by

MELVIN MATHEW
KH.SC.I5MAT17021

Of Bachelor of Science in Mathematics during the year 2020.

PARAMESWARAN.R — PROJECT GUIDE

DR.ARCHANA — HEAD OF THE DEPARTMENT

Submitted for the Project Viva-Voce examination held on__________________

---------------------------- --------------------------

Internal Examiner External Examiner

DECLARATION

I affirm that the Project work titled “Graph Theory**”** being submitted in partial fulfillment for the award of the **DEGREE OF BACHELOR OF SCIENCE** in **MATHEMATICS** is the original work carried out by me. It has not formed the part of any other project work/ internship submitted for award of any degree or diploma, either in this or any other University**.**

Place: Kochi (Signature)

Date: Name of the student: Melvin Mathew

Register Number: KH.SC.I5MAT17021

DEDICATION

TO MY FAMILY

ACKNOWLEDGEMENT

This project has taken sufficient amount of time and effort from my side but its implementation would not have been possible without the help of many generous personalities. I would like to extend my sincere gratitude towards all of them.

I thank Mr.R.Parameswaran, my project guide, for his continuous support and supervision. I also thank other faculties of Department of Mathematics, Amrita School of Arts and Sciences, Kochi for their support and help. I am also grateful for the constant support of my classmates. The encouragement that I received from my parents was also important at every stage and I am thankful to them. I acknowledge every single person who has helped in the successful completion of this project.

CONTENTS

AN INTRODUCTION TO GRAPHS

INTRODUCTION

A graph is a picture designed to express words, particularly the connection between two or more quantities. Graphs are a common method to visually illustrate relationship in the data. The purpose of a graph is to present data that are too numerous or complicated to be described adequately in the text and in less space. Many real-world situations are conveniently be described by the means of a diagram consisting of a set of points together with lines joining certain pairs of these points. For example, the points could represent people, with lines joining pairs of friends; or the points might be communication centres, with lines representing communication links. Notice that in such diagrams one is mainly interested in whether or not two given points are joined by a line; the manner in which they are joined is immaterial. A mathematical abstraction of situations of this type given rise to the concept of a graph.

HISTORY

The paper written by Leonhard Euler on the Seven Bridges of Königsberg and published in 1736 is regarded as the first paper in the history of graph theory. One of the most famous and stimulating problems in graph theory is the four colour problem: "Is it true that any map drawn in the plane may have its regions coloured with four colours, in such a way that any two regions having a common border have different colours?" The four colour problem remained unsolved for more than a century. In 1969 Heinrich Heesch published a method for solving the problem using computers. A computer-aided proof produced in 1976 by Kenneth Appel and Wolfgang Haken makes fundamental use of the notion of "discharging" developed by Heesch. The proof involved checking the properties of 1,936 configurations by computer, and was not fully accepted at the time due to its complexity. A

simpler proof considering only 633 configurations was given twenty years later by Robertson, Seymour, Sanders and Thomas.

DEFINITION

A graph G= (V(G),E(G)) consists of two finite sets:

V(G), the vertex set of the graph, often denoted by just V, which is a nonempty set of elements called vertices, and E(G), the edge set of the graph, often denoted by just E, which is a possibly empty set of elements called edges, such that each edge e in E is assigned an unordered pair of vertices (u, v), called the end vertices of e.

Vertices are also sometimes called points, nodes, or just dots. If e is an edge with end vertices u and vthen e is said to join u and v. An example of graph is given below.

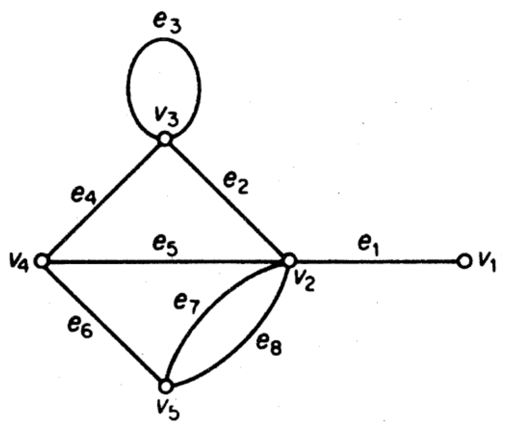


Fig.1.1

In the given figure vertex set is given as V and E

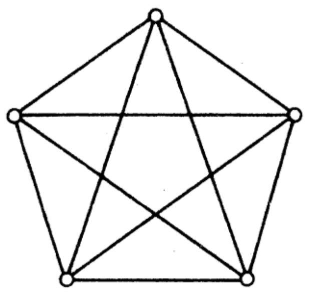
$V=\{v_1,v_2,v_3,v_4,v_5\}$ $E=\{e_1,e_2,e_3,e_4,e_5,e_6,e_7,e_8\}$

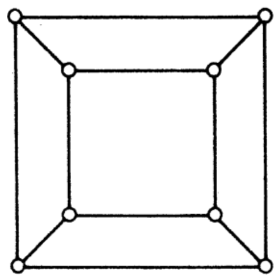
A vertex of G which is not the end of any edge is called isolated.Two vertices which are joined by an edge are said to be adjacent or neighbours.The set of all neighbours of a fixed vertex v of G is called the neighbourhood set of v and is denoted by N(v).A vertex is incident to an edge if the vertex is one of the two vertices the edge connects. For example from the above figure the vertex v_1 is incident to edge e_1.If two or more edges of graph G have the same

end vertices then these edges are called parallel. In the above figure e_7 and e_8 are parallel edges. Edge that connect vertex to itself is called loop. Here e_3 is the loop. The degree of a vertex is the number of edges that are incident of the vertex. The degree of a vertex is denoted deg(v) or degv.A vertex of a graph is called odd or even depending on whether its degree is odd or even. For example the vertex v_4 has degree 3 and hence it is an odd vertex.

TYPES OF GRAPHS

- **A graph is called simple if it has no loops or parallel edges.**
- **A graph which is not simple is called a multigraph.**
- **A complete graph is a simple graph in which each pair of distinct vertices is joined by an edge. Fig.1.2 given below is an example of complete graph.**
- **An empty (or trivial) graph is a graph with no edges.**
- **A bipartite graph is a graph whose vertices can be divided into disjoint and independent sets and such that every edge connects a vertex in to one in. Vertex sets and are usually called the parts of the graph. Fig.1.3 given below is an example of bipartite graph.**
- **A complete bipartite is a simple bipartite graph where every vertex of the first set is connected to every vertex of the second set and it is denoted by $K_{m,n}$. Fig.1.4 given below shows an example of complete bipartite graph.**
- **An-regular graph is a graph where every vertex has same degree say n.**
- **The complete bipartite graphs $K_{1,n}$, known as the star graphs, are trees.**





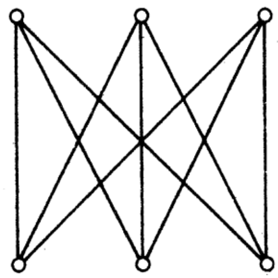


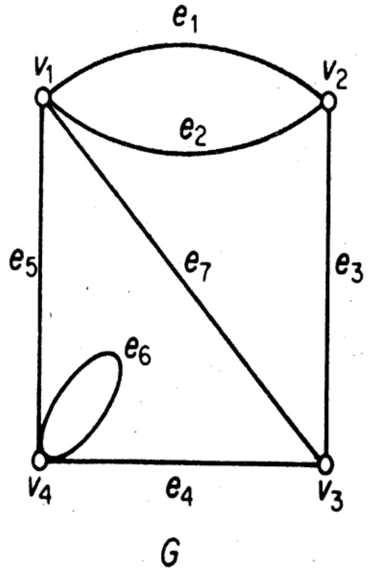
Fig.1.2 Fig.1.3 Fig.1.4

Complete graph **Bipartite graph** **Complete bipartite**

MATRIX REPRESENTATION OF GRAPH

To any graph G there corresponds a v*e matrix called the incidence matrix of G. Let us denote the vertices of G by $v_1, v_2, \ldots\ldots v_v$ and the edges by $e_1, e_2, \ldots\ldots e_e$. Then the incidence matrix of G is the matrix $M(G)=[m_{ij}]$, where m_{ij} is the number of times (0,1or2) that v_i and e_j are incident. The incidence matrix of a graph is just a different way of specifying the graph.

Another matrix associated with G is the adjacency matrix; this is the v×v matrix $A(G)=[a_{ij}]$, in which a_{ij} is the number of edges joining v_i and v_j. A graph, its incidence matrix, and its adjacency matrix are shown in below fig.1.5



	e_1	e_2	e_3	e_4	e_5	e_6	e_7
v_1	1	1	0	0	1	0	1
v_2	1	1	1	0	0	0	0
v_3	0	0	1	1	0	0	1
v_4	0	0	0	1	1	2	0

M(G)

	v_1	v_2	v_3	v_4
v_1	0	2	1	1
v_2	2	0	1	0
v_3	1	1	0	1
v_4	1	0	1	1

A(G)

Fig.1.5

GRAPH ISOMORPHISM

A graph $G_1 = (V_1,E_1)$ is said to be isomorphic to the graph $G_2 = (V_2, E_2)$ if there is a one-to-one correspondence between the vertex sets V_1 and V_2 and a one-to-one correspondence between the edge sets E_1 and E_2in such a way that if e_1 is an edge with end vertices u_1 and v_1 in G_1 then the corresponding edge e_2 in G_2 has its end points the vertices u_2 and v_2 in G_2 which correspond to u_1 and v_1 respectively. Such a pair of correspondences is called a graph isomorphism.Some pairs of isomorphic graphs are shown below

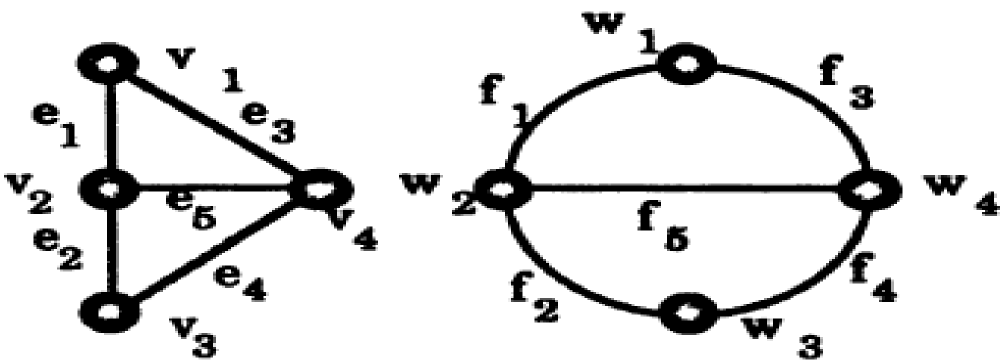


Fig.1.6

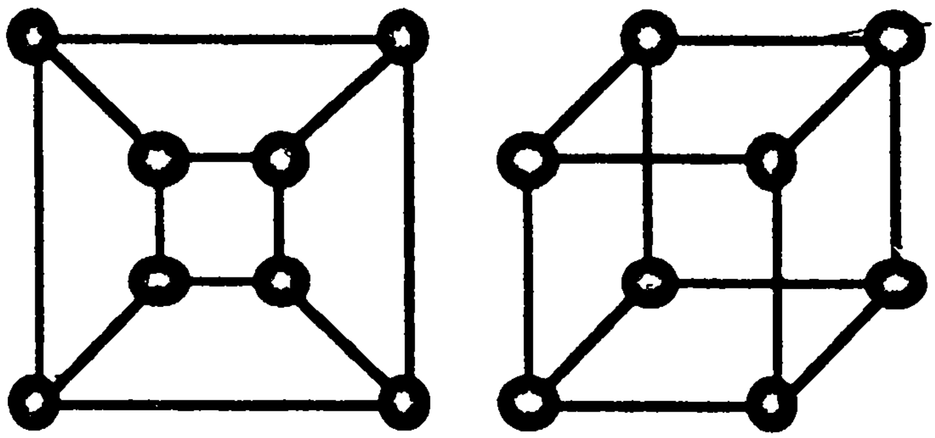


Fig.1.7

SUB GRAPH

Let H be a graph with vertex set V(H) and edge set E(H) and, similarly, let G be a graph with vertex set V(G) and edge set E(G). Then we say that H is a <u>subgraph</u>of G if V(H) $\subseteq$ of V(G) and E(H) $\subseteq$ of E(G). In such a case, we also say that G is a supergraph of H.

For example, in Figure 1.8, G_1 is a subgraph of both G_2 and G_3 but G_3 is not a subgraph of G_2.

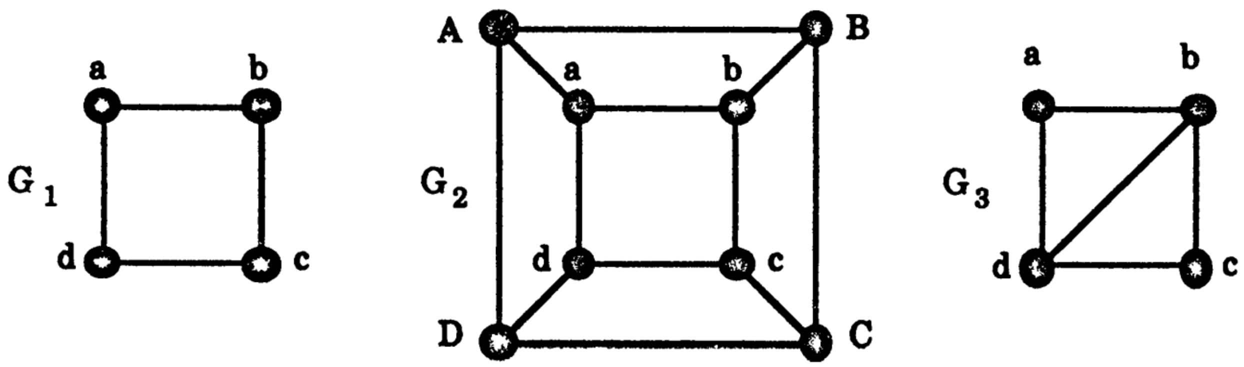


Fig1.8 $G_1 \subseteq G2$, $G_1 \subseteq G_3$ but $G_3 \not\subseteq G_2$

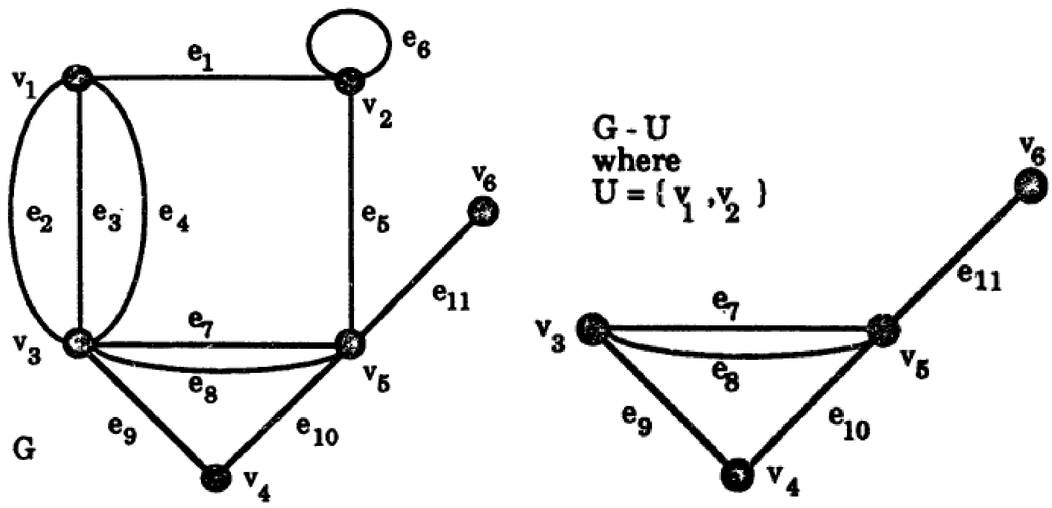
Any graph isomorphic to a subgraph of G is also referred to as a subgraph of G. If H is a subgraph of G then we write H$\subseteq$G. When H$\subseteq$G but H$\neq$G, i.e., V(H) $\neq$V(G) or E(H)$\neq$E(G), then H is called a proper subgraph of G. A spanning subgraph (or spanning supergraph) of G is a subgraph (or supergraph) H with

V(H)= V(G), i.e., H and G have exactly the same vertex set. In fig.1.8 G_1 is a proper spanning subgraph of G_3.

If G = (V, E) and U is a proper subset of V then G-U denotes the subgraph of G with vertex set V-U and whose edges are all those of G which are not incident with any vertex in U.

If F ⊆ of the edge set E then G-F denotes the subgraph of G with vertex set V and edge set E-F, i.e., obtained by deleting all the edges in F, but not their endpoints.

G-U and G-Fare also referred to as vertex deleted and edge deleted subgraphs (respectively).



By deleting from a graph G all loops and in each collection of parallel edges all edges but one in the collection we obtain a simple spanning subgraph of G, called the underlying simple graph of G.

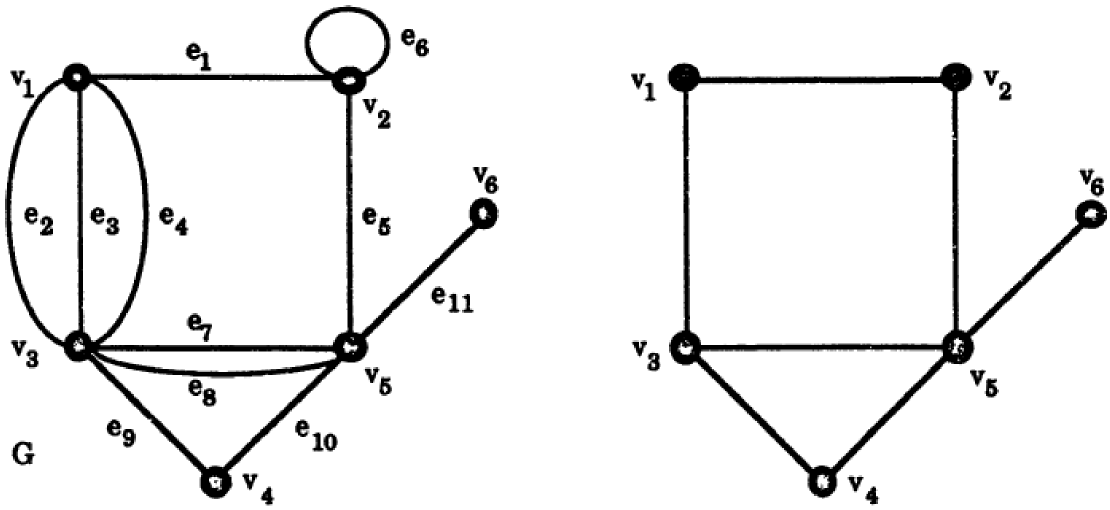


Fig.1.10 A graph and its underlying simple graph

If U is a nonempty subset of the vertex set V of the graph G then the subgraph G(U) of G induced by U is defined to be the graph having vertex set U and edge set consisting of those edges of G that have both ends in U.

Similarly if F is a nonempty subset of the edge set E of G then the subgraph G[F] of G induced by F is the graph whose vertex set is the set of ends of edges in F and whose edge set is F.

For the graph Gin fig.1.10 taking $U = \{v_2, v_3, v_5)$ and $F = \{e_1, e_3, e_5, e_7, e_9\}$ we get G[U] and G[F]

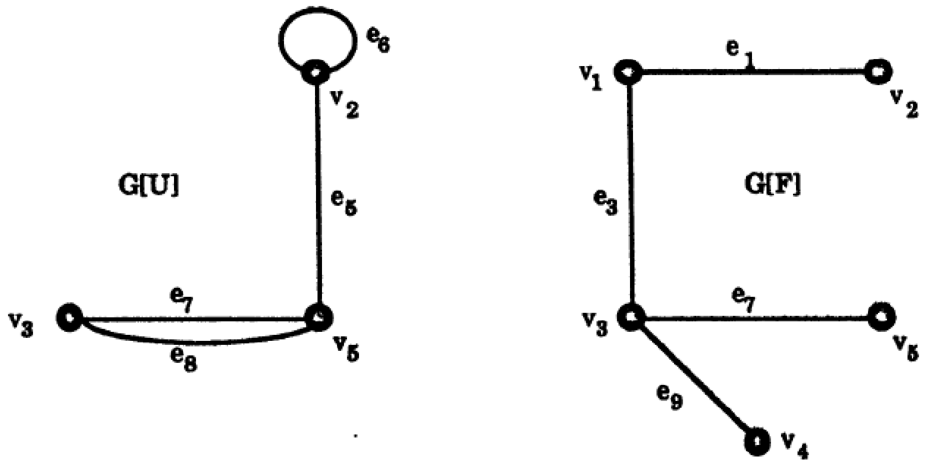


Fig: 1.11G[U] and G[F] for $U = \{v_2,v_3,v_5)$ and $F = \{e_1,e_3,e_5,e_7,e_9\}$.

Two subgraphs G_1, and G_2 of a graph G are said to be <u>disjoint</u> if they have no vertex in common, and <u>edge disjoint</u> if they have no edge in common.

Given two subgraphs G_1 and G_2 of G, the union $G_l U G_2$, is the subgraph of G with vertex set consisting of all those vertices which are in either G_1 or G_2 (or both) and with edge set consisting of all those edges which are in either G_1, or G_2 (or both); symbolically

$V(G_1 U G_2) = V(G_1) U V(G_2)$,

$E(G_1 U G_2) = E(G_1) U E(G_2)$.

For example, Figure 1.12 shows G[U] U G[F] for the subgraphs G[U] and G[F] of Figure 1.11.

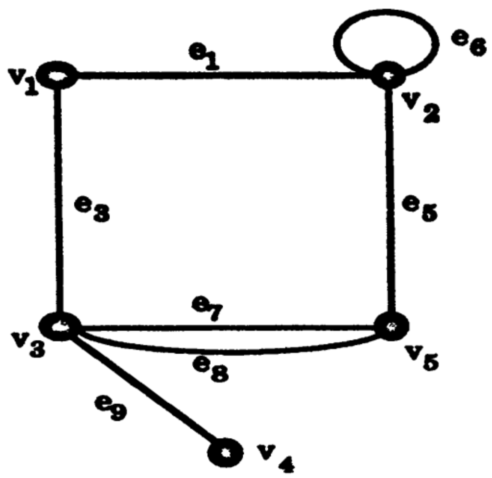


Fig.1.12: G[U]UG[F]

If G_1 and G_2 are two subgraphs of G with at least one vertex in common then the intersection $G_1 \cap G_2$ is given by

$V(G_1 \cap G_2) = V(G_1) \cap V(G_2)$,

$E(G_1 \cap G_2) = E(G_1) \cap E(G_2)$.

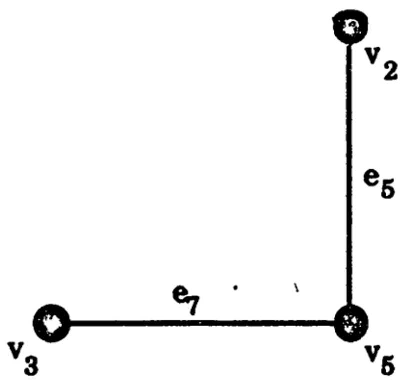


Fig1.13: G[U]∩G[F]

THE FIRST THEOREM OF GRAPH THEORY/HAND SHAKING THEOREM:

For any graph G with e edges and n vertices, $V_1, V_2, \ldots\ldots, V_n$,

$$\sum_{i=1}^{n} d(v_i) = 2e$$

COROLLARY:

In any graph G there is an even number of odd vertices.

$\sum_{u\in U} d(u) + \sum_{w\in W} d(w) = \sum_{v\in V} d(v) = 2e,$

$$\sum_{w\in W} d(w) = 2e - \sum_{u\in U} d(u)$$

FUSION

Let u and v be distinct vertices of a graph G. We can construct a new graph G_1 by fusing (or identifying) the two vertices, namely by replacing them by a single new vertex x such that every edge that was incident with either u or v in G is now incident with x, i.e., the end u and the end v become end x.

Thus the new graph G_1 has one less vertex than G but the same number of edges as G and the degree of the new vertex x is the sum of the degrees of u and v. We illustrate the process in Figure 1.14.

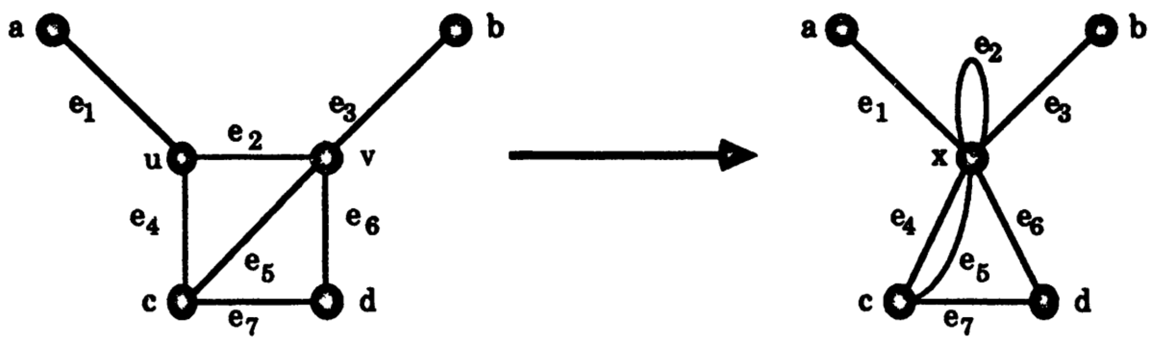


Fig.1.14: The fusion of vertices u and v

If v_i is fused to its neighbour v_j to form the new vertex w then, since each edge of the form v_iv_k or of the form v_iv_k gets changed to one of the form wv_k, it follows that in the adjacency matrix of the new graph G the entries in the row (and column) corresponding to w are just the sum of the corresponding entries given by v_i and $v_j$in the adjacency matrix for G.

One can more precisely describe this as the following two step process:

Step 1: Change u's row to the sum of u's row with v's row and (symmetrically) change u's column to the sum of u's column with v's column.

Step 2: Delete the row and column corresponding to v. The resulting matrix is the adjacency matrix of the new graph G_1.

WALKS AND PATH

A walk in a graph G is a finite sequence

W= $v_0e_1v_1e_2v_2$……………………$v_{K-1}e_kv_k$

whose terms are alternately vertices and edges such that, for $1 \leq i \leq k$, the edge e; has ends V_{i-1} and v_i. Thus each edge e_i is immediately preceded and succeeded by the two vertices with which it is incident.

We say that the above walk W is a v_0 - v_k walk or a walk from v_0 to v_k. The vertex $v_0$is called the origin of the walk W, while v_k is called the terminus of

W. Note that v_0 and v_k need not be distinct. The vertices $v_1,...,v_{k-1}$ in the above walk W are called its internal vertices. The integer k, the number of edges in the walk, is called the length of W. Note that in a walk there may be repetition of vertices and edges.

A trivial walk is one containing no edges.Thus, for any vertex v of G, W = v gives a trivial walk. It has length 0.

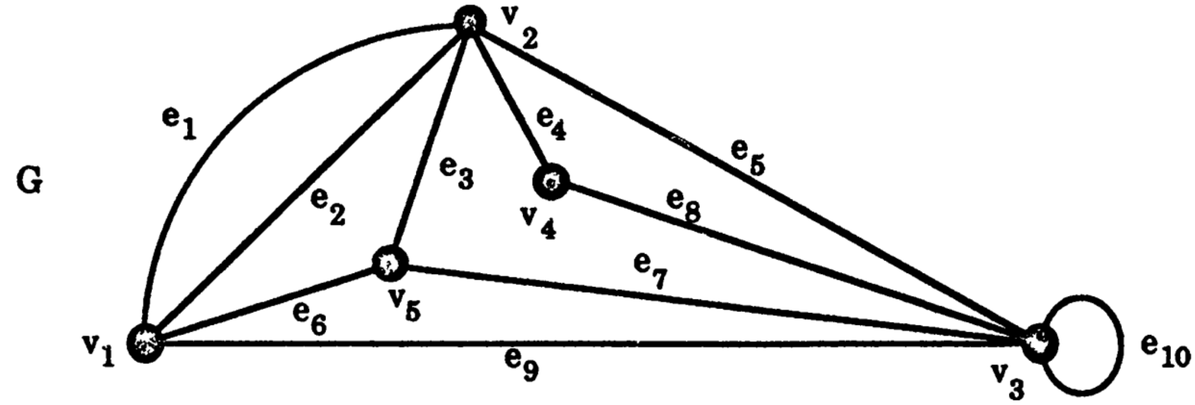


Fig.1.15

In Figure 1.15, W1 = v_1 e_1 v_2 e_5 v_3 e_{10} v_3 e_5 v_2 e_3 v_5 and W2 = v_1 e_1 v_2 e_1 v_1 e_1 v_2 are both walks, of length 5 and 3 respectively, from v_1 to v_5 and from v_1 to v_2 respectively.

Given two vertices u and v of a graph G, a u- v walk is called closed or open depending on whether u = v or $u \neq v$.

If the edges $e_1,e_2,...e_k$ of the walk $W=v_0e_1v_1e_2v_2.....e_kv_k$ then W is called a trail. In other words, a trail is a walk in which no edge is repeated.

If the vertices $v_0,v_1,...,v_k$ of the walk $W=v_0e_1v_1e_2v_2...e_kv_k$ are distinct then W is called a path. Clearly any two paths with the same number of vertices are isomorphic. A path with n vertices will sometimes be denoted by P_r. Note that P_n has length n - 1.

In other words, a path is a walk in which no vertex is repeated. Thus in a path no edge can be repeated either, so every path is a trail. Not every trail is a path.

THEOREM 2: **Given any two vertices u and v of a graph G, every u-v walk contains a u-v path, i.e., given any walk**

$$W = ue_1v_1.........v_{k-1}e_kv$$

then, after some deletion of vertices and edges if necessary, we can find a subsequence P of W which is a u-v path.

PROOF: **If u = v, i.e., if W is closed, then the trivial path P= u will do.**

Now suppose $u \neq v$, i.e., W is open and let the vertices of W be given, in order, by

$U = u_0, u_1, u_2, ..., u_{k-1}, u_k = v$.

If none of the vertices of G occurs in W more than once then W is already a u-v path and so we are finished by taking P=W.

So now suppose that there are vertices of G that occur in W twice or more. Then there are distinct i, j, with i <j, say, such that $u_i = u_j$. If the terms $u_i, u_{j+1}, ..., u_{j-2}$ (and the preceding edges) are deleted from W then we obtain a u-v walk W, having fewer vertices than W. If there is no repetition of vertices in W_1, then W_1 is a U-V path and setting P = W_1 finishes the proof.

If this is not the case, then we repeat the above deletion procedure until finally arriving at a u-v walk that is a path, as required.

DEFINITION

A vertex u is said to be connected to a vertex v in a graph G if there is a path in G from u to v.A graph G is called <u>connected</u> if every two of its vertices are connected. A graph that is not connected is called <u>disconnected</u>.

Given any vertex u of a graph G, let C(u) denote the set of all vertices in G that are connected to u. Then the subgraph of G induced by C(u) is called the connected component containing u, or simply the component containing u.

The graph G of Figure 1.16 has 6 components.

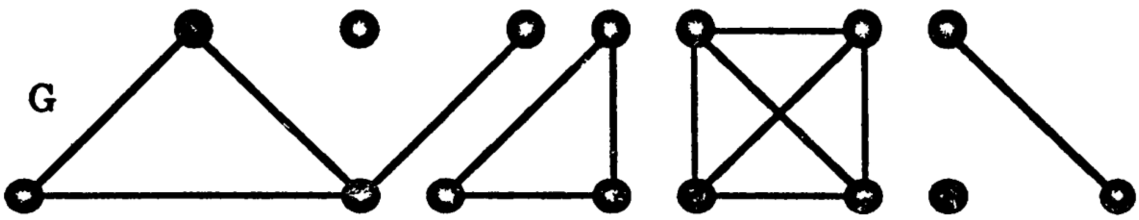


Fig.1.16: A graph with six connected components

A nontrivial closed trail in a graph G is called a cycle if its origin and internal vertices are distinct. In detail, the closed trail C = $v_1v_2...v_nv_1$ is a cycle if

1. **C has at least 1 edge and $v_1,v_2,...,v_n$ are n distinct vertices.**

A cycle of length k, i.e., with k edges, is called a k-cycle. A k-cycle is called odd or even depending on whether k is odd or even.

A 3-cycle is often called a <u>triangle</u>.

Clearly any two cycles of the same length are isomorphic. An n-cycle, i.e., a cycle with n vertices, will sometimes be denoted by C_n.

For example in Figure 1.17, C = $v_1\ v_2\ v_3\ v_4\ v_1$ is a 4-cycle, T = $v_1\ v_2\ v_5\ v_3\ v_4\ v_5\ v_1$ is a nontrivial closed trail which is not a cycle (because v_5 occurs twice as an internal vertex), and C" = $v_1\ v_2\ v_5\ v_1$ is a triangle.

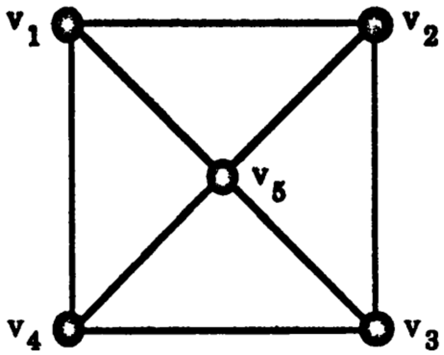


Fig.1.17

TREES AND CONNECTIVITY

TREE

A connected graph without cycles is defined as a tree. A graph without cycles is called an acyclic graph. In other words, a graph is called a tree if it is a connected acyclic graph.

A tree is an undirected graph G that satisfies any of the following equivalent conditions:

- **G is connected and acyclic (contains no cycles).**
- **G is acyclic, and a simple cycle is formed if any edge is added to G.**
- **G is connected, but would become disconnected if any single edge is removed from G.**
- **G is connected and the 3-vertex complete graph K_3 is not a minor of G.**
- **Any two vertices in G can be connected by a unique simple path.**

If G has finitely many vertices, say n of them, then the above statements are also equivalent to any of the following conditions:

- **G is connected and has n − 1 edges.**
- **G is connected, and every subgraphof G includes at least one vertex with zero or one incident edges. (That is, G is connected and 1-degenerate.)**
- **G has no simple cycles and has n − 1 edges.**

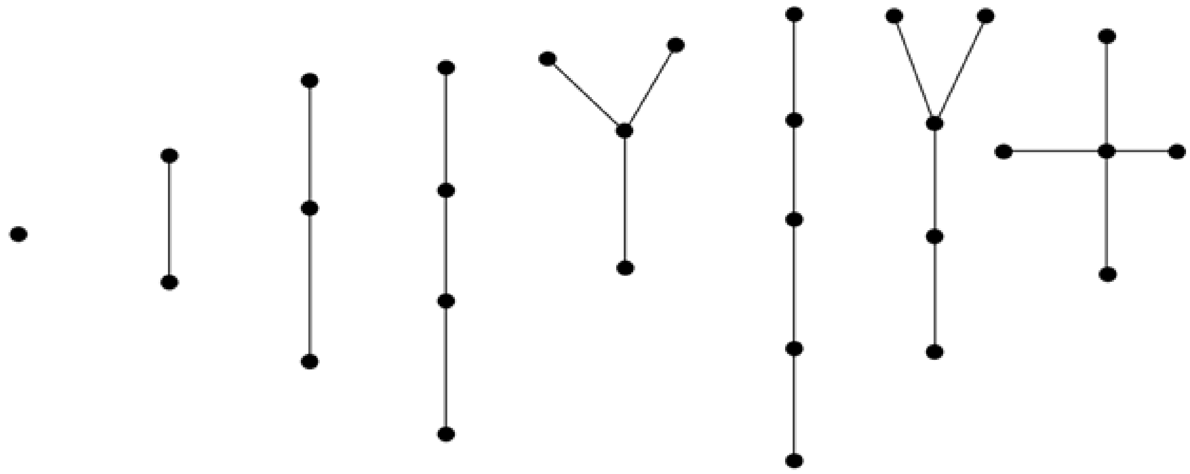


Fig.2.1

In the above example, all are trees with fewer than 6 vertices.

THEOREM 3: **In a tree, any two vertices are connected by a unique path.**

PROOF: **Suppose that the result is false. Let G be a tree and assume that there are two distinct (u,v)- paths P1 and P2in G . Since P1 and P2are not equal, there is an edge e=xy of P1 that is not an edge of P2. Clearly the graph (P1 U P2) – e is connected. It therefore contains an (x,y)-path P. But then P + e is a cycle in the acyclic graph G, which is a contradiction.**

THEOREM 4: **If in a graph G, there is one and only one path between every pair of vertices, then graph G is a tree.**

PROOF: **There is the existence of a path between every pair of vertices so we assume that graph G is connected. A circuit in a graph implies that there is at least one pair of vertices a and b, such that there are two distinct paths between a and b. Since G has one and only one path between every pair of vertices. G cannot have any circuit. Hence graph G is a tree.**

THEOREM 5: If T is a tree with n vertices then it has precisely n-1 degree.

PROOF: Let n be the number of vertices in a tree (T).
If n=1, then the number of edges=0.
If n=2, then the number of edges=1.
If n=3, then the number of edges=2.

Hence, the statement (or result) is true for n=1, 2, 3.

Let the statement be true for n=m. Now we want to prove that it is true for n=m+1.

Let e be the edge connecting vertices say V_i and V_j. Since G is a tree, then there exists only one path between vertices V_i and V_j. Hence if we delete edge e it will be disconnected graph into two components G_1 and G_2 say. These components have less than m+1 vertices and there is no circuit and hence each component G_1 and G_2 have m_1 and m_2 vertices.

Now the total number of edges = $(m_1-1) + (m_2-1) + 1$

$= (m_1 + m_2) - 1$

$= m + 1 - 1$

$= m$.

Hence for n=m+1 vertices there are m edges in a tree (T). By the mathematical induction the graph exactly has n-1 edges. .

BRIDGE

An edge 'e' of a graph G is called a bridge if the subgraph G-e has more connected components than G has. In graph theory, a bridge is an edge of a graph whose deletion increases its number of connected components. Equivalently, an edge is a bridge if and only if it is not contained in any cycle. A graph is said to be bridgeless if it contains no bridges.

Following are some example graphs with bridges highlighted with red colour.

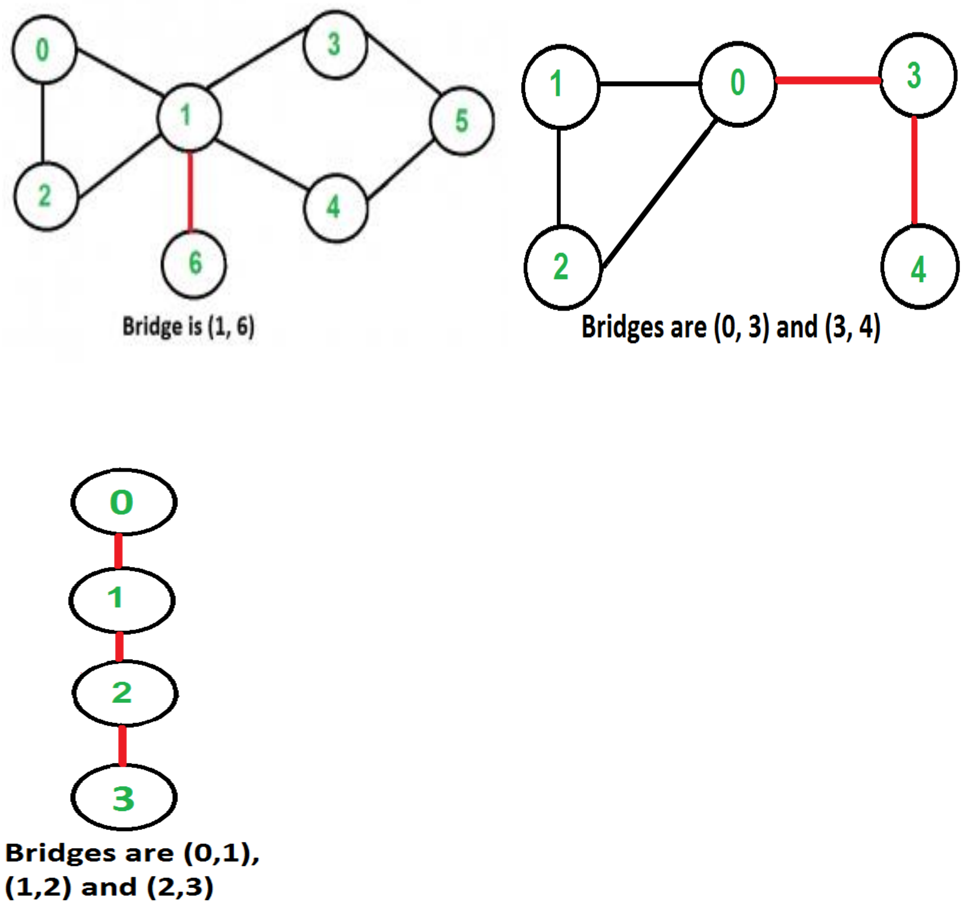


Fig.2.2

SPANNING TREE

Let G be a graph. The subgraph H of G is called a spanning subgraph of G if the vertex set of H is the same as the vertex set of G. A spanning tree of a graph G is a spanning subgraph of G that is a tree.

A graph G is connected if and only if it has a spanning tree.

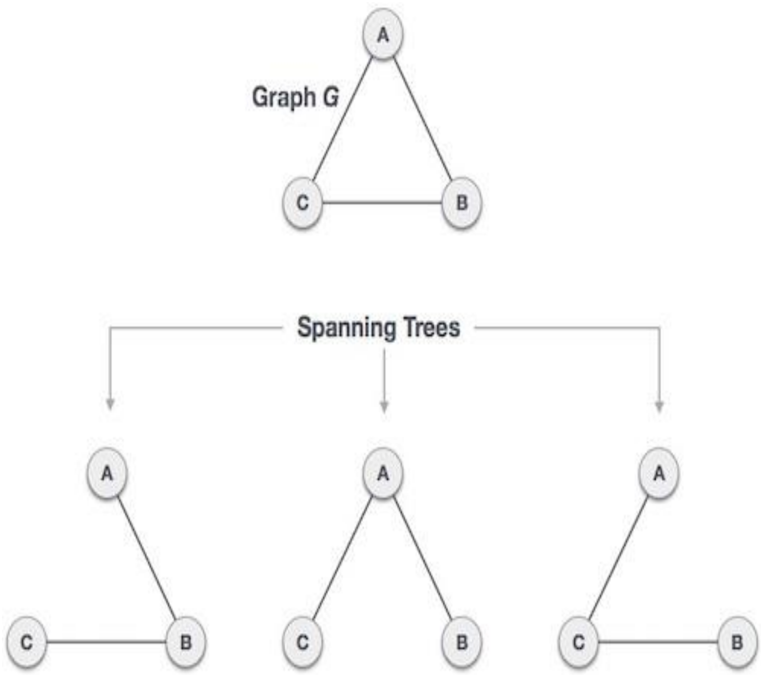

Graph G
A
C
B
Spanning Trees
A
C
B
A
C
B
A
C
B

Fig2.3

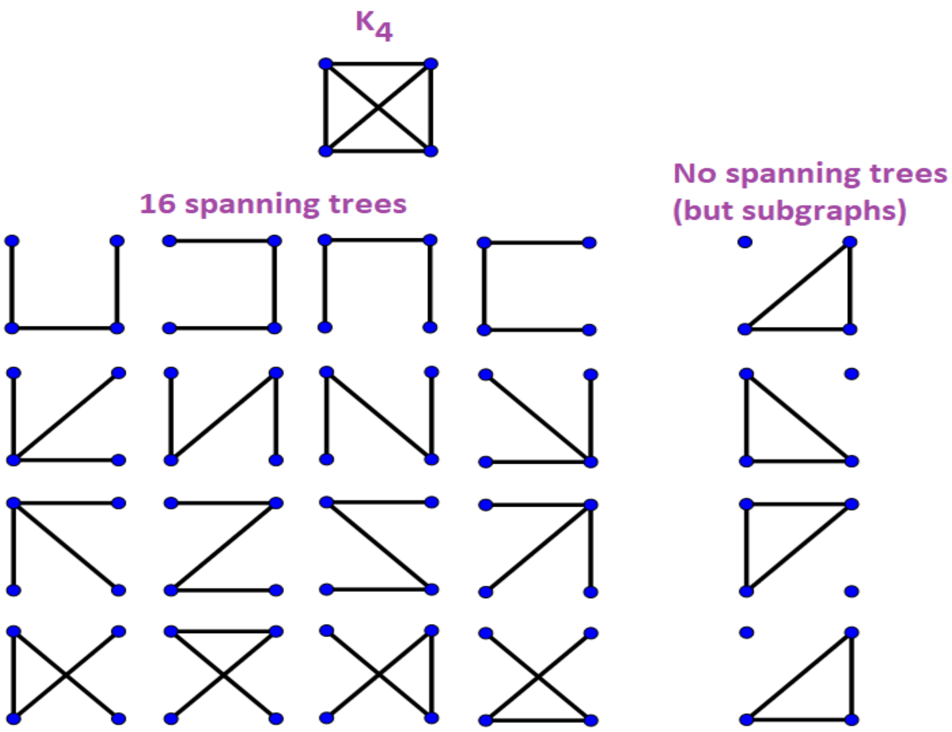

K4
16 spanning trees
No spanning trees
(but subgraphs)

Fig.2.4

CONNECTOR PROBLEMS

The problem in graph theory to find the minimum total length of a connected graph are called connector problems. Kruskal's algorithm and Prim's algorithm are two standard approaches to solving this problem.

Suppose certain villages in an area are to be joined to a water supply situated in one of the villages. The system of pipes is to consist of pipelines connecting the water towers of two villages. For any two villages we know how much it would cost to build a pipeline connecting them, provided such a pipeline can be built at all. This is an example of a connector problem and we solve it using spanning trees and the concept of weighted graph.

A weighted graph is a graph G in which each edge *e* has been assigned a real number W(e), called the weight (or length) of e. If H is a subgraph of weighted graph, the weight W(H)of His the sum of weights $W(e_1)+.....+W(e_k)$where $\{e_1,...e_k\}$ is the set of edges of H.

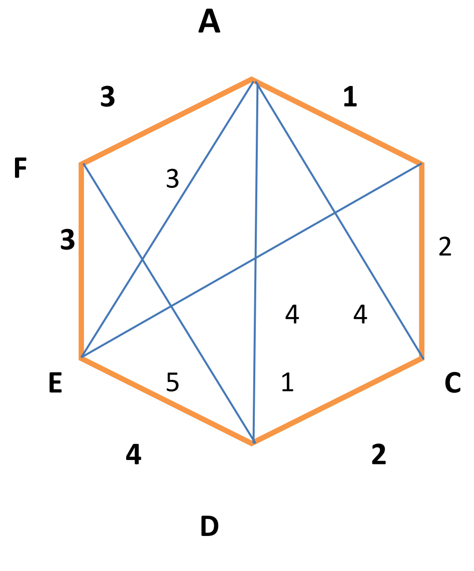


Fig.2.5: A weighted graph

In the above figure, A, B, C, D, E and F are the vertices representing villages and the lack of an edge from B to D (for example) indicates that it is not possible to build a pipeline from B to D. The number (weight) 4 assigned to

the edge from A to C indicates the cost of building a pipeline from A to C (for example).

SHORTEST PATH PROBLEMS

In graph theory, the shortest path problem is the problem of finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized.

The problem of finding the shortest path between two intersections on a road map may be modeled as a special case of the shortest path problem in graphs, where the vertices correspond to intersections and the edges correspond to road segments, each weighted by the length of the segment.

The most important algorithms for solving this problems are:

- **Dijkstra's Algorithm**
  This algorithm solves the single-source shortest path problem with non-negative edge weight. For a given source node in the graph, Dijkstra's Algorithm finds the shortest path between that node and every other.It can also be used for finding the shortest paths from a single node to a single destination node by stopping the algorithm once the shortest path to the destination node has been determined. Dijkstra's algorithm can be used to find the shortest route between one city and all other cities.

- **Breadth First Search (BFS) Algorithm**

Breadth-first search (BFS) is an algorithm for traversing or searching tree or graph data structures. Graph traversal means visiting every vertex and edge exactly once in a well-defined order.BFS is a traversing algorithm where we start traversing from a selected node (source or starting node) and traverse the graph layerwise thus exploring the neighbour nodes (nodes which are directly connected to source node) and then move towards the next-level neighbour nodes. Breadth-first search can be used to solve many problems in graph theory including copying garbage collection, Cheney's algorithm and

also finding the shortest path between two nodes u and v, with path length measured by number of edges.

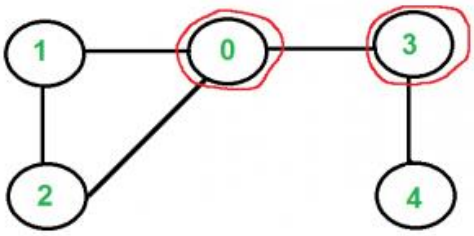
CUT VERTICES AND CONNECTIVITY

The number of component of a graph G is denoted by ω(G). A vertex v of a graph G is called a cut vertex (or articulation point) of Gif ω(G-v)>ω(G).

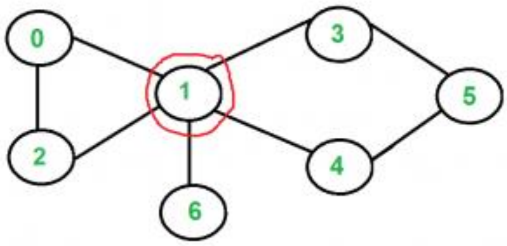
Let 'G' be a connected graph. A vertex V ∈ G is called a cut vertex of 'G', if 'G-V' (Delete 'V' from 'G') results in a disconnected graph. Removing a cut vertex from a graph breaks it in to two or more graphs.

Articulation points represent vulnerabilities in a connected network – single points whose failure would split the network into 2 or more disconnected components. They are useful for designing reliable networks.
For a disconnected undirected graph, an articulation point is a vertex removing which increases number of connected components.

Following are some example graphs with articulation points encircled with red colour.



Articulation points are 0 and 3



Articulation Point is 1

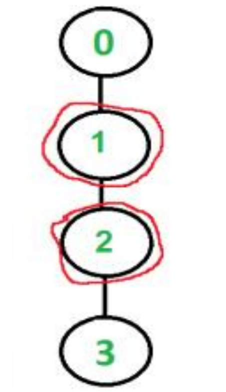


Fig.2.6

Let G be a simple graph. The (vertex) connectivity of G, denoted by k(G), is the smallest number of vertices in G whose deletion from G leaves either a disconnected graph or K_1.

A simple graph G is called n-connected (where n>=1) if k(G)>= n.

Letu and v be two vertices of a graph G. A collection {$P_{(1)}$,…..,$P_{(n)}$} of u-v paths is said to be internally disjoint if, given any distinct pair $P_{(i)}$and $P_{(j)}$in the collection, u and v are the only vertices $P_{(i)}$and $P_{(j)}$have in common.

GRAPH COLOURING

In graph theory, graph colouring is a special case of graph labelling; it an assignment of labels traditionally called colours to elements of a graph subjects to certain constraints. Actual colours have nothing at all to do with this, graph colouring is used to solve problems where you have a limited amount of resources or other restrictions. The colours are just an abstraction for whatever resource you're trying to optimize, and the graph is an abstraction of your problem.

VERTEX COLOURING

Let G be graph. A(vertex)colouring of G assigns colours, usually denoted by 1,2,3,………,to the vertices of G, one colour per vertex, so that adjacent vertices are assigned different colours.

A k-colouring of G is a colouring which consists of k different colours and in this case G is said to be k-colourable.

The, minimum number n for which there is an n-colouring of the graph G is called the chromatic index (or chromatic number) of G and is denoted by χ(G). If χ(G)=k we say that G is k-chromatic.

THEOREM 6: **Let G be a nonempty graph. Then χ(G) = 2 if and only if G is bipartite.**

PROOF: Let G be bipartition V=X∪Y. Assigning colour 1to all vertices in X and colours 2 to all vertices in Y gives a 2-colouring for G and so, since G is nonempty, χ(G) = 2.

Conversely, suppose that χ(G) = 2. Then g has a 2-colouring. Denote by X the set of all those vertices coloured 1 and by Y the set of all those vertices coloured 2. Then no two vertices in X are adjacent and similarly for Y. Thus any edge in G must join a vertex in X and a vertex in Y. Hence G is bipartite with bipartition V=X∪Y.

NOTE: **For a graph G we let $\Delta(G)=\max\{d(v) : v \text{ is a vertex of } G\}$. Thus $\Delta(G)$ is the maximum vertex degree of G.**

THEOREM 7: **For any graph G, $\chi(G) \leq \Delta(G)+1$.**

PROOF: **We use induction on n, the number of vertices in G. The theorem is clearly true for n=1 since here $G=K_1$, $\chi(G)=1$ and $\Delta(G)=0$.**

Now suppose that the result is true for all graphs with n-1 vertices, where n is a fixed integer greater than 1, and let G be some graph with n vertices. Choose a vertex v of G. Then the subgraph G-v has n-1 vertices and so, by the induction assumption, $\chi(G-v) \leq \Delta(G-v)+1$. This allows us to choose a vertex colouring of G-v involving $\chi(G-v)+1$ colours. Now our vertex v has at most $\Delta(G)$ neighbours in G and so these neighbhours use up at most $\Delta(G)$ colours in the colouring of G-v. Thus if $\Delta(G)= \Delta(G-v)$ there is at least one colour not used by v's neighbours and we can use such a colour for v, giving a $(\Delta(G)+1)$-colouring for G. On the other hand, if $\Delta(G)\neq \Delta(G-v)$ then $\Delta(G-v)$ then $\Delta(G-v)< \Delta(G)$ and simply colouring v with a brand new colour gives a $(\Delta(G-v)+2)$-colouring of G which is good enough since $\Delta(G-v)+2\leq \Delta(G)+1$. Hence, in both cases, we have, $\chi(G)\leq \Delta(G) +1$.

CRITICAL GRAPHS

A graph G is called k-critical if $\chi(G)=k$ and $\chi(G-v)<k$ for each vertex v of G.

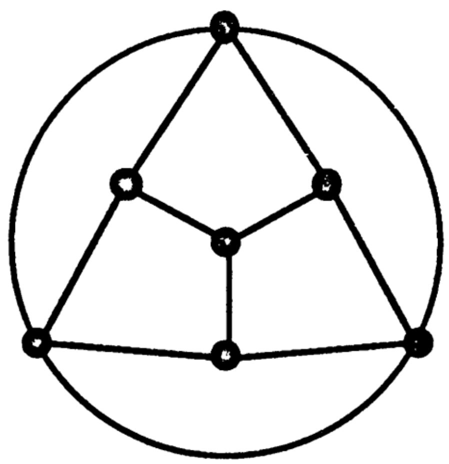


Fig.3.1: 4-critical graph

THEOREM 8: **Let G be a k-critical graph. Then**

- **The degree of every vertex of G is at least k-1,**
- **G has no pair of subgraph G_1 and G_2 for which $G=G_1 \cup G_2$ and $G_1 \cap G_2$ is a complete graph,**
- **G-v is connected for every vertex v of G(provided k>1).**

PROOF:

- **Suppose that v is a vertex of G with d(v) < k-1. Since G is k-critical, the subgraph G-v has a (k-1)-colouring. Since v has at most k-2 neighbours these neighbours do not use up all the colours in this (k-1)-colouring of G-v. By colouring v with one of these unused colours we extend the colouring of G-v to a (k-1)-colouring of G. this is a contradiction since χ(G)=k. Hence every vertex v has degree at least k-1, as required.**
- **Suppose that $G=G_1 \cup G_2$ whereG_1andG_2 are subgraph with $G_1 \cap G_2 =K_t$. Since G is k-critical, G_1 and G_2 both have chromatic index at most k-1. Choose a (k-1)-colouring for G_1 and for G_2. In the overlap $G_1 \cap G_2$, since this is complete, every vertex has a different colour (in each of the(k-1)-colourings). This enables us the rearrange the colours in the (k-1)-colourings of G_2 sop that it gives the same colour to each vertex in $G_1 \cap G_2$ as the colouring of G_1 gives. Combining the two colourings then produces a(k-1)-colouring of all of G. This is impossible since χ(G)=k. Thus no such subgraphs G_1,G_2 exist.**
- **Suppose that G-v is disconnected for some vertex v of G. Then G-v has subgraphs H_1 and H_2 with $H_1 \cup H_2$=G-v and $H_1 \cap H_2$ =V. Set G_1 andG_2 as the subgraphs of G induced by H_1 together with v and H_2 together with v respectively. Then $G=G_1 \cup G_2$and $G_1 \cap G_2 =K_1$(with v_k as the single vertex). This contradictions above and so G-v cannot be disconnected.**

CLIQUE

For any graph G a complete subgraph of G is called a clique of G. The number of vertices in a largest clique of G is called the clique number of Gand denoted by cl(G).

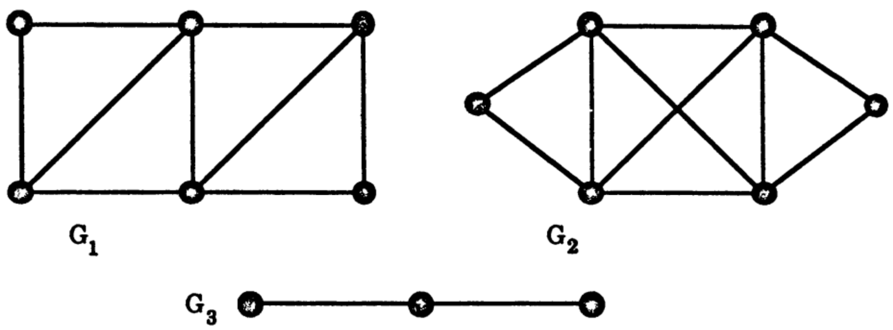


Fig.3.2

For example in fig.3.2, $cl(G_1)=3$, $cl(G_2)=4$ and $cl(G_3)=2$.

EDGE COLOURING

Let G be a nonempty graph. An edge colouring of G assigns colours, usually denoted by 1,2,3…., to the edge of G, one colour per edge, so that adjacent edges are assigned different colours.

A k-edge colouring of G is a colouring of G which consist of k different colours and in this case G is said to be k-edge colourable.

The minimum number n for which there is an n-colouring of G is called the edge chromatic number(or edge chromatic index) of G and is denoted by $\chi_1(G)$. If $\chi_1(G)=k$ we say that G is k-edge chromatic.

If H is a subgraph of G, then $\chi_1(H)\leq \chi_1(G)$.

Letting $\Delta(G)$ denote the maximum vertex degree of G as usual, we have

$\Delta(G) \leq \chi_1(G)$,

since if v is any vertex of G with $d(v)=\Delta(G)$ then the $\Delta(G)$ edges incident with v much each have a different colour in any edge colouring of G.

THEOREM 10: **Let $G=K_n$, the complete graph on n vertices, $n\geq2$.**

Then $\Delta(G)$ (=n-1) if n is even

$\chi_1(G)$ ={

$\Delta(G)+1$ (=n) if n is odd.

PROOF: We first suppose that n is odd. Draw G as usual so that its vertices form a regular polygon(with n edges on the perimeter having the same length).Colour the edges around the perimeter using a different colour for each edge. Now each of the remaining "internal" edges of *G* is parallel and we assign it the same colour as we have assigned to this perimeter edge.Fig.3.3 show K_7partially edge coloured in this way.

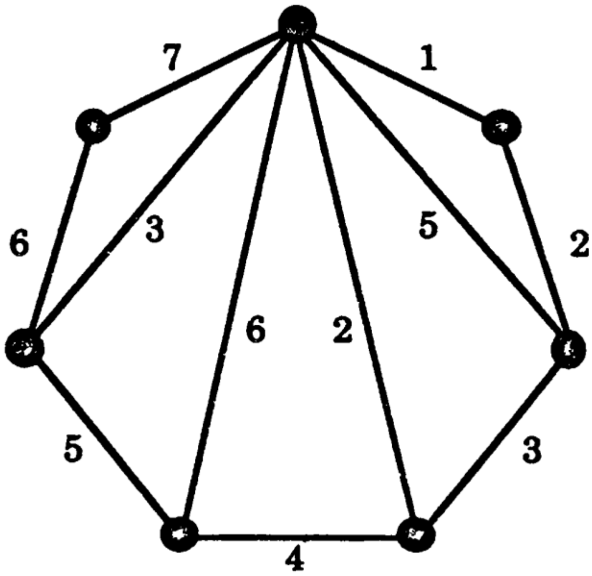


Fig.3.3: The beginnings of an edge colouring in K_7

Then two edges have the same colour only if they are parallel and from this it follows that we do have an edge colouring of *G*.Since it involves n colours we have shown thatχ_1(G) $\leq$ n (=Δ(G)+1).

Now suppose G has an (n-1)-colouring.Now,from the definition of an edge colouring, the edges of one particular colour from a matching in G and so,since n is odd, the maximum possible number of these is $\frac{1}{2}$(n-1). This implies that there at most (n-1)$\times$ $\frac{1}{2}$(n-1) edges in *G*, a contradiction sinceK_n has $\frac{1}{2}$n(n-1)edges.Hence G does not have an (n-1) –colouring and so χ_1(G)=n; as required.

We now deal; with the case when n is even. Let *v* be some fixed vertex of G. Then G-*v* is complete, with n-1 vertices. Since n-1 is odd we can give it an (n-1)-colouring,as described above.With this colouring there is a colour absent from each vertex,with different vertices having different absentees.G is reformed from *G-v* by joining each vertex w of *G-v* to *v* by an edge. Colour each such edge with the colour absent from w. This gives an (n-1)-colouring and since Δ(G) = n-1 we getχ_1(G) = Δ(G) =n-1, as required.

MAP COLOURING

A map is defined to be a plane connected graph with no bridges. A map G is said to be k- face colourable if we can colour in such a way that no two adjacent faces,i.e.,two faces sharing a common boundary edge,have the same colour.

The Four Colour Conjecture: **Every map is 4-face colourable.**

Theorem 11:(The Five Colour Theorem,Heawood)

The faces of a map can always be coloured with five or fewer colours.

Theorem 12: (The Four Colour Theorem)

Every map can be coloured in four or fewer colours.

CONCLUSION

This project has attempted to demonstrate some important concepts in Graph Theory. Graph Theory enables us to study and model networks and solve some difficult problems inherently capable of being modelled using networks. It is a very important and very vast topic with application spread across many fields.

REFERENCE

- **A FIRST LOOK AT GRAPH THEORY: John Clark and Derek Allan Holton,**
- **A TEXTBOOK OF GRAPH THEORY, SECOND EDITION: R. Balakrishnan and K. Ranganathan**
- **GRAPH THEORY WITH APPLICATIONS: A. Bondy and U. S. R. Murty**