Graph Theory & Its Applications

A Project

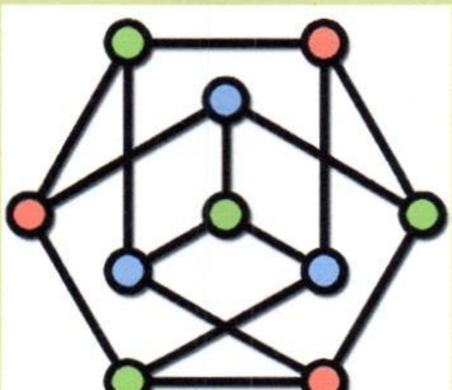
Submitted by

Department of Mathematics,

Pattamundai College, Pattamundai

March 2020.

Session 2019-20



Report

A project on "Graph Theory & Its Applications" was under taken by the students of Department of Mathematics under the guidance of Sri Arabinda Pandab, HOD Mathematics. It took two months (Feb & March 2020) to carried out the project. Konigsberg was a city in Russia situated on the Pregel river which serves as the residence of the dukes of Prussia in the 16th century. Today the city is known as Kaliningrad and is major industrial and commercial center of western Russia. The river Pregel flowed through the town dividing into four regions. In the eighteen century, seven bridges connected the four regions. Konigsberg people used to take long walk through town on Sunday. They wondered whether it was possible to start at one location in the town travel across all the bridges without crossing any bridge twice and return to the starting point. This problem was first solved by the prolific Swiss mathematician Leonhard Euler, who as a consequence of his solution invented the branch of mathematics now known as Graph Theory. Euler solution consisted of representing the problem by graph with four regions represented by four vertices and the seven bridges as seven edges.

The students found some results of Graph Theory like Sperner's lemma, Euler's tour and Hamilton cycle, Matrix representation of graphs, Incedence matrix, Circuit matrix. Also they got knowledge about applications of graph like The shortest path problem and Dijkstra's algorithm, The Chinese postman problem and Fleury's algorithm, The Travelling salesman problem, Application to switching network, The connector problem and Kruskal's algorithm.

Finally, the project was completed and submitted on 12th March 2020.

12/3/20
Arabinda Pandab
HOD, Mathematics

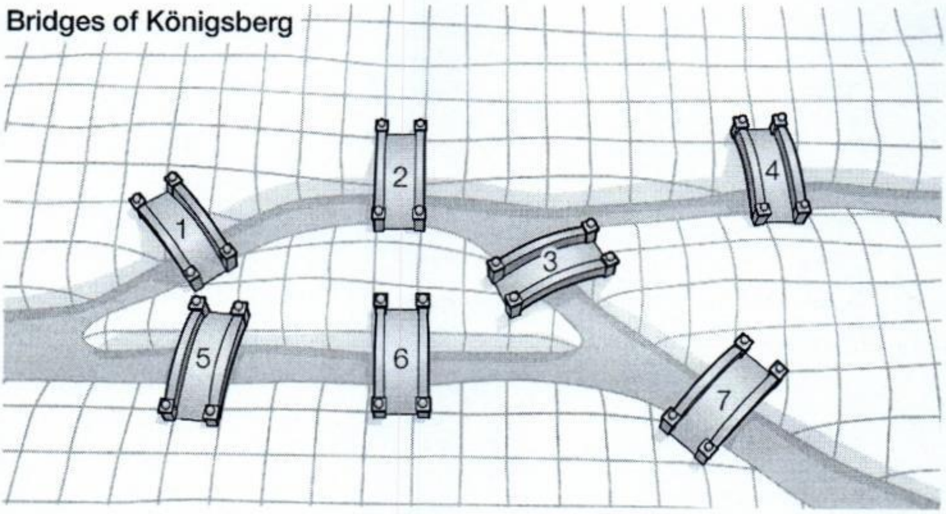
Contents

Chapter 1

INTRODUCTION

1.1 Introduction

Konigsberg was a city in Russia situated on the Pregel river which serves as the residence of the dukes of Prussia in the 16th century. Today the city is known as Kaliningrad and is major industrial and commercial center of western Russia. The river Pregel flowed through the town dividing into four region,as in the followed picture.



In the eighteen century, seven bridges connected the four region. Konigsberg people used to take long walk through town on Sunday. They wondered whether it was possible to start at one location in the town travel across all the bridges without crossing any bridge twice and return to the starting point. This problem was first solved by the prolific swiss mathematician Leonhard Euler, who as a consequence of his solution invented the branch of mathematics now known as Graph Theory. Euler solution consisted of representing the problem by graph with four regions represented by four vertices and the seven bridges as seven edges.

Chapter 2

SOME RESULTS ON GRAPH THEORY

2.1 Some results on Graph theory

Definition 2.1.1. *A graph G(V,E) is the order pair consist of a set of objects* $V = \{v_1, v_2, v_3, ...\}$ *called vertices and another set* $E = \{e_1, e_2, e_3, ...\}$ *called edges such that each edge* e_k *is identified with an unordered pair* (v_i, v_j) *of vertices. The vertices* v_i, v_j *associated with edge* e_k *are called the end vertices of* e_k.

The graph that has neither a self loop nor a parallel edges is called simple graph.

Definition 2.1.2. *The degree of a vertex v in a graph G is defined by "deg v" is the number of edges which are incident on v. Each loop on a vertex v contribute 2 to the degree of v.*

Theorem 2.1.3. *(Handshaking Theorem) The sum of the degree of the vertex of graph is equal to twice of the number of edges i.e.,* $\sum_{v \in V} deg(v) = 2|E|$.

Theorem 2.1.4. *In a graph the number of odd degree vertices is always even.*

Definition 2.1.5. *With each edge e of G let there be associated a real number W(e) called it's weight. Then G together with these weight on it's edge is called as weighted graph.*

Definition 2.1.6. *A simple graph G is said to be complete graph if every vertex of G is connected with every other vertex of G.*

Definition 2.1.7. *Let G and H are two graphs with vertex set V(G) and V(H) and edge set E(G) and E(H) respectively such that* $V(H) \subseteq V(G)$ *and* $E(H) \subseteq E(G)$ *then, we say H is subgraph of G.*

If V(H)=V(G) but $E(H) \subseteq E(G)$ then, we say H is a spanning subgraph of G. If H is a subgraph of a weighted graph, the weight W(H) of

H is the sum of the weight $\sum_{e \in E(H)} W(e)$ on it's edges. Let G(V,E) be any graph then a walk in G is a finite alternating sequence of vertices and edges $v_0, e_1, v_1, e_2, ..., v_{n-1}, e_n, v_n$ such that v_{k-1} and v_k are end vertices of edge e_k, $1 \leq k \leq n$. If the end vertices of walk are distinct then it is called open walk. Every open walk in which no vertex appear more then one is called a path.

2.2 Sperner's lemma

Every continious mapping f of a closed n-disk to itself has a fixed point(i.e. a point x such that f(x)=x). This powerful theorem is known as *Brouwer's fixed point theorem.*

Sperner's lemma concerns the decomposition of a simplex (line segment,triangle,tetrahedron) into simplice. For the shake of simplicity we shall deal the two dimensional case.

Let T be a closed triangle in the plane. A subdivision of T in to a finite number of smaller triangle is said to be simplical if any two intersecting triangles have either a vertex or a whole side common as shown in fig-1.

Suppose that a simplical subdivision of T is given, then, a labeling of vertices of triangle in the subdivision in three symbols 0, 1 and 2 is said to be proper if,

1. The three vertices of T are labeled 0,1 and 2 (in any order) and
2. For $0 \leq i \leq j \leq 2$ each vertex on the side of T joining vertices labeled i and j is labelled either i or j.

We call a triangle in the subdivision which receive all three labels a distinguished triangle. The proper labeling in fig 1(b) has three distinguished triangle.

Theorem 2.2.1. *Sperner's Lemma:*

Every proper labeled simplical subdivision has a triangle has an odd number of distinguished triangle.

Proof. Let T_0 denote the region outside T and $T_1, T_2, ..., T_n$ be the triangle of the subdivision. Construct a graph on the vertex set $\{v_0, v_1, ..., v_n\}$ by

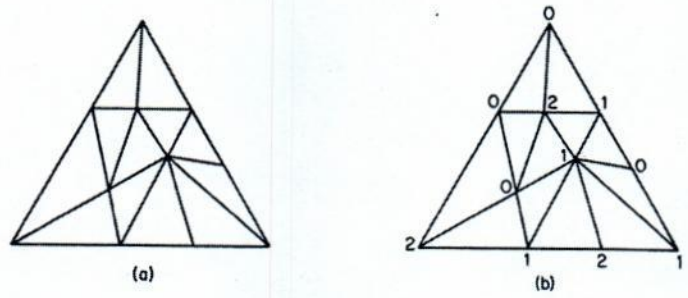
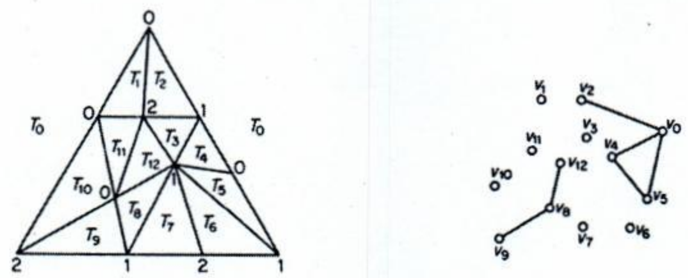


Figure 2.1: (a) A simplical subdivision of a triangle (b)A propper labelling of the subdivision

joining v_i and v_j, when ever the common boundary of T_i and T_j is an edge with labels 0 and 1. In this graph v_0 is clearly of odd degree as an odd



number of the vertices $v_1, v_2, ..., v_n$ are of odd degree. Now we have seen that none of these vertices can have degree three and so those with odd degree must have degree one. But a vertex v_i is of degree one iff the triangle T_i is distinguished. □

2.3 Euler's tour and Hamilton cycle

A trail that traverses every edge of G is called an Euler trail/path of G, because Euler was first to investigate the existence of such path in graph. A tour of G is closed walk that traverses each edge of G atleast once. An Euler tour is a tour which traverses each edge exactly once. A graph is said to be *Eulerian* if it contains an Euler tour. An edge is said to be cut edge or cut arc or bridge if deletion of that edge disconnects the graph G.

Theorem 2.3.1. *A non empty connected graph is Eulerian if and only if it has no vertices of odd degree.*

Proof. Let G be a connected Euler graph, then we have to prove that all the vertices of G are of even degree. Since G is an Euler graph, G contains an Euler circuit starting from a vertex v_1 in G. Suppose the circuit traversing all the edges of G is $v_1e_1v_2e_2...v_ne_nv_{n+1}$ where $v_n = v_{n+1}$. In this Euler circuit all the edges are distinct but some of the vertices may be repeated. It is clear that the pair of successive edges e_k and e_{k+1}, $1 \leq k \leq n-1$ contribute 2 to the degree of the vertex v_{k+1}. Therefore the vertices $v_2, v_3, ..., v_n$ are of even degree. Besides the vertex v_1 gets a contribution 2 its degree from its initial and final edge e_1 and e_n. Thus all the vertices are of even degree.

Suppose all the vertices of a connected graph G is even degree. Then we have to prove G is an Euler graph. Let v be any vertex of G. Construct a walk starting at v and going through the edge of G such that no edge is traced more then once. Since v is of even degree we shall eventually arrive at v when the taceing comes to end. So it is a closed walk say (K). If this K contain all the edges of G then G is an Euler graph. If K doesn't contain all the edges of G then we delete all the edges of K from G to obtain a new sub graph $K' = G \setminus K$. Since all the vertices of G and K are of even degree it follows that the vertices of K' are also even degree. Since G is connected K' must touch K atleast one vertex say 'u'. We again construct a new walk in the graph K' starting with u and this walk must terminate at vertex u. As all the vertices of K' must even degree . Now this walk in K' can be combined with K to form a new walk with start and end at vertex v. This argument can be repeated until we obtain an Euler circuit which traverses all the edges of G. Hence G is Euler graph. □

Theorem 2.3.2. *A connected multigraph G is Semieulerian if and only if there are exactly two vertices of odd degree.*

Definition 2.3.3. A circuit which traverses each vertex exactly once is called Hamilton circuit. A graph is said to be Hamiltonian if it contain atleast one Hamilton circuit. A path that contains every vertex of G is called a Hamilton path of G. Similarly a cycle that contain every vertex of G is called Hamilton cycle.

Theorem 2.3.4. *If G is Eulerian, then any walk in G constructed by Fleury's algorithm is an Euler path of G.*

Proof. Let G be Eulerian, and let $W_n = v_0 e_1 v_1, ..., e_n v_n$ be a trail in G constructed by Fleury's algorithm. Clearly, the end vertex v_n must be of degree zero in G_n. It follows that $v_n = v_0$; in other words, W_n is closed walk.

Suppose, now, that W_n is not an Euler tour of G, and let S be the set of vertices of positive degree in G_n. Then S is nonempty and $v_n \in \bar{S}$, where $\bar{S} = V \setminus S$. Let m be the largest integer such that $v_m \in S$ and $v_{m+1} \in \bar{S}$. Since W_n terminates in $\bar{S}$, e_{m+1} is the only edge of $[S, \bar{S}]$ in G_m, and hence is a cut edge of G_m.

let e be any other edge of G_m incident with v_m. It follows that e must also be a cut edge of G_m, and hence of $G_m[S]$. But since $G_m[S] = G_n[S]$, every vertex in $G_m[S]$ is of even degree. However this implies that $G_m[S]$ has no cut edge, which is a contradiction. Hence proved. □

2.4 Matrix representation of graphs

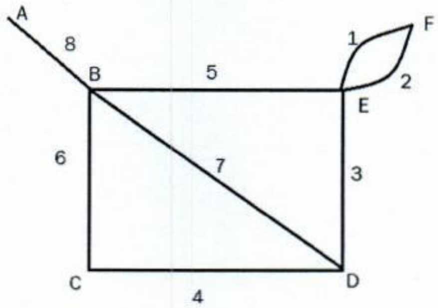
Although pictorial representation of a graph is very convenient for a visual study, other representations are better for computer processing. A matrix is convenient and useful way of representing a graph to a computer. Matrices lend themselves easily to mechanical manipulation. Besides, many known result can be readily applied to study the structural property of graph from an algebraic point of view. In many application of graph theory, such as in electrical network analysis and operation research, matrix also turn out to be the natural way of expressing the problem.

2.5 Incedence matrix

Let G be a graph with n vertices and e edges, and no self loops. Define an n by e matrix A=$[a_{ij}]$ where the element of matrix A is is defined by,

$$a_{ij} = \begin{cases} 1, & if\ jth\ edge\ e_j\ is\ incident\ on\ ith\ vertex\ v_i,\ and \\ 0 & otherwise. \end{cases}$$

Example 2.5.1.
The incidence matrix of the following figure



is $I[a_{ij}]=\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$.

2.6 Adjacency matrix

Let G be a graph with n vertices $v_1, v_2, v_3, ..., v_n$ with $n > 0$. The adjacency matrix A_G with respect to $v_1, v_2, v_3, ..., v_n$ is a $n \times n$ matrix $[a_{ij}]$ such that

$a_{ij} =$ no. of edges from v_i to v_j.

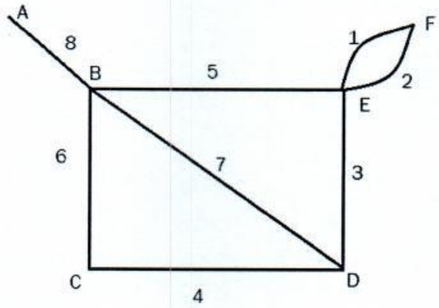
2.7 Circuit matrix

Let the no of different circuits in a graph G be q and the number of edges in G be e. Then a circuit matrix $B = [b_{ij}]$ of G is a $q \times e$ (0,1) matrix defined as follows:

$$b_{ij} = \begin{cases} 1, & if\ ith\ circuit\ includes\ jth\ edge,\ and \\ 0 & otherwise. \end{cases}.$$

And the circuit matrix is written as B(G).

Example 2.7.1.
Let us consider a Graph



The above graph has four different circuits, $\{1, 2\}$, $\{3, 5, 7\}$, $\{4, 6, 7\}$,and $\{3, 4, 6, 5\}$. Therefore, it's circuit matrix is a 4×8 (0,1) matrix as shown below:

$$B(G) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Theorem 2.7.2. *let A and B be, the incidence matrix and circuit matrix respectively of self loop free graph whose columns are arranged using the same order of edges. Then every row of B is orthogonal to every row of A; that is,*
$A \cdot B^T = B \cdot A^T = 0 \pmod{2}$.

Proof. Consider a vertex v and circuit Γ in a graph G. Either v is in Γ or not in Γ. If v is not in Γ , there is no edge in the circuit Γ that is incident on v. On the other hand, if v is in Γ, the number of those edge in the circuit Γ that are incident on v is exactly two.

Keeping this view, consider the ith row in A and jth row in B. Since the edges are arranged in the same order, the nonzero entries in the corresponding position occur only if the particular edge is incident on the ith vertex and is also in the jth circuit.

If the ith vertex is not in the jth circuit, there is no such nonzero entry, and the dot product of the two rows is zero. If ith vertex is in the jth circuit, there will be exactly two $1's$ in the sum of the products of individual entries.

Since $1 + 1 = 0 \ (mod\ 2)$, the dot product of the two arbitary rows one from A and the other from B is zero. Hence proved. □

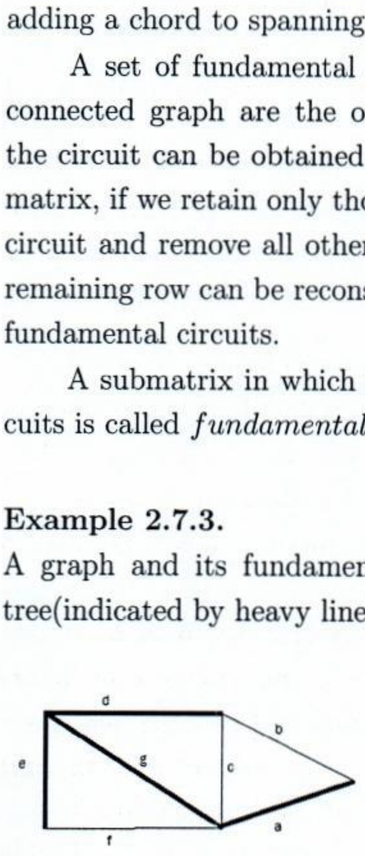
2.7.1 Fundamental circuit matrix and rank of B

For a connected graph G, let T be the spanning tree of G, then adding any one chord to T will create exactly one circuit, such a circuit formed by adding a chord to spanning tree is called *fundamental circuit.*

A set of fundamental circuits with respect to any spanning tree in a connected graph are the only independent circuit in a graph.The rest of the circuit can be obtained as ring sum of these circuits. Thus in a circuit matrix, if we retain only those rows that correspond to a set of fundamental circuit and remove all other rows, we would not lose any information. The remaining row can be reconstituted from the row corresponding to the set of fundamental circuits.

A submatrix in which all rows correspond to a set of fundamental circuits is called *fundamental circuit matrix* and denoted as B_f.

Example 2.7.3.
A graph and its fundamental circuit matrix with respect to a spanning tree(indicated by heavy line) are shown below



If n is the number of vertices and e is the no of edges in a connected graph, then B_f is an $(e-n+1) \times e$ matrix because the number of fundamental circuit is $e - n + 1$, each fundamental circuit being produced by one chord.

Let us arrange the columns in B_f such that all the $e - n + 1$ chords correspond to the first $e - n + 1$ columns. Furthermore, let us rearrange the row such that the first row correspond to the fundamental circuit made by the chord in the first column, the second row to the fundamental circuit made by the second, and so on..

The fundamental circuit matrix of the above fig is

$$B_f = \left(\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{array}\right)$$

A matrix B_f thus arranged can be written as

$$B_f = [I\mu|B_t] \tag{2.1}$$

Where I_μ is an identity matrix of order $\mu = e-n+1$, and B_t is the remaining μ by (n-1) sub matrix, corresponding to the branches of the spanning tree.

Theorem 2.7.4. *If B is a circuit matrix of a connected graph G with e edges and n vertices, then rank of B is* $e - n + 1$.

Proof. If A is an incidence matrix of G, then

$$A \cdot B^T = 0 \ (mod\ 2).$$

Therefore according to Sylvester's theorem we have, rank of A + rank of B $\leq e$ that is, rank of B $\leq e-$ rank of A. Since rank of A $= n - 1$, we have rank of B $\leq e - n + 1$. But rank of B $\geq$ $e - n + 1$. Therefore we must have rank of B $= e - n + 1$. □

Chapter 3

APPLICATIONS OF GRAPHS

3.1 The shortest path problem and Dijkstra's algorithm

Given a railway network connecting various town, determine the shortest route between two specified town in the network.

Solution: Here one must find in a weighted graph a path of minimum weight connecting two specified vertices u_0 and v_0; the weights represent distance by rail between directly linked towns and are therefore non negative.

We now present an algorithm for solving the shortest path problem. For clarity of exposition, we shall refer to the weight of a path in a weighted graph as its length; similarly the minimum weight of a (u,v) path will be called the distance between (u,v) and denoted by d(u,v).

The algorithm to be described was discovered by Dijkstra (1959) and independently by Hillier (1960). It finds not only a shortest (u_0, v_0) path, but shortest path from u_0 to all other vertices of G.

Dijkstra's algorithm

The basic idea of Dijkstra's algorithm is as follows;

Suppose that S is a proper subset of V such that $u_0 \in S$, and let $\bar{S}$ denote V/S. If P=$u_0, u_1, ..., \bar{u}\bar{v}$ is a shortest path from u_0 to $\bar{S}$ then clearly $\bar{u} \in S$ and the $(u_0, \bar{u})$ section of P must be a shortest $(u_0, \bar{u})$ path. Therefore

$$d(u_0, \bar{v}) = d(u_0, \bar{u}) + w(\bar{u}\bar{v})$$

and the distance from u_0 to $\bar{S}$ is given by the formula,

$$d(u_0, \bar{S}) = \min_{\substack{u \in S \\ v \in \bar{S}}} \{d(u_0, u) + w(u, v)\} \tag{3.1}$$

This formula is the basis of Dijkstra's algorithm. Starting with the set $S_0 = \{u_0\}$ an increasing sequence $S_0, S_1, ..., S_{v-1}$ of subset of V is constructed in such a way that at the end of stage i, shortest path from u_0 to all vertices in S_i are known.

The first step is to determine a vertex nearest to u_0. This is achieved by computing $d(u_0, \bar{S})$ and selecting a vertex $u_1 \in \bar{S}_0$ such that $d(u_0, u_1) = d(u_0, \bar{S}_0)$; by (3.1)

$$\begin{aligned} d(u_0, \bar{S}_0) &= \min_{\substack{u \in S_0 \\ v \in \bar{S}_0}} \{d(u_0, u) + w(u, v)\} \\ &= \min_{v \in \bar{S}_0} \{w(u_0, v)\} \end{aligned}$$

and so $d(u_0, \bar{S}_0)$ is easily computed.

We now set $S_1 = \{u_0, u_1\}$ and letP_1 denote the path u_0u_1; this is clearly a shortest (u_0, u_1) path. In general if the set $S_k = \{u_0, u_1, ..., u_k\}$ and corresponding shortest path $P_1, P_2, ..., P_k$ have already been determined, we computed $d(u_0, \bar{S}_k)$ using (3.1) and select a vertex $u_{k+1} \in \bar{S}_k$ such that $d(u_0, u_{k+1}) = d(u_0, \bar{S}_k)$. By (3.1)

$$d(u_0, u_{k+1}) = d(u_0, u_j) + w(u_j u_{k+1})$$

for some$j \leqslant k$; we get a shortest (u_0, u_{k+1}) path by adjoining the edge $u_j u_{k+1}$ to the path P_j.

Now we adopt the following labeling procedure. Throughout the algorithm each vertex v caries a label *l(v)* which is an upper bound on $d(u_0, v)$. Initially $l(u_0) = 0$ and $l(v_0) = \infty$ for $v \neq u_0$. As the algorithm proceeds,these labels are modified so that at the end stage of i; $l(u) = d(u_0, u)$ for $u \in S$ and $l(v) = \min_{u \in S_{i-1}} \{d(u_0, u) + w(uv)\}$ for $v \in \bar{S}_i$

The algorithm is given by:

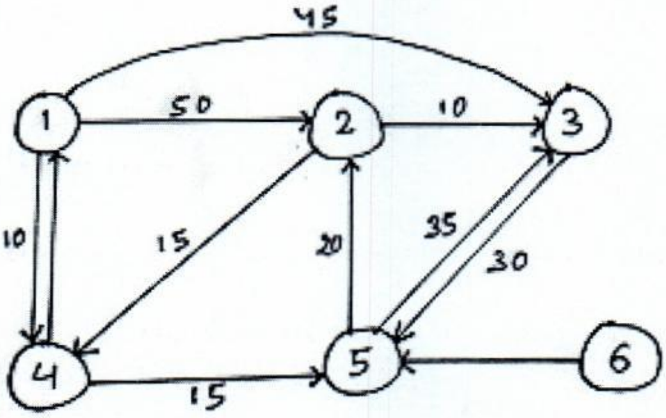
Step 1. Set $l(u_0) = 0$, $l(v) = \infty$ for $v \neq u_0$; $S_0 = \{u_0\}$ and i=0.

Step 2. For each $v \in \bar{S}_i$ replace *l(v)* by $\min\{l(v), l(u_i) + w(u_i v)\}$. Compute $\min_{v \in S_i}\{l(v)\}$ and let u_{i+1} denote a vertex for which this minimum is attained.

Set $S_{i+1} = S_i \cup \{u_{i+1}\}$

Step 3. If $i = v - 1$ stop. If $i < v - 1$ replace i by i+1 and go to step 2.

When the algorithm terminates the distance from u_0 to v is given by the final value of the lable $l(v)$.



Example 3.1.1.

Starting vertex is 1 and we construct the table to get shortest path.

Selected Vertex	2	3	4	5	6
4	50	45	10	∞	∞
5	50	45	10	25	∞
2	45	45	10	25	∞
3	45	45	10	25	∞
6	45	45	10	25	∞

Here we can not find any shortest path from vertex 1 to vertex 6.

3.2 The Chinese postman problem and Fleury's algorithm

In this job a postman pickup mail at the post office, deliverers it and then return to the post office. He must of course cover each street in his area atleast once, subject to this condition he wishes to choose his route in such a way that he walks as little as possible. This problem is known as *Chinese postman problem*, since it was first considered by a Chinese mathematician, Kuan (1962).

In a weighted graph, we define the weight of a tour $v_0e_1v_1...e_nv_0$ to be $\sum_{i=1}^{n} w(e_i)$. Clearly, the Chinese postman problem is just that of finding a minimum weight tour in a weighted connected graph with non negative weights. We shall refer to such a tour as an optimal tour.

If G is Eulerian then any Euler tour of G is optimal tour because an Euler tour is a tour that traverses each edge exactly once. The Chinese postman problem is easily solved in this case. The algorithm to solve the problem is given by,

Fleury's Algorithm

Step 1. Choose an arbitrary vertex v_0 and set $W_0 = v_0$.

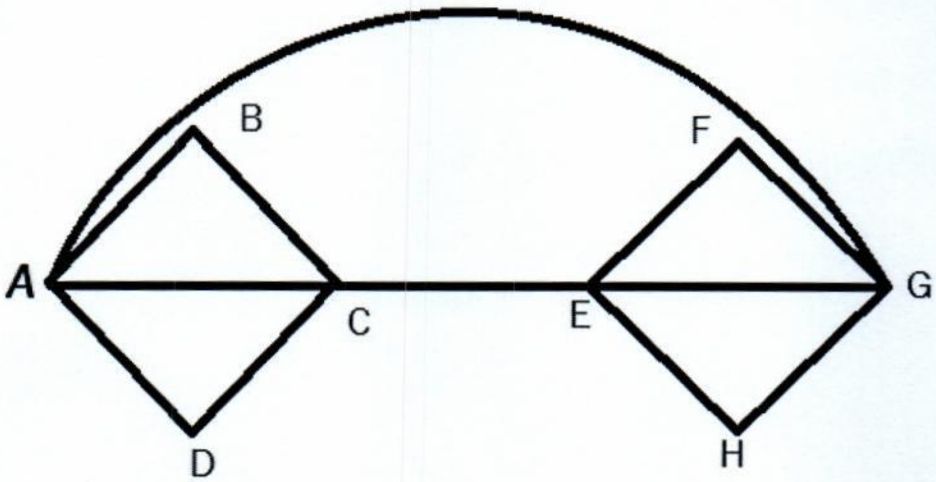
Step 2. Suppose that the trail $W_i = v_0e_1v_1...e_iv_i$ has been chosen. Then choose an edge e_{i+1} from $E \setminus \{e_1, e_2, e_3, ..., e_i\}$ in such a way that

(i) e_{i+1} is incident with v_i;

(ii) Unless there is no alternative e_{i+1} is not a cut edge of $G_i = G \setminus \{e_1, e_2, ..., e_i\}$.

Step 3. Stop when step 2 can no longer be implemented.

Example 3.2.1.



Current path	Next edge
$W_0 = A$	A,B
$W_1 = A, B$	B,C
$W_2 = A, B, C$	C,A
$W_3 = A, B, C, A$	A,D
$W_4 = A, B, C, A, D$	D,C
$W_5 = A, B, C, A, D, C$	C,E
$W_6 = A, B, C, A, D, C, E$	E,G
$W_7 = A, B, C, A, D, C, E, G$	G,F
$W_8 = A, B, C, A, D, C, E, G, F$	F,E
$W_9 = A, B, C, A, D, C, E, G, F, E$	E,H
$W_{10} = A, B, C, A, D, C, E, G, F, E, H$	H,G
$W_{11} = A, B, C, A, D, C, E, G, F, E, H, G$	G,A
$W_{12} = A, B, C, A, D, C, E, G, F, E, H, G, A$	

From this table we get an Euler circuit, hence the graph is Eulerian so the tour W_{12} is optimal tour.

3.3 The Travelling salesman problem

A travelling salesman wishes to visit a number of towns and then return to his starting point. Given the travelling times between towns, how should he plan his itinerary so that he visits each town exactly once and travels in all for as short a time as possible? This is known as the travelling salesman problem. In graphical terms, the aim is to find a minimum-weight Hamilton cycle in a weighted complete graph, We shall call such a cycle an *optimal cycle*. In contrast with the shortest path problem and the connector problem, no efficient. algorithm for solving the traveling salesman problem is known. It is therefore desirable to have a method for obtaining a reasonably good (but not necessarily optimal) solution. We shall show how some of our previous theory can be employed to this end.

One possible approach is to first find a Hamilton cycle C. Perhaps the simplest such modification is as follows.

Let $C = v_1v_2v_3....v_vv_1$, Then for all i and j such that $1 < i+1 < j < v$, we can obtain a new Hamilton cycle

$$C_{ij} = \ v_1v_2...v_iv_{j-1}...v_{i+1}v_{j+1}v_{j+2}...v_vv_1$$

by deleting the edges v_iv_{i+1} and v_jv_{j+1} and adding the edges v_iv_j and $v_{i+1}v_{j+1}$, as shown in figure(2) If for some i and j

$$w(v_iv_j) + w(v_{i+1}v_{j+1}) < w(v_iv_{i+1}) + w(v_jv_{j+1})$$

the cycle C_{ij} will be an improvement on C. After performing a sequence of above modifications,one is left with a cycle that can be improved no more by these methods. This final cycle will almost certainly not be optional, but it is a reasonable assumption that it will often be fairly good; for greater accuracy, the procedure can be repeated several times, starting with a different cycle each time.

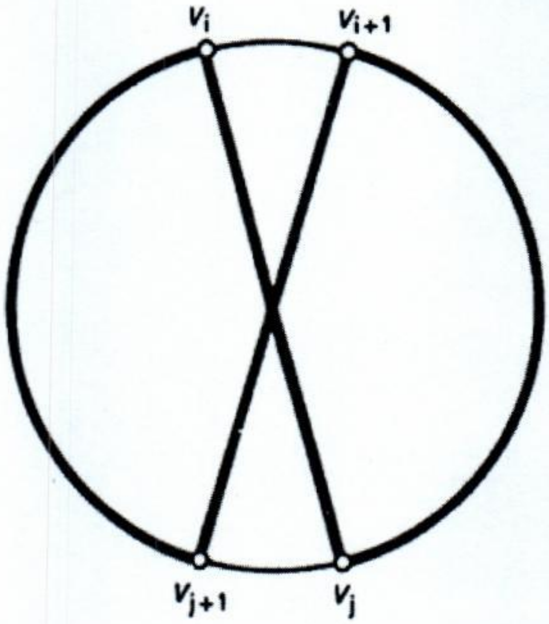
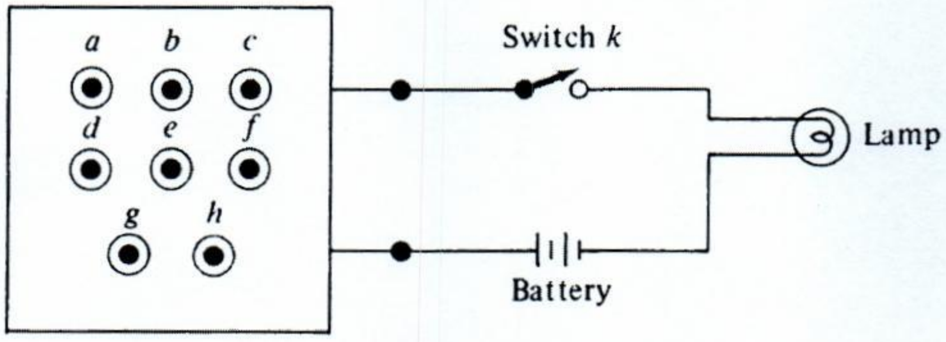


Figure 3.1:

3.4 Application to switching network

Suppose you are given a box that contains a switching network consisting of eight switches $a, b, c, d, e, f, g, and\ h$. The switches can be turned on or off from outside. You are asked to determine how the switches are connected inside the box, without opening the box, of course.

One way to find the answer is to connect a lamp at the available terminals in series with a battery and additional switch k, as shown in the fig;



The box

In this experiment, suppose we discover that the combinations that turn on the lamp are eight:

(a,b,f,h,k),(a,b,g,k),(a,e,f,g,k),(a,e,h,k),(b,c,e,h,k),(c,f,h,k),(c,g,k),(d,k)

*Solution:*Consider the switching network as a graph whose edges represent switches. We can assume that the graph is connected, and has no self loop. Since a lit lamp implies the formation of circuit, we can regard the preceding list as a partial list of circuit in the corresponding graph. With this list we form a circuit matrix:

$$\mathrm{B}=\begin{pmatrix} 1&1&0&0&0&1&0&1&1\\ 1&1&0&0&0&0&1&0&1\\ 1&0&0&0&1&1&1&0&1\\ 1&0&0&0&1&0&0&1&1\\ 0&1&1&0&1&0&0&1&1\\ 0&0&1&0&0&1&0&1&1\\ 0&0&1&0&0&0&1&0&1\\ 0&0&0&1&0&0&0&0&1 \end{pmatrix}$$

Next to simplify the matrix, we should remove the obviously redundant circuits.

Observe that the following ring sums of circuits give rise to other circuits:

$$(a,b,g,k) \oplus \ (c,f,h,k) \oplus (c,g,k) = (a,b,f,h,k),$$

$$(a,b,g,k) \oplus (a,e,h,k) \oplus (c,g,k) = (b,c,e,h,k),$$

$$(a,e,h,k) \oplus \oplus (c,f,h,k) \oplus (c,g,k) = (a,e,f,g,k).$$

Therefore we can delete the first, third, and fifth rows from matrix B, without any lose of information.

Remaining is a 5 by 9 matrix B_1

$$B_1 = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Our next goal is to bring the matrix B_1 to the form of (2.1)

For this we interchange columns to get B_2:

$$\begin{array}{c} \begin{array}{ccccccccc} b & e & f & g & d & a & c & h & k \end{array} \\ B_2 = \left(\begin{array}{ccccc|cccc} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

Adding the fourth row in B_2 to the first, we get B_3.

$$
\begin{array}{c}
\begin{array}{ccccccccc} b & e & f & g & d & a & c & h & k \end{array} \\
B_3 = \left(\begin{array}{ccccc|cccc}
1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{array}\right) = [I_5 \mid F]
\end{array}
$$

We note that there are no redundant circuits in matrix B_3, and B_3 is a fundamental circuit matrix of the required graph. Since the rank of B_3 is five, and the network was assumed to be connected we have the following information about the graph:
number of edges e = 9
nullity μ= 5
rank r=4
number of vertices n=5.

Now we have to construct a graph from it's fundamental circuit matrix. So we have to construct an incidence matrix from B_3.

Since the rows in the incidence matrix are orthogonal to those in B_3 according to eq we must first look for a 4 by 9 matrix M, whose rows are linearly independent and are orthogonal to those of B_3.
since

$$B_3 = [I_5 \mid F],$$

an orthogonal matrix B_3 is

$$
\begin{aligned}
M &= [-F^T \mid I_4] \\
&= [F^T \mid I_4],
\end{aligned}
$$

because in mod 2 arithmatic -1 = 1. Thus

$$\begin{array}{c} \begin{array}{ccccccccc} b & e & f & g & d & a & c & h & k \end{array} \\ M = \left(\begin{array}{ccccc|cccc} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right) \end{array}$$

Clearly, the rank of M is four, and it is easy to check that

$$B_3 \cdot M^T \;=\; 0$$

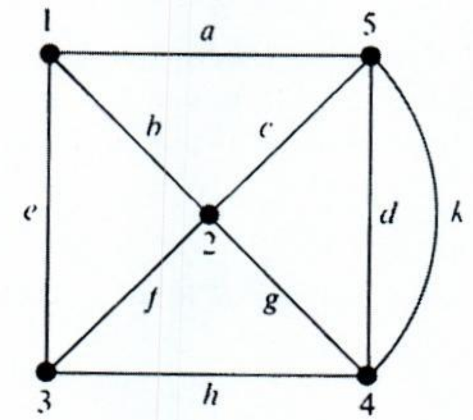
Before M can be regarded as a reduced incidence matrix, it must have at most two 1's in each column. This can be achieved by adding (mod 2) the third row to the fourth in M, which gives us M'.

$$\begin{array}{c} \begin{array}{ccccccccc} b & e & f & g & d & a & c & h & k \end{array} \\ M' = \left(\begin{array}{ccccc|cccc} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{array}\right) \end{array}$$

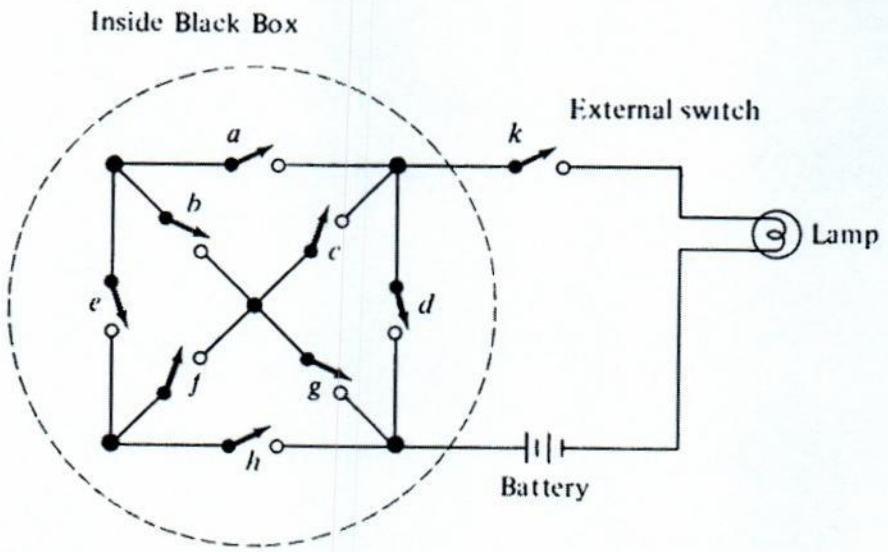
Matrix M' is a the reduced incidence matrix. The incidence matrix A can be obtained by adding a fifth row to M' such that there are exactly two 1's in every column; that is,

$$
\begin{array}{cc}
 & \begin{array}{ccccccccc} b & e & f & g & d & a & c & h & k \end{array} \\
A = & \begin{pmatrix}
1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1
\end{pmatrix}
\end{array}
$$

From the incidence matrix A we can readily construct the graph and hence the corresponding switching network,as shown below;



(a)



3.5 The connector problem and Kruskal's algorithm

A railway network connecting a number of towns is to be setup. Given the cost c_{ij} of constructing a direct link between towns v_i and v_j, design such a network to minimize the total cost of construction. This is known as the *connector problem.*

By regarding each town as vertex in weighted graph with weights $w(v_iv_j) = c_{ij}$, it is clear that this problem is just that of finding, in a weighted graph G, a connected spanning subgraph of minimum weight. Moreover, since the weight represent costs, they are certainly non negative, and we may therefore assume that such a minimum weight spanning tree of a weighted graph will be called *optimal tree.* Now we present an algorithm to find an optimal tree in a non trivial weighted connected graph. At first let us consider each weight is 1 *i.e.* $w(e) = 1$. Then an optimal tree is a spanning tree with as few edges as possible. Since each spanning tree of a graph has the same number of edges, so we need to construct some spanning tree of the graph. A simple inductive algorithm for finding such a tree is given by:

Kruskal's Algorithm

The Kruskal's algorithm is:

Step 1. Choose a link e_1 such that $w(e_1)$ is as small as possible.

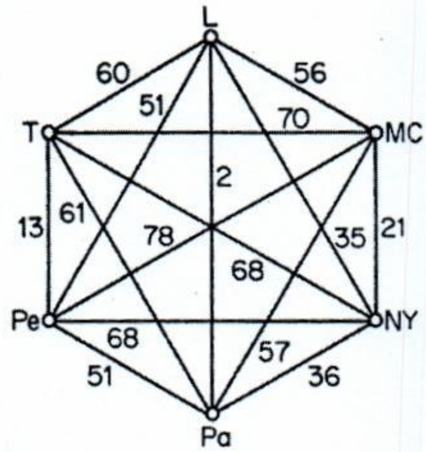
Step 2. If edges $e_1, e_2, ..., e_i$ have been chosen, then choose an edge e_{i+1} from $E \setminus \{e_1, e_2, ..., e_i\}$ in such a way that $G[\{e_1, e_2, ..., e_{i+1}\}]$ is acyclic.

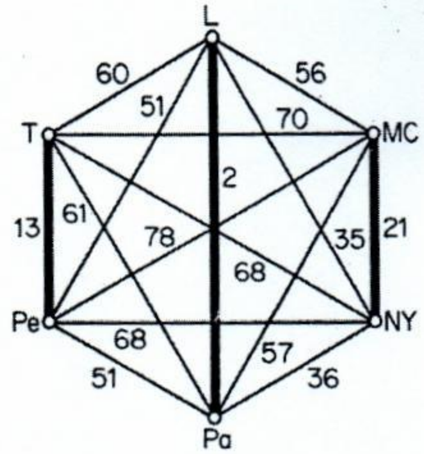
Step 3. Stop when step 2 cannot be implemented further.

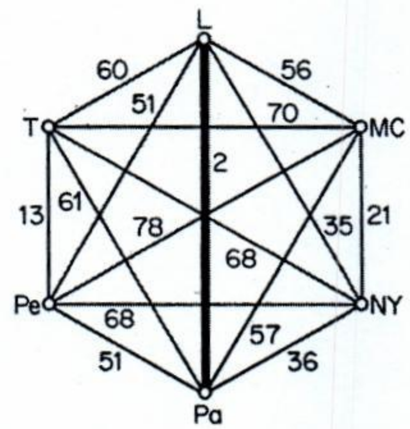
Let us consider an example *i.e.* Consider the table of airline distances in miles between six of the largest cities in the world, London, Mexico City, New York, Paris, Peaking and Tokyo:

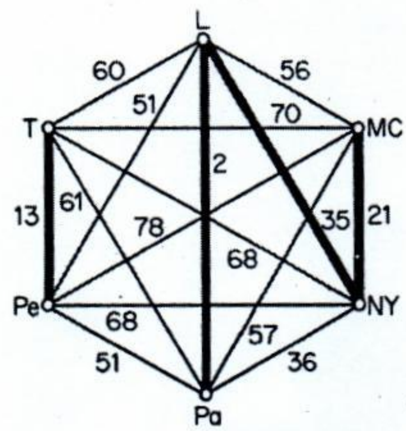
	L	MC	NY	Pa	Pe	T
L	—	5558	3469	214	5074	5959
MC	5558	—	2090	5725	7753	7035
NY	3469	2090	—	3636	6844	6757
Pa	214	5725	3636	—	5120	6053
Pe	5074	7753	6844	5120	—	1307
T	5959	7035	6757	6053	1307	—

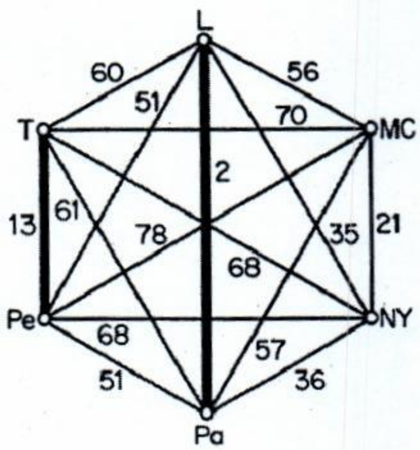
This table determines a weighted complete graph with vertices L,MC,NY,Pa,Pe and T. The construction of an optimal tree in this graph is shown below. For our convenience the distances are shown in hundreds of miles.

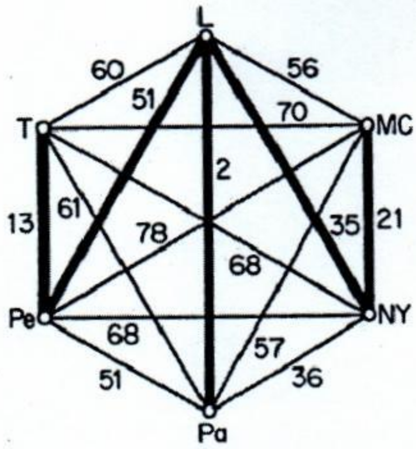

L
60
56
51
70
T
MC
2
13
61
78
35
21
68
Pe
NY
68
57
51
36
Pa


L
60
56
51
70
T
MC
2
13
61
78
35
21
68
Pe
NY
68
57
51
36
Pa


L
60
56
51
70
T
MC
2
13
61
78
35
21
68
Pe
NY
68
57
51
36
Pa


L
60
56
51
70
T
MC
2
13
61
78
35
21
68
Pe
NY
68
57
51
36
Pa


L
60
56
51
70
T
MC
2
13
61
78
35
21
68
Pe
NY
68
57
51
36
Pa


L
60
56
51
70
T
MC
2
13
61
78
35
21
68
Pe
NY
68
57
51
36
Pa

Bibliography

[1] D. B. West, Introduction To Graph Theory, Prentice Hall India Learning Private Limited, 2002

[2] G. Zhang, P. Chartrand, Introduction To Graph theory, Mc Graw Hill India, 2007

[3] J. A. Bondy and U.S.R. Murty, Graph Theory with Applications, Department of Combinatoriecs and Optimization, University of Waterloo, Ontario,Canada, Macmillan press Ltd. 1976

[4] R. J. Wilson, Introduction to Graph Theory, Addison Wesley Longman Limited, 1998

Signature of the Students by whom Project "Graph Theory & Its Applications" is carried out.

Sl No	Roll Number	Name of the Student	Signature
1	BS17-069	RAJANIKANTA JENA	Rajanikanta Jena
2	BS17-074	RASMIRANJAN SETHY	Rasmi Ranjan Sethi
3	BS17-098	SK ASATULLAH	SK Asatullah.
4	BS17-104	SWAGATIKA DAS	Swagatika Das.