UNIT 9: TRAVERSABILITY

UNIT STRUCTURE

9.1 Learning Objectives
9.2 Introduction
9.3 Euler graphs
9.4 Hamiltonian Graph
9.5 Shortest Path Problem
9.6 The Traveling Salesman Problem (TSP)
9.7 The Chinese Postman Problem
9.8 Let Us Sum Up
9.9 Answers to Check Your Progress
9.10 Further Reading
9.11 Model Questions

9.1 LEARNING OBJECTIVES

After going through this unit, you will be able to-

- understand Eulerion and Hamiltonian paths and circuits
- apply theorems and conditions to determine the existence of Eulerian or Hamiltonian circuits
- apply the concepts of Eulerian and Hamiltonian graphs to solve Problems in graph theory and understand real world applications.

9.2 INTRODUCTION

In this unit we learn about traversability Eulerian and Hamiltonian graphs and various properties exhibited by these graphs.

Traversability refers to the ability to traverse a graph by visiting each edge or vertex in a specific manner. Eulerian graphs, named after the swiss mathematician Leonhard Euler, are graphs that visits every edge exactly once. Hamiltonian, graphs, named after the Irish mathematician, William Rowan Hamilton, are graphs that visit every vertex exactly once.

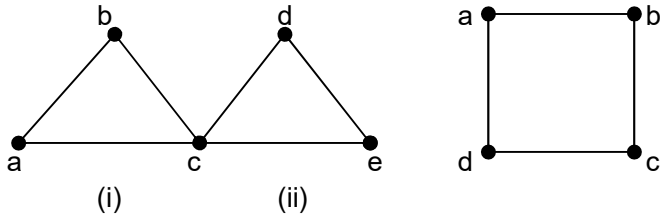
9.3 EULER GRAPHS

If some walk in a graph contains all the edges of the graph, then the walk is called and Euler line.

Since the Euler line contains all the edges of the graph and Euler graph is always connected. Euler line is also Known as Euler path or Euler walk. In other words, it may be described as : “If there is a connected graph with a walk that passes through each and every edge of the graph only once, then that type of walk is known as the Euler with or Euler path”.

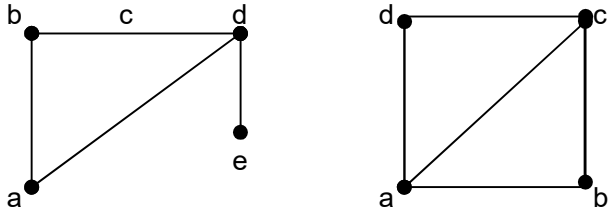
Euler Circuit: An Eulerian cycle is a path in a graph that visits every edge exactly once and starts and ends at the same vertex.

Example of Eulerian circuits:



A graph containing an Euler circuit is Known as an Euler graph.

Graphs having Euler path but Euler Circuit"



Theorem: A given connected graph G is a Euler graph if and only if all vertices of G are of even degree.

Proof: Suppose G is a Euler graph. It therefore consists of a Euler line (which is a closed walk). In tracing this walk, we observe v, it goes through two `now' edges incident on $v$: with one we `entered' v, and with the other `exited'.

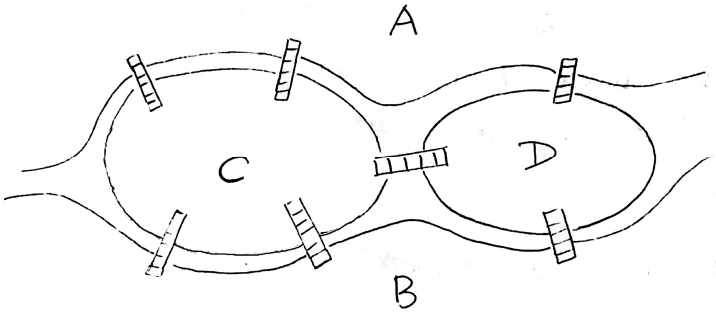
This is time not only of all intermediate vertices of walk but also of the terminal vertex, because we `exited' and `entered' the same vertex at the beginning and end of the walk respectively thus, if G is a Euler graph, the degree of every vertex is even.

Conversely, assume that all vertices of G are of even degree. Now, we construct a walk starting at an arbitrary vertex v and going through the edges of G such no edge is traced more than once.

We continue tracing as far as possible. Since every vertex is of even degree, we can exit from every vertex we enter; the tracing cannot stop at any vertex except v. And since v is also of even degree, we shall eventually reach v when the tracing comes to an end. If this closed walk `h', which we just traced, includes all the edges of G, then G is a Euler graph. If not. We remove from G all the edges in h and obtain a sub-graph h' of G formed by the remaining edges. Since both G and H have all their vertices of even degree, the degrees of the vertices of h' are also even. Moreover, h' must touch h at least at one vertex `a' because G is connected. Starting from `a' we can again construct a new walk in graph h' Since all the vertices of h' are of even degree, this walk in h' to from a new walk, which starts and ends at vertex v and has more edges

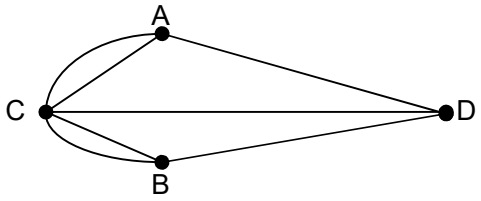
than h. This process can be repeated until we obtain a closed walk that traverses all the edges of G. Thus, G is Euler graph.

Konigsberg Bridge Problem: Two islands C and D formed by the Pregel River in Konigsberg (then capital of East Prussia but now renamed Kaliningrad and in West soviet Russia) were connected to each other and to the banks A and B with seven bridges as shown:



The problem was to start at any of the four land areas of the city A, B, C or D and walk over of the seven bridges exactly once return to the starting point. (without swimming Euler Presented this by the following graph:

- Vertices represent land areas
- Edges represent bridges.



We observe that not all of its vertices are of even degree. Hence, it is not a Euler graph. Thus, it is not possible to walk over each of the seven bridges exactly once and return to the starting point.

Theorem : A connected graph G is a Euler graph if only if it can be decomposed into circuits.

Proof: Suppose graph G can be decomposed into circuits; that is, G is a union of edge disjoint circuits. Since the degree of every vertex in a circuit is two, the degree of every vertex in G is even. Hence, G is a Euler graph.

Conversely, let G be a Euler graph. Consider a vertex v_1 There are at least two edges incident at v_1. Let one of these edges be between v_1 and v_2. Since vertex v_2 is also of even degree, it must have at least another edge, say between v_2 and v_3. Proceeding this way, we eventually arrive at a vertex that has previously been traversed, thus forming a circuit Γ.

Let us remove Γ from G. All vertices in the remaining graph (not necessarily connected) must also be of even degree. From the remaining graph remove another

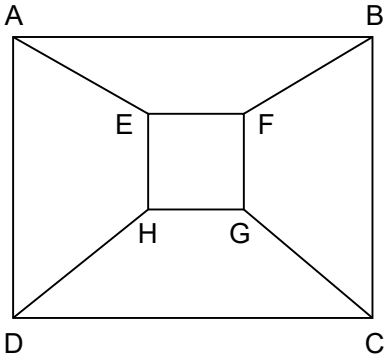
circuit in exactly the same way as we removed Γ from G continue this process, until no edges are left. Hence, the theorem is verified.

9.4 HAMILTONIAN GRAPH

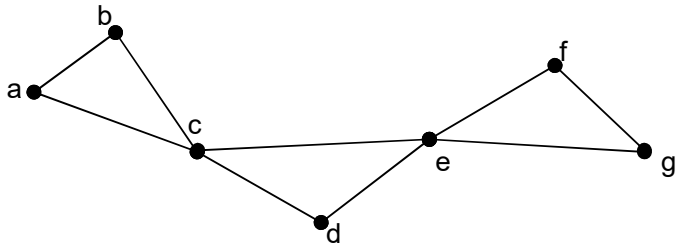
A Hamiltonian path is a simple path in a graph that visits every vertex exactly once.

A Hamiltonian path that starts and ends at the same vertex forming a closed loop, is known as a Hamiltonian cycle or a Hamiltonian Circuit.

Example:



ABFGCDHEA is a Hamiltonian circuit.
Example of a graph having Hamiltonian path but not Hamiltonian Circuit:



abcdefg is a Hamiltonian path.

A Hamiltonian graph is a connected graph that contains a Hamiltonian cycle. If such a cycle does not exist, the graph is considered non- Hamiltonian

Ore's Theorem : Let G be a simple graph with n vertices. Where n ≥ 3. If for every pair of non-adjacent vertices $\cup$ and ν, the sum of their degrees satisfies deg (u) + deg ($\cup$) + deg (ν) ≥ n, then G is Hamiltonian.

Proof: Let G be a simple graph which satisfies the given condition, but G is not Hamiltonian. This implies G does not have a Hamiltonian cycle.

Consider a maximal Hamiltonian path in G, which is a Hamiltonian path to which no more edges can be added without destroying its path property. Let this path be $(v_1, v_2, \ldots, v_n)$

Since G is not Hamiltonian, the edge (v_1, v_n) does not exist in G. Because if it did, the path would then form a Hamiltonian cycle.

Now, for any vertex v_1 in the path (where 2 ≤ I ≤ n) consider the pair of vertices v_1 and v_i. The potential edges are (v_1, v_i) and (v_{i-1}, v_n).

The given hypothesis states that for any non-adjacent vertices $\cup$ and v,

deg $(\cup)$ + deg (v) ≥ n.

If both edges (v_1, v_i) and (v_{i-1}, v_n) existed, then they could be combined with the path to form a Hamiltonian cycle. However, this would contradict our assumption that G is not Hamiltonian and that the path is maximal. Therefore, at most one of these two edges can exist. This means that for each I from 2 to n, at most one edge exists connecting v_1 to some v_i or v_{i-1} or v_n. The total number of such edges incident to either v_1 or v_n is thus limited to n — 1 choices of i.

Therefore, sum of degrees for v_1 and v_n i,e, deg (v_1) + deg (v_n) < n. This contradicts our hypothesis.

Thus, our assumption that G is not Hamiltonian is wrong.

Dirac's Theorem : Let G be a simple graph with n vertices, where n ≥ 3.
If deg (v) ≥ $n/2$ for each vertex v, then the graph G is Hamiltonian graph.

W CHECK YOUR PROGRESS

Q 1: Draw a graph which is Hamiltonian out not Eulerian.
Q 2: Draw a graph which is Eulerian but not Hamiltonian.

9.5 SHORTEST PATH PROBLEM

The shortest path problem in graph theory involves finding a path between two vertices in a graph that minimizes the total sum of the weights of its constituent edges.

Types of shortest path problems.

(i) Single-source shortest path : It finals the shortest paths from a single starting vertex to all other vertices in the graph.
(ii) Single-pair shortest path : It finds the shortest path between one specific source vertex and one specific destination vertex.

(iii) All pairs shortest path : It finds the shortest paths between every possible pair of vertices in the graph.

(iv) Single destination shortest path : It finds the shortest path from all other vertices to a single destination vertex in the graph.

Common Algorithms:

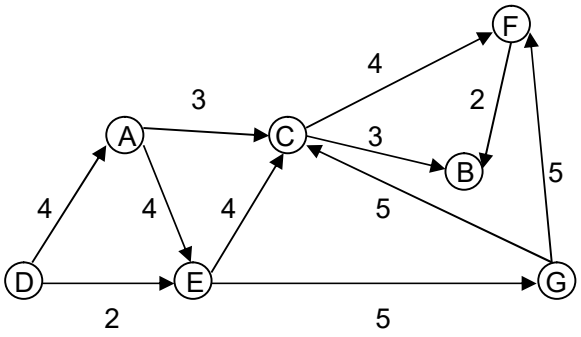
(i) Dijkstra's Algorithm: It is an efficient algorithm for finding the shortest path from a single source to all other vertices in a weighted graph, but it does not handle negative edge weights.

(ii) Bellman Ford Algorithm: This algorithm is capable of finding shortest paths from a single source to all other vertices, even in graphs with negative edge weights.

(iii) Floyd - Warshall Algorithm: It is used for the all pairs shortest path problem.

Example:

(1)

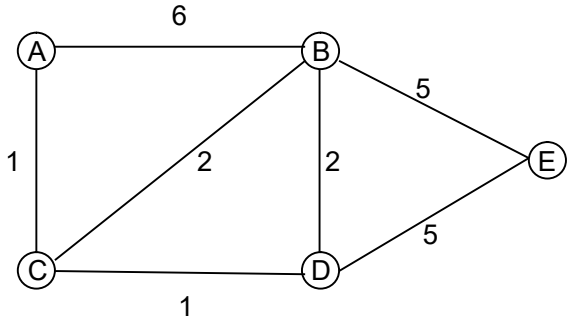


The shortest path from vertex D to vertex F in the graph above is:

D $\rightarrow$ E $\rightarrow$ C $\rightarrow$ F

With a total path weight of 2+4+4= 10. Other paths from D to F are also possible, but they have a higher total weight, So they cannot be considered to be the shortest path.

(2)



The shortest path from A to E is:

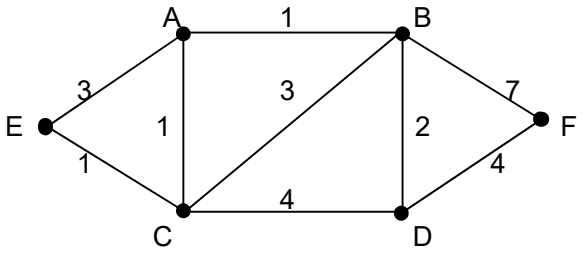
A $\rightarrow$ C $\rightarrow$ D $\rightarrow$ E

With a total weight of 1+1+5 =7

Note: In the shortest path problem, a Graph can represent anything from a road network to a communication network, where the vertices can be intersections, cities and the edges can be roads, flight paths or data links.

W CHECK YOUR PROGRESS

Q 3: Find the shortest path from E to F



9.6 THE TRAVELING SALESMAN PROBLEM (TSP)

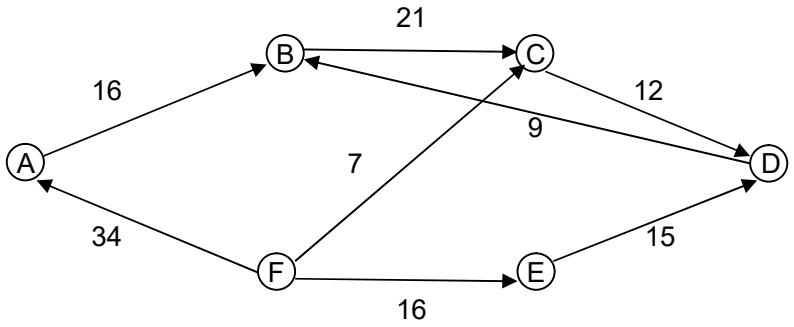
A traveling salesman wishes to go to a certain numbers of destinations in order to sell objects. He wants to travel to each destination exactly once and return have taking the shortest total route.

Each voyage can be represented as a graph G = (V, E) where each destination, including his home, is a vertex and if there is a direct route that connects two distinct destinations then there is an edge between those two vertices. The traveling salesman problem is solved if there exists a shortest route that visits each destination once and permits the salesman to return home. (This route will be a Hamiltonian cycle) The goal is to minimize the total cost or distance traveled, where the edge weights represent these costs.

Breakdown of TSP

- **Vertices:** Represent location or `cities' that need to be visited.
- **Edges:** Each edge has a numerical value (weight) that signifies the `cost' of traveling along that route, which could be distance, time or money. The edges connect the vertices, representing travel paths or routes between locations.
- **Hamiltonian cycle:** A tour that visits every vertex in the graph exactly once and ends at the starting vertex, forming a closed loop.
- **The Problem:** To find the Hamiltonian cycle with the minimum total weight (cost).

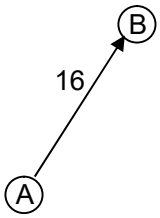
Example: Consider the following graph with six cities and the distances between them:



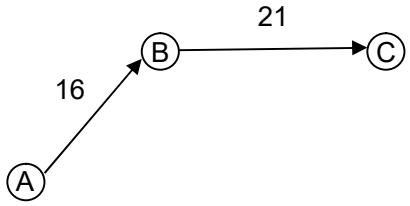
Let the origin be A.

Since the origin is mentioned, the solution must always start and end at that node.

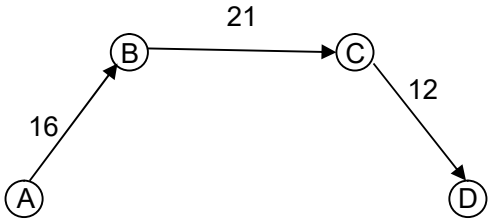
Among the edges leading from A, A $\rightarrow$ B has the shortest distance.



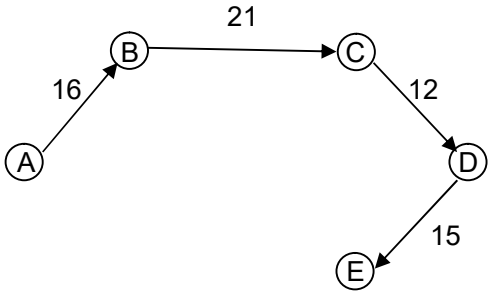
Then B $\rightarrow$ C has the shortest and only edge between them, therefore it is included in the output graph.



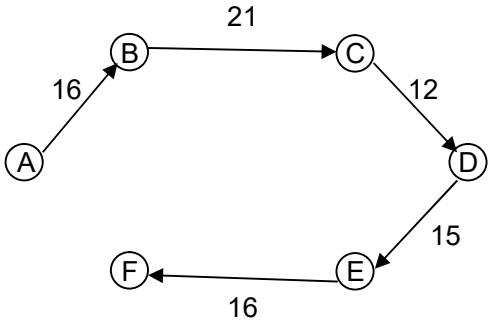
There is only one edge between C $\rightarrow$ D, therefore, it is added to the output graph.



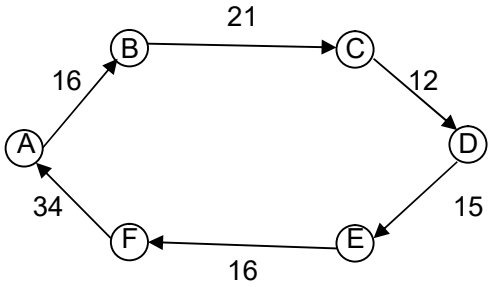
There are two outward edges from D. Even though D $\rightarrow$ E has lower distance than D $\rightarrow$ E, but B is already visited once and it would form a cycle if added to the output graph. Therefore, D $\rightarrow$ E is added to the output graph.



There is only edge from E, i. e, E -> F. Therefore, it is added.



Again, even though $F \rightarrow C$ has lower distance than $F \rightarrow A$, but $F \rightarrow A$ is added into the output graph in order to avoid the cycle that would form since C is already visited once.



$\therefore$ The shortest path originates and ends at A is: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow A$

The cost of the path is: 16+21+12+15+16+34 = 114

Note: This approach of solving TSP is known as the greedy algorithm.

9.7 THE CHINESE POSTMAN PROBLEM

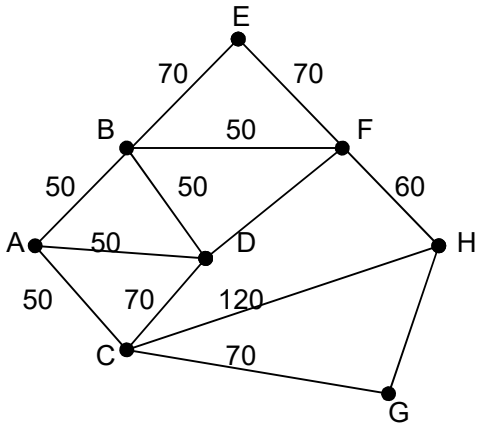
The Chinese Postman Problem (also known as Route Inspection Problem or Guan's route Problem) seeks the shortest possible closed path (circuit) to visit every edge of a connected, undirected graph at least once. The solution involves finding the minimum cost to add duplicate edges to the graph, to create a graph where a eulerian circuit exists, meaning every edge is traversed exactly once, starting and ending at the same vertex.

An algorithm for finding an optimal Chinese postman route is:

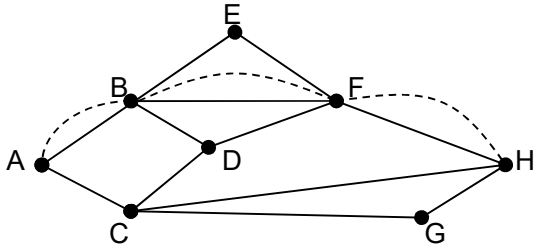
Step 1: List all odd vertices.
Step 2: List all possible pairings of odd vertices.
Step 3: For each Pairing, find the edges that connect the vertices with the minimum weight.
Step 4: Find the pairings such that the sum of the weights is minimized.
Step 5: On the original graph, add the edges that have been found in step 4.
Step 6: The length of an optimal Chinese postman route is the sum of all the edges added to the total found in step 4.
Step 7: A route corresponding to this minimum weight can then be easily found.

Example:

We apply the above algorithm to the following problem:



Step 1: The odd vertices are A and H
Step 2: There is only one way of pairing these odd vertices, namely AH
Step 3: The shortest way of joining A to H is using path $A \to B \to F \to H$, a total of 50+50+60 = 160 length.
Step 4: Draw these edges onto the original network.



Step 5: The length of the optimal Chinese postman rote is the sum of all the edges in the original network, which is,
50+50+50+50+70+70+70+120+60+60+50+70+70 = 840
Plus the answer found in step 4. Which is 160.

∴ Length of the optimal Chinese postman route is 840+160 = 1000

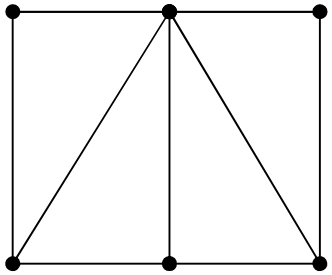
Step 6: One possible route corresponding to this length is A D C G H C B D F B E F H F B A, but many other possible routes of the same minimum length can be found.

9.8 LET US SUM UP

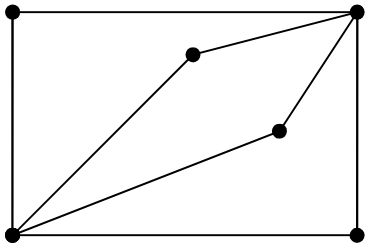
- In this unit we explored several fundamental concepts in graph theory, including Hamiltonian graphs, Euler graphs, shortest path problems, traveling salesman problem and Chinese postman problem. These concepts and algorithms have far-reaching implications in various fields, including:
- Logistics & Transportation: Route planning, scheduling and optimization.
- Network Optimization: Network design, routing and optimization.
- Computer Science: Algorithm design, data structures and computational complexilty.

9.9 ANSWERS TO CHECK YOUR PROGRESS

Answer to Q 1: Hamiltonian but not Eulerian graph



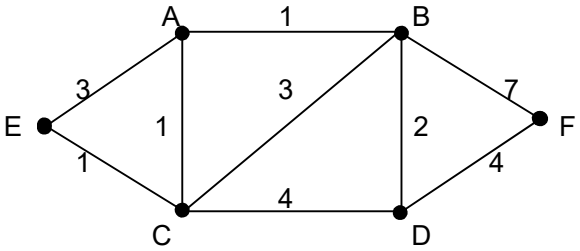
Answer to Q2: Euler but not Hamiltonian graph.



Answer to Q3: Shortest path from E to F is:

$E \rightarrow C \rightarrow A \rightarrow B \rightarrow D \rightarrow F$

i.e ECABDF



Length is 1+1+1+2+4 = 9

O9.10 FURTHER READING

1. N. Deo “Graph Theory with Applications to Engineering and Computer Science”
2. “Arc Routing : Theory, solntions & Applications”- by Moshe Dror.
3. Thomas H. Cormen, Charles E. Leiserson, R. L. Rivest, C. Stein, “Introduction to Algorithms’’.

M9.11 MODEL QUESTIONS

Q 1: Draw a graph having both Euler circuit and Hamiltonian circuit.
Q 2: Draw a graph having neither an Euler circuit nor a Hamiltonian circuit.
Q 3: Prove Dirac’s Theorem.

*** *** ***