CONNECTEDNESS IN DIRECTED GRAPHS

Definitions

A directed graph is said to be *strongly connected*, if there is a path from V_i to V_j and from V_j to V_i where V_i and V_j are any pair of vertices of the graph.

For a directed graph to be strongly connected, there must be a sequence of directed edges from any vertex in the graph to any other vertex.

A directed graph is said to be *weakly connected*, if there is a path between every two vertices in the underlying undirected graph. In other words, a directed graph is weakly connected if and only if there is always a path between every two vertices when the directions of the edges are disregarded. Clearly any strongly connected directed graph is also weakly connected.

A simple directed graph is said to be *unilaterally connected*, if for any pair of vertices of the graph, at least one of the vertices of the pair is reachable from the other vertex.

We note that a unilaterally connected digraph is weakly connected, but a weakly connected digraph is not necessarily unilaterally connected. A strongly connected digraph is both unilaterally and weakly connected.

For example, let us consider the graphs shown in the Figs 7.55, 7.56 and 7.57.

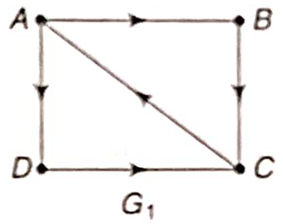

Fig. 7.55

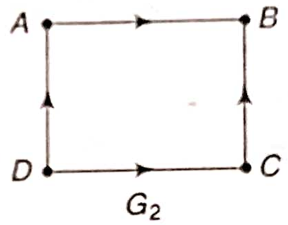

Fig. 7.56

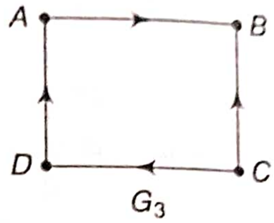

Fig. 7.57

G_1 is a strongly connected graph, as the possible pairs of vertices in G_1 are (A, B), (A, C), (A, D), (B, C), (B, D) and (C, D) and there is a path from the first vertex to the second and from the second vertex to the first in all the pairs.

For example, let us take the pair (A, B). Clearly the path from A to B is $A - B$ and the path from B to A is $B - C - A$.

Similarly if we take the pair (B, D), the path from B to D is $B - C - A - D$ and the path from D to B is $D - C - A - B$.

Clearly G_2 is only a weakly connected graph.

G_3 is unilaterally connected, since there is a path from A to B, but there is no path form B to A. Similarly, there is a path from D to B, but there is no path from B to D.

Definition

A subgraph of a digraph G that is strongly connected but not contained in a larger strongly connected subgraph viz., the maximal strongly connected subgraph is called the *strongly connected component of* G [see Example (7.8)].

SHORTEST PATH ALGORITHMS

A graph in which each edge 'e' is assigned a non-negative real number $w(e)$ is called a *weighted graph* $w(e)$ called the *weight of the edge* 'e' may represent distance, time, cost etc. in some units.

A *shortest path* between two vertices in a weighted graph is a path of least weight. In an unweighted graph, a shortest path means one with the least number of edges.

In this section, we shall deal with the problem of finding the shortest path between any two vertices in a weighted graph. Many algorithms are available to find the shortest path in a weighted graph. We shall discuss two of them here one discovered by Edsger Dijkstra and the other by Warshall.

Dijkstra's Algorithm

To find the length (or weight) of the shortest path between two vertices, say a and z, in a weighted graph, the algorithm assigns numerical labels to the vertices of the graph by an iterative procedure. At any stage of iteration, some vertices will have temporary labels (that are not bracketed) and the others will have permanent labels (that are bracketed). Let us denote the label of the vertex v by $L(v)$.

Initial Iteration 0

Let V_0 denote the set of all the vertices v_0 of the graph. The starting vertex is assigned the permanent label (0) and all other v_0's the temporary label ∞ each. Let $V_1 = V_0 - \{v_0^*\}$, where v_0^* is the starting vertex which has been assigned a permanent label.

Iteration 1

Let the elements of V_1 be now denoted by v_1. (The elements v_1 are the same as the elements v_0 excluding v_0^*.) For the elements of V_1 that are adjacent to v_0^*, the temporary labels are revised by using $L(v_1) = L(v_0^*) + w(v_0^* v_1)$, where $L(v_0^*) = 0$, $w(v_0^* v_1)$ is the weight of the edge $v_0^* v_1$ and for the other elements of V_1, the previous temporary labels are not altered. Let v_1^* be the vertex among the v_1's for which $L(v_1)$ is minimum. If there is a tie for the choice of v_1^*, it is broken arbitrarily. Now $L(v_1^*)$ is given a permanent label. Let $V_2 = V_1 - \{v_1^*\} \equiv \{v_2\}$.

Iteration i

For the elements of Vi that are adjacent to v_{i-1}^*, the temporary labels are revised by using $L(v_i) = L(v_{i-1}^*) + w(v_{i-1}^* v_i)$ and for the other elements of V_i, the previous temporary labels are not altered. If the temporary label to be assigned to any vertex in the i^{th} iteration is greater than or equal to that assigned to it in the $(i-1)^{th}$ iteration, the previous label is not changed.

The iteration is stopped when the final vertex z is assigned a permanent label eventhough some vertices might not have been assigned permanent labels. The permanent label of z is the length of the shortest path from a to z. The shortest path itself is identified by working backward from z and including those permanently labeled vertices from which the subsequent permanent labels arose.

We will now consider an example and explain Dijkstra's algorithm step by step. Let us assume that the shortest path from the vertex A to the vertex F is required in the weighted graph, given in Fig. 7.58.

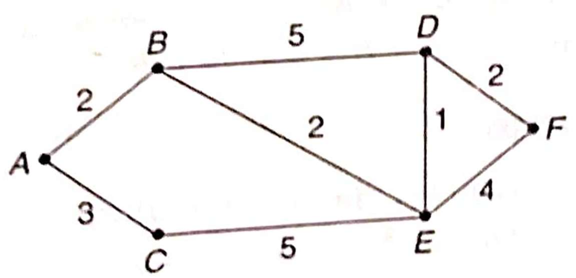

Fig. 7.58

Iteration Number	Details	
0.	V_0: A B C D E F $L(v_0)$: (0) ∞ ∞ ∞ ∞ ∞	Initial labels for all the vertices are assumed. A gets the permanent label and $L(A^*) = 0$ is bracketed.
1.	V_1: A* B C D E F $L(v_i)$: — (2) 3 ∞ ∞ ∞	B and C are adjacent vertices for A^*. $L(B) = L(A^*) + w(A^*B) = 0 + 2 = 2$ $L(C) = L(A^*) + w(A^*C) = 0 + 3 = 3$ Since $L(B) < L(C)$, B gets the permanent label and $L(B^*) = 2$ is bracketed.
2.	V_2: A* B* C D E F $L(v_2)$: — — (3) 7 4 ∞	D and E are adjacent vertices to B^*. $L(D) = L(B^*) + w(B^*D) = 2 + 5 = 7$ $L(E) = L(B^*) = w(B^*E) = 2 + 2 = 4$ Since C is not adjacent to B^*, $L(C)$ is brought forward from the previous iteration as 3. Since $L(C)$ is minimum among $L(C)$, $L(D)$ and $L(E)$, C gets the permanent label and $L(C^*) = 3$ is bracketed.
3.	V_3: A* B* C* D E* F $L(v_3)$: — — — 7 (4) ∞	D and F are not adjacent to C^*. So $L(D)$ and $L(F)$ are brought forward from iteration (2). $L(E) = L(C^*) + w(C^*E) = 3 + 5 = 8$ Since the revised $L(E) >$ the previous $L(E) >$ the previous value of $L(E) = 4$ is retained. Now E gets the permanent label and $L(E^*) = 4$ is bracketed.
4.	V_4: A* B* C* D E* F $L(v_4)$: — — — (5) — 8	D and F are adjacent to E^* $L(D) = L(E^*) + w(E^*D) = 4 + 1 = 5$ $L(F) = L(E^*)\ w(E^*F) = 4 + 4 = 8$ Since $L(D) < L(F)$, D gets the permanent label and $L(D^*) = 5$ is bracketed.
5.	V_5: A* B* C* D* E* F $L(v_5)$: — — — — — (7)	Since F is the only vertex adjacent to D^* and since $L(F) = L(D^*) + w(D^*F) = 5 + 2 = 7$, the final vertex F gets the permanent label and $L(F^*) = 7$ is bracketed.

Since $L(F^*) = 7$, the length of the shortest path from A to $F = 7$.

To find the shortest path, we work backward from F explained as follows: F became F^* from D^* in iteration (5); D became D^* from E^* in iteration (4); E became E^* from B (but not from C), as $L(E) = L(E^*)$ assumed the label 4 in iteration (2) itself; B became B^* from A^* in iteration (1).

Hence, the shortest path is $A - B - E - D - F$.

Warshall's Algorithm

Warshall's algorithm determines the shortest distances between all pairs of vertices in a graph. It is popular because it is easier to describe than the other algorithm and it can be applied to a directed graph too without any change. The algorithm is explained below:

The weight matrix $W = (w_{ij})$ of the given graph is first formed, where

$$w_{ij} = \begin{cases} w(ij), & \text{if there is an edge from } v_i \text{ to } v_j \\ 0\ , & \text{if there is no edge.} \end{cases}$$

Let there be n vertices $v_1, v_2, \ldots, v_n$ in the graph. Now a sequence of matrices $L_0, L_1, \ldots L_n$ are formed, where $L_r = \{l_r(i, j)\}$.

$l_r(ij)$, the ij^{th} entry of L_r is computed by using the rule

$$l_r(i, j) = \min\,[l_{r-1}(i, j);\ l_{r-1}(i, k) + l_{r-1}(k, j)],$$

where k takes the values $1, 2, \ldots, n$ in the first, second, ...n^{th} iterations respectively.

The initial matrix L_0 is the same as the weight matrix W except that each non-diagonal 0 in W is replaced by ∞.

The final matrix L_n is the required shortest (path) distance matrix L the ij^{-th} entry of which gives the length of the shortest path between the vertices v_i and v_j.

Note Warshall's algorithm can be applied to find the shortest distance matrix, in the case of directed pseudograph with loops and parallel edges also. But in this case all 0's are replaced by ∞.

We will now take an example and explain Warshall's algorithm setp by step.

We require the shortest distance matrix for the undirected graph, given in Fig. 7.59.

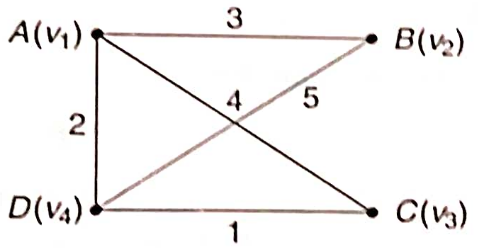

Fig. 7.59

$$W = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{pmatrix} 0 & 3 & 4 & 2 \\ 3 & 0 & 0 & 5 \\ 4 & 0 & 0 & 1 \\ 2 & 5 & 1 & 0 \end{pmatrix} \end{array}; \quad L_0 = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{array} \begin{array}{c} \begin{array}{cccc} v_1 & v_2 & v_3 & v_4 \end{array} \\ \begin{pmatrix} 0 & 3 & 4 & 2 \\ 3 & 0 & \infty & 5 \\ 4 & \infty & 0 & 1 \\ 2 & 5 & 1 & 0 \end{pmatrix} \end{array}$$

By Warshall's algorithm,

$$l_1(1, 2) = \min\{l_0(1, 2);\ l_0(1, 1) + l_0(1, 2)\}$$
$$= \min\{3;\ 0 + 3\} = 3$$
$$l_1(1, 3) = \min\{l_0(1, 3);\ l_0(1, 1) + l_0(1, 3)\}$$
$$= \min\{4;\ 0 + 4\} = 4$$
$$l_1(1, 4) = \min\{l_0(1, 4);\ l_0(1, 1) + l_0(1, 4)\}$$
$$= \min\{2;\ 0 + 2\} = 2$$
$$l_1(2, 3) = \min\{l_0(2, 3);\ l_0(2, 1) + l_0(1, 3)\}$$
$$= \min\{\infty;\ 3 + 4\} = 7$$

$$l_1(2, 4) = \min\{l_0(2, 4); l_0(2, 1) + l_0(1, 4)\}$$
$$= \min\{5; 3 + 2\} = 5$$
$$l_1(3, 4) = \min\{l_0(3, 4); l_0(3, 1) + l_0(1, 4)\}$$
$$= \min\{1; 4 + 2\} = 1$$

Since L_0 is a symmetric matrix, L_1 and the subsequent matrices L_2, L_3 and L_4 will also be symmetric. Using the symmetry, we get

$$L_1 = \begin{pmatrix} 0 & 3 & 4 & 2 \\ 3 & 0 & 7 & 5 \\ 4 & 7 & 0 & 1 \\ 2 & 5 & 1 & 0 \end{pmatrix}$$

Now $$l_2(1, 2) = \min\{l_1(1, 2); l_1(1, 2) + l_1(2, 2)\}$$
$$= \min\{3; 3 + 0\} = 3$$

Similarly proceeding, we get,

$$l_2(1, 3) = 4;\ l_2(1, 4) = 2;\ l_2(2, 3) = 7;\ l_2(2, 4) = 5;\ l_2(3, 4) = 1$$

Hence, $$L_2 = \begin{pmatrix} 0 & 3 & 4 & 2 \\ 3 & 0 & 7 & 5 \\ 4 & 7 & 0 & 1 \\ 2 & 5 & 1 & 0 \end{pmatrix}$$

Proceeding in the same way, we can get,

$$L_3 = \begin{pmatrix} 0 & 3 & 4 & 2 \\ 3 & 0 & 7 & 5 \\ 4 & 7 & 0 & 1 \\ 2 & 5 & 1 & 0 \end{pmatrix} \text{ and } L_4 = \begin{pmatrix} 0 & 3 & 3 & 2 \\ 3 & 0 & 6 & 5 \\ 3 & 6 & 0 & 1 \\ 2 & 5 & 1 & 0 \end{pmatrix}$$

L_4 gives the shortest distances between all pairs of vertices. The corresponding shortest paths are given by the following matrix:

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & — & AB & ADC & AD \\ B & BA & — & BADC & BD \\ C & CDA & CDAB & — & CD \\ D & DA & DB & DC & — \end{array}$$

WORKED EXAMPLES 7(B)

Example 7.1 Find which of the following vertex sequences are simple paths, paths, closed paths (circuits) and simple circuits with respect to the graph shown in Fig. 7.60.

(a) $A - D - E - B - C$
(b) $A - D - B - C - E$
(c) $A - E - C - B - E - A$
(d) $C - B - D - A - E - C$
(e) $A - D - B - E - C - B$

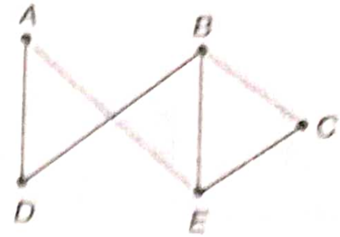

Fig. 7.60

(a) $A - D - E - B - C$ is not a path, since DE is not an edge of the given graph.

(b) $A - D - B - C - E$ is a simple path between the vertices A and E, since the vertices and edges involved are distinct.

(c) $A - E - C - B - E - A$ is a closed path, since the initial and final vertices are the same and the vertex E appears twice.

(d) $C - B - D - A - E - C$ is a simple circuit, since the initial and final vertices are the same and the vertices and edges are distinct.

(e) $A - D - B - E - C - B$ is a path (but not a simple path) as the vertex B appears twice.

Example 7.2 Find all the simple paths from A to F and all the circuits in the graph given in Fig. 7.61.

The simple paths from A to F are the following:

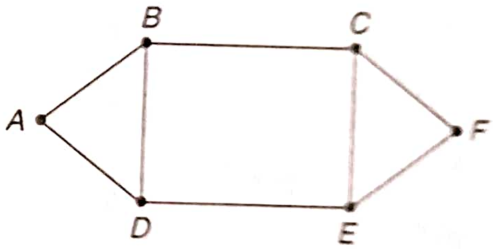

Fig. 7.61

1. $A - B - C - F$;
2. $A - D - E - F$;
3. $A - B - D - E - F$;
4. $A - D - B - C - F$;
5. $A - B - C - E - F$;
6. $A - D - E - C - F$;
7. $A - B - D - E - C - F$; 8. $A - D - B - C - E - F$

The circuits in the graph are the following:

1. $A - B - D - A$; 2. $C - F - E - C$; 3. $B - C - E - D - B$;
4. $A - B - C - E - D - A$; 5. $B - C - F - E - D - B$;
6. $A - B - C - F - E - D - A$.

Example 7.3 Find all connected subgraphs of the graph shown in Fig. 7.62 containing all of the vertices of the original graph and having as few edges as possible. In these subgraphs which are paths and simple paths from A to G?

The graphs in Figs 7.62(a), 7.62(b) and 7.62(c) are the connected subgraphs required. However they are not connected components of the original graph in

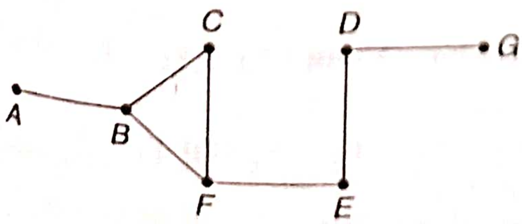

Fig. 7.62

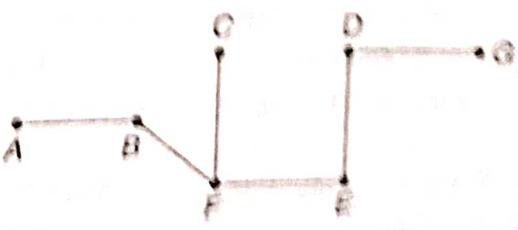

Fig. 7.62(a)

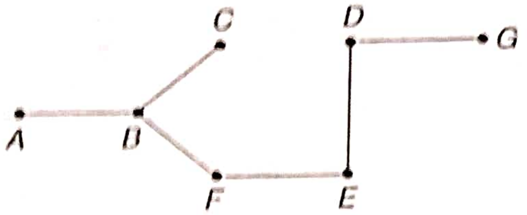

Fig. 7.62(b)

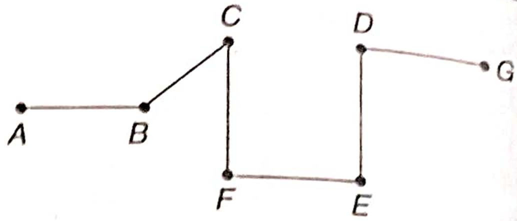

Fig. 7.62(c)

Fig. 7.62. In Fig. 7.62(a), $A - B - F - E - D - G$ is a simple path from A to G, whereas $A - B - F - C - F - E - D - G$ is a path from A to G.

In Fig. 7.62(b), $A - B - F - E - D - G$ is a simple path, whereas $A - B - C - B - F - E - D - G$ is a path. In Fig. 7.62(c), $A - B - C - F - E - D - G$ is a simple path containing all the vertices of the original graph.

There are no closed paths and circuits in the subgraphs, whereas they are present in the original graph.

Example 7.4 Using circuits, examine whether the following pairs of graphs G_1 and G_2 given in Figs 7.63 and 7.64 are isomorphic or not.

(a)

Vice
9307

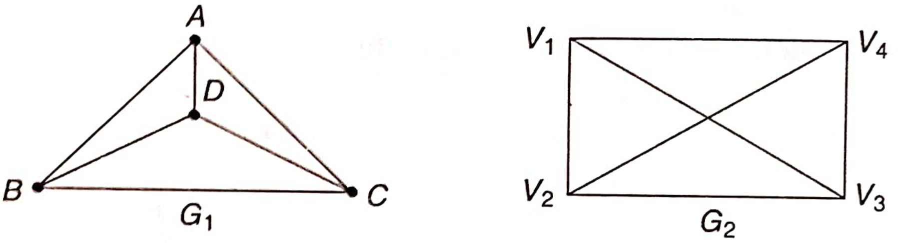

Fig. 7.63

(b)

copy

at . . .

o. 491

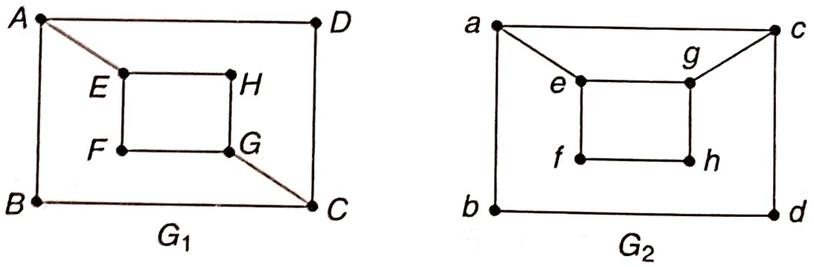

Fig. 7.64

(a) G_1 and G_2 have 4 vertices each and 6 edges each. Also all the 4 vertices in both the graphs are of degree 3 each. Hence, the necessary conditions for isomorphism are satisfied.

Now $A - B - D - A$, $A - C - D - A$ and $A - B - C - A$ are circuits of length 3 each in G_1.

Also $A - B - C - D - A$, $A - B - D - C - A$ and $A - D - B - C - A$ and circuits of length 4 each in G_1

Similarly $V_1 - V_2 - V_4 - V_1$, $V_1 - V_3 - V_4 - V_1$ and $V_1 - V_2 - V_3 - V_1$ are circuit of length 3 each in G_2.

Also $V_1 - V_2 - V_3 - V_4 - V_1$, $V_1 - V_2 - V_4 - V_3 - V_1$ and $V_1 - V_4 - V_2 - V_3 - V_1$ are circuits of length 4 each in G_2.

Hence, the two graphs G_1 and G_2 are isomorphic.

(b) G_1 and G_2 have 8 vertices each and 10 edges each.

Also there are 4 vertices each of degree 3 and 4 vertices each of degree 2 in G_1 and G_2.

Hence, the conditions necessary for isomorphism are satisfied.

Now there is only one circuit of length 4 from A to A, viz., $A - B - C - D - A$ in G_1, but there are two circuits of length 4 each from a to a, namely, $a - b - d - c - a$ and $a - e - g - c - a$.

Hence, the two graphs G_1 and G_2 are not isomorphic.

Example 7.5 Find the number of paths of length 4 from the vertex D to the vertex E in the undirected graph shown in Fig. 7.65 analytically. Identify those paths from the graphs.

The adjacency matrix of the given graph is

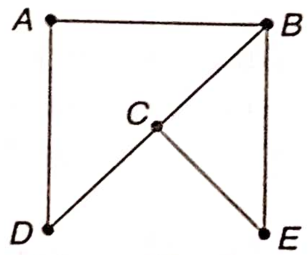

Fig. 7.65

$$A = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

By matrix multiplication, $A^2 = \begin{bmatrix} 2 & 0 & 2 & 0 & 1 \\ 0 & 3 & 1 & 2 & 1 \\ 2 & 1 & 3 & 0 & 1 \\ 0 & 2 & 0 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 \end{bmatrix}$

Again, by matrix multiplication, we get

$$A^4 = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{bmatrix} 9 & 3 & 11 & 1 & 6 \\ 3 & 15 & 7 & 11 & 8 \\ 11 & 7 & 15 & 3 & 8 \\ 1 & 11 & 3 & 9 & 6 \\ 6 & 8 & 8 & 6 & 8 \end{bmatrix} \end{array}$$

The entry in the $(4 - 5)^{th}$ position of A_4 is 6.

Hence, there are 6 paths each of length 4 from D to E.

Those 6 paths identified from the given graphs are as follows:

1. $D - A - D - C - E$; 2. $D - C - D - C - E$; 3. $D - A - B - C - E$;
4. $D - C - E - C - E$; 5. $D - C - E - B - E$; 6. $D - C - B - C - E$.

Example 7.6 Find the number of paths of length 4 from the vertex B to the vertex D in the directed graph shown in Fig. 7.66 analytically. Name those paths using the graph.

The adjacency matrix of the given graph is

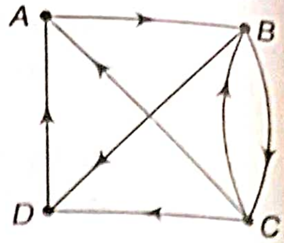

Fig. 7.66

$$A = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array}\begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

By matrix multiplication, we get

$$A^2 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Again by matrix multiplication, we get

$$A^4 = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array}\begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 1 & 2 & 1 & 1 \\ 2 & 2 & 3 & 3 \\ 3 & 3 & 2 & 3 \\ 2 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

The entry in the (BD) position of A^4 is 3. Hence, there are 3 paths each of length 4 from B to D.

They are (i) $B - C - B - C - D$, (2) $B - C - A - B - D$ and (3) $B - D - A - B - D$.

Example 7.7 Find which of the following graphs given in Fig. 7.67 is strongly, weakly or unilaterally connected. Give the reasons.

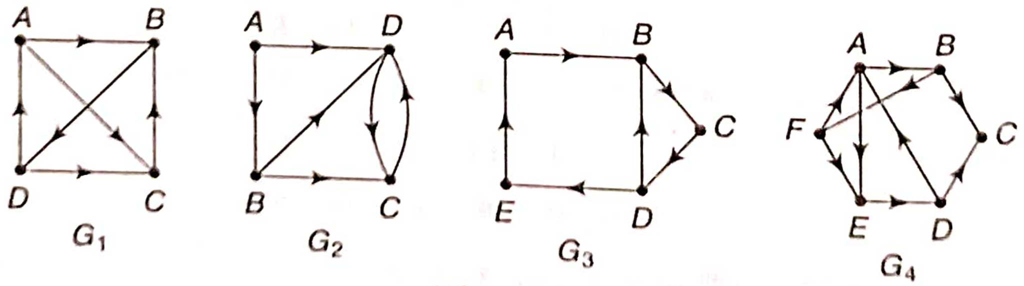

Fig. 7.67

(i) G_1 is strongly connected, since there is a path from each of the possible pairs of vertices, namely, (A, B), (A, C), (A, D), (B, C), (B, D) and (C, D), to the other are as follows:

A and B: $A - B$ and $B - D - A$

A and C: $A - C$ and $C - B - D - A$

A and D: $A - B - D$ and $D - A$

B and C: $B - D - C$ and $C - B$
B and D: $B - D$ and $D - A - B$
C and D: $C - B - D$ and $D - C$

(ii) G_2 is unilaterally connected since there is one-way path only for the 5 of the 6 possible pairs of vertices as given below:

A and B: $A - B$ and no path from B to A
A and C: $A - D - C$ and no path from C to A
A and D: $A - D$ and no path from D to A
B and C: $B - C$ and no path from C to B
B and D: $B - D$ and no path from D to B
C and D: $C - D$ and $D - C$

(iii) G_3 is not strongly connected, since there are no paths from A to the other 4 vertices. However there is one-way path only for some of the 10 possible pairs of vertices. Hence, G_3 is unilaterally connected and also weakly connected.

(iv) G_4 is unilaterally connected, since there is no path from C to the other vertices, but C can be reached from them.

Example 7.8 Find the strongly connected components of the graph shown in Fig. 7.68.

The strongly connected components of the given graph are $ABHI$ and $CDFG$, since they are strongly connected and they are not contained in larger strongly connected subgraphs.

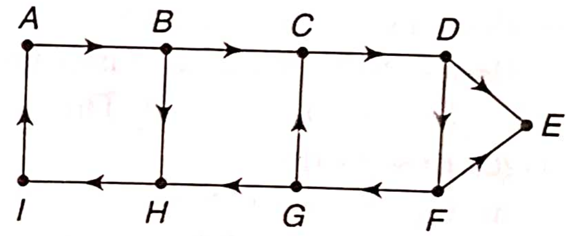

Fig. 7.68

Example 7.9 Explain *Konisberg bridge problem.* Represent the problem by means of graph. Does the problem have a solution?

There are two islands A and B formed by a river. They are connected to each other and to the river banks C and D by means of 7 bridges as shown in Fig. 7.69. The problem is to start from any one of the 4 land areas A, B, C, D, walk across each bridge exactly once and return to the starting point.

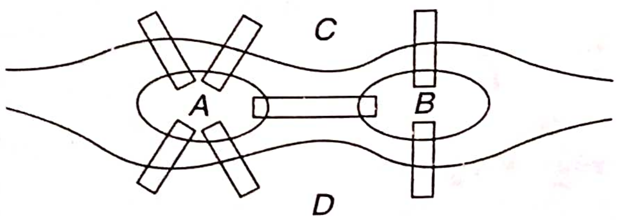

Fig. 7.69

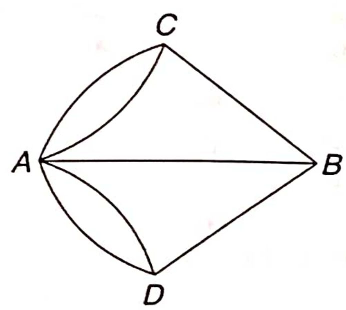

Fig. 7.70

This problem is the famous Konisberg bridge problem.

When the situation is represented by a graph, with vertices representing the land areas and the edges the bridges, the graph will be as shown in Fig. 7.70.

This problem is the same as that of drawing the graph in Fig. 7.70 without lifting the pen from the paper and without retracing any line.

In other words, the problem is to find whether there is an Eulerian circuit (viz., a simple circuit containing every edge) in the graph. But a connected graph has an Eulerian circuit if and only if each of its vertices is of even degree.

In the present case all the vertices are of odd degree. Hence, Konisberg bridge problem has no solution.

Example 7.10 Find an Euler path or an Euler circuit, if it exists in each of the three graphs in Fig. 7.71. If it does not exist, explain why?

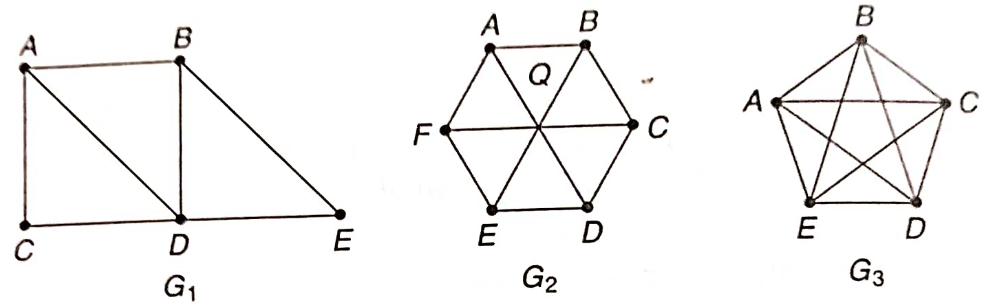

Fig. 7.71

In G_1, there are only two vertices, namely, A and B of degree 3 and other vertices are of even degree.

Hence, there exists an Euler path between A and B. The actual path is $A - B - E - D - A - C - D - B$. This is an Eulerain path, as it includes each of the 7 edges exactly once.

In G_2, there are 6 vertices of odd degree. Hence, G_2 contains neither an Euler path nor an Euler circuit.

In G_3, all the vertices are of even degree. Hence, there exists an Euler circuit in G_3.

It is $A - B - C - D - E - A - C - E - B - D - A$. This circuit is Eulerian, since it includes each of the 10 edges exactly once.

Example 7.11 Find a Hamiltonian path or a Hamiltonian circuit, if it exists in each of the three graphs in Fig. 7.72. If it does not exist, explain why?

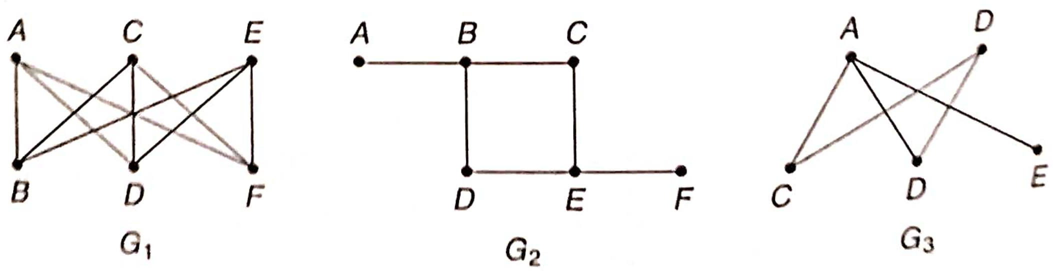

Fig. 7.72

G_1 contains a Hamiltonian circuit, for example $A - B - C - D - E - F - A$. In fact there are 5 more Hamiltonian circuits in G_1, namely, $A - B - C - F - E - D - A$, $A - B - E - D - C - F - A$, $A - B - E - F - C - D - A$, $A - D - C - B - E - F - A$ and $A - D - E - B - C - F - A$.

G_2 contains neither a Hamiltonian path nor a Hamiltonian circuit, since any path containing all the vertices must contain one of the edges $A - B$ and $E - F$ more than once.

G_3 contains 2 Hamiltonian paths from C to E and from D to E, namely, $C - B - D - A - E$ and $D - B - C - A - E$, but no Hamiltonian circuits.

Example 7.12 Give an example of a graph which contains
(i) an Eulerian circuit that is also a Hamiltonian circuit
(ii) an Eulerian circuit and a Hamiltonian circuit that are distinct
(iii) an Eulerian circuit, but not a Hamiltonian circuit
(iv) a Hamiltonian circuit, but not an Eulerian circuit
(v) neither an Eulerian circuit nor a Hamiltonian circuit.

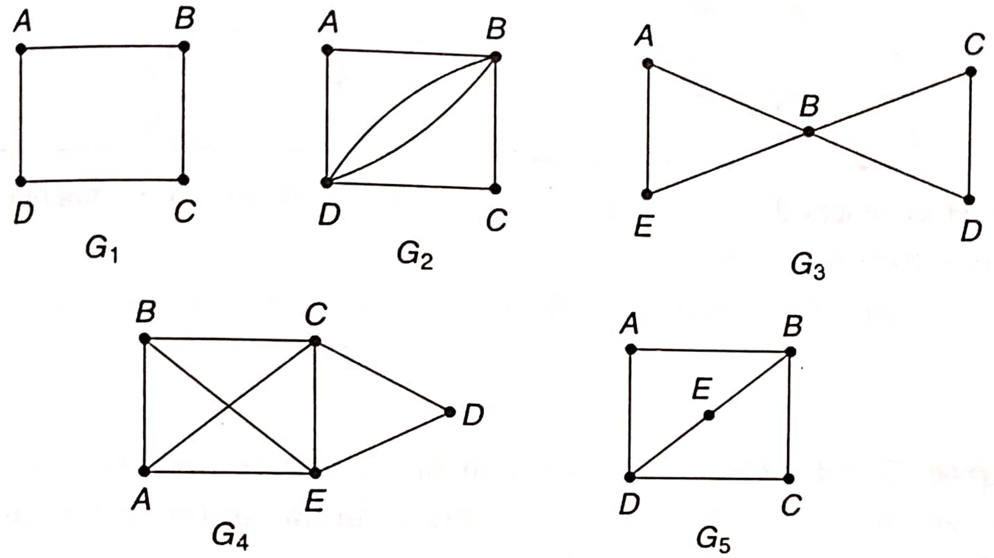

Fig. 7.73

(i) The circuit $A - B - C - D - A$ in G_1 consists of all edges and all vertices, each exactly once.
$\therefore$ G_1 contains a circuit that is both Eulerian and Hamiltonian.

(ii) G_2 contains the Eulerian circuit $A - B - D - B - C - D - A$ and the Hamiltonian circuit $A - B - C - D - A$, but the two circuits are different.

(iii) G_3 contains the Eulerian circuit $A - B - C - D - B - E - A$, but this circuit is not Hamiltonian, as the vertex B is repeated twice.

(iv) G_4 contains the Hamiltonian circuit $A - B - C - D - E - A$. However, it does not contain Eulerian circuit as there are 4 vertices each of degree 3.

(v) In G_5, degree of B and degree of D are each equal to 3. Hence, there is no Euler circuit in it. Also no circuit passes through each of the vertices exactly once.

Example 7.13 Use Dijkstra's algorithm to find the shortest path between the vertices A and H in the weighted graph given in Fig. 7.74.

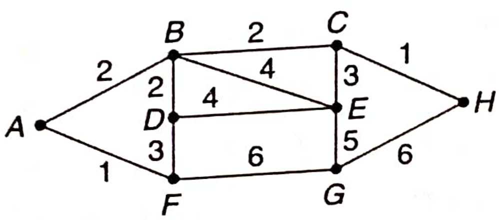

Fig. 7.74

Number	Dijkstra's Iteration: Details of V and $L(v)$									Adjacent vertices of latest v^*
0.	V_0 :	A		C	D	E	F	G	H	B and F
	$L(v_0)$:	(0)		∞	∞	∞	∞	∞	∞	
1.	V_1 :	A^*	B	C	D	E	F	G	H	D and G
	$L(v_1)$:	—	2	∞	∞	∞	(1)	∞	∞	
2.	V_2 :	A^*	B	C	D	E	F^*	G	H	C, D and E
	$L(v_2)$:	—	(2)	∞	4	∞	—	7	∞	
3.	V_3 :	A^*	B^*	C	D	E	F^*	G	H	E and H
	$L(v_3)$:	—	—	(4)	4	6	—	∞	∞	
4.	V_4 :	A^*	B^*	C^*	D	E	F^*	G	H	—
	$L(v_4)$:	—	—	—	∞	7	—	∞	(5)	

Since H is reached from C, C is reached from B and B is reached from A, the shortest path is $A - B - C - H$.

$$\begin{aligned} \text{Length of the shortest path} &= w(AB) + w(BC) + w(CH) \\ &= 2 + 2 + 1 \\ &= 5. \end{aligned}$$

Example 7.14 Find the shorted distance matrix and the corresponding shortest path matrix for all the pairs of vertices in the undirected graph given in Fig. 7.75, using Warshall's algorithm.

The weight matrix of the given graph is given by

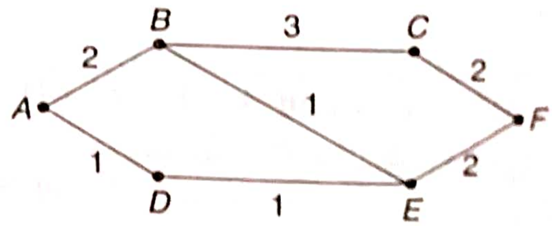

Fig. 7.75

$$W = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{pmatrix} 0 & 2 & 0 & 1 & 0 & 0 \\ 2 & 0 & 3 & 0 & 1 & 0 \\ 0 & 3 & 0 & 0 & 0 & 2 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 & 2 & 0 \end{pmatrix} \end{array}$$

The initial distance (length) matrix L_0 is got from W by replacing all the non-diagonal 0's by ∞ each. Thus

$$L_0 = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{pmatrix} 0 & 2 & \infty & 1 & \infty & \infty \\ 2 & 0 & 3 & \infty & 1 & \infty \\ \infty & 3 & 0 & \infty & \infty & 2 \\ 1 & \infty & \infty & 0 & 1 & \infty \\ \infty & 1 & \infty & 1 & 0 & 2 \\ \infty & \infty & 2 & \infty & 2 & 0 \end{pmatrix} \end{array}$$

Since all the L_r matrices are symmetric with zero diagonal elements, we need only compute the following elements in the successive L_r matrices:

$l_{12}, l_{13}, l_{14}, l_{15}, l_{16};\quad l_{23}, l_{24}, l_{25}, l_{26};\quad l_{34}, l_{35}, l_{36};\quad l_{45}, l_{46}$ and l_{56}

For the L_1 matrix, the above elements are given by

$$l_{ij} = \min [l_{ij};\ l_{i1} + l_{ij} \text{ of the } L_0 \text{ matrix}]$$

Thus,

$$\begin{aligned}
l_{12} \text{ of } L_1 &= \min [l_{12};\ l_{11} + l_{12} \text{ of } L_0] \\
&= \min [2;\ 0 + 2] = 2 \text{ and so on.} \\
l_{23} \text{ of } L_1 &= \min [l_{23};\ l_{21} + l_{13} \text{ of } L_0] \\
&= \min [3;\ 2 + \infty] = 3 \text{ and so on.} \\
l_{34} \text{ of } L_1 &= \min [l_{34};\ l_{31} + l_{14} \text{ of } L_0] \\
&= \min [\infty;\ \infty + 1] = \infty \text{ and so on.} \\
l_{45} \text{ of } L_1 &= \min [l_{45};\ l_{41} + l_{15} \text{ of } L_0] \\
&= \min [1;\ 1 + \infty] = 1 \text{ and so on.} \\
l_{56} \text{ of } L_1 &= \min [l_{56};\ l_{51} + l_{16} \text{ of } L_0] \\
&= \min [2;\ \infty + \infty] = 2 \text{ and so on.}
\end{aligned}$$

Hence,

$$L_1 = \begin{pmatrix} 0 & 2 & \infty & 1 & \infty & \infty \\ 2 & 0 & 3 & 3 & 1 & \infty \\ \infty & 3 & 0 & \infty & \infty & 2 \\ 1 & 3 & \infty & 0 & 1 & \infty \\ \infty & 1 & \infty & 1 & 0 & 2 \\ \infty & \infty & 2 & \infty & 2 & 0 \end{pmatrix}$$

Proceeding like this, the required elements of L_r matrix are obtained by using the rule l_{ij} of $L_r = \min [l_{ij};\ l_{ir} + l_{rj}$ of $L_{r-1}]$, where $r = 2, 3, 4, 5, 6$.

Accordingly, the successive matrices are given by:

$$L_2 = \begin{pmatrix} 0 & 2 & 5 & 1 & 3 & \infty \\ 2 & 0 & 3 & 3 & 1 & \infty \\ 5 & 3 & 0 & 6 & 4 & 2 \\ 1 & 3 & 6 & 0 & 1 & \infty \\ 3 & 1 & 4 & 1 & 0 & 2 \\ \infty & \infty & 2 & \infty & 2 & 0 \end{pmatrix}; \quad L_3 = \begin{pmatrix} 0 & 2 & 5 & 1 & 3 & 7 \\ 2 & 0 & 3 & 3 & 1 & 5 \\ 5 & 3 & 0 & 6 & 4 & 2 \\ 1 & 3 & 6 & 0 & 1 & 8 \\ 3 & 1 & 4 & 1 & 0 & 2 \\ 7 & 5 & 2 & 8 & 2 & 0 \end{pmatrix}$$

$$L_4 = \begin{pmatrix} 0 & 2 & 5 & 1 & 2 & 7 \\ 2 & 0 & 3 & 3 & 1 & 5 \\ 5 & 3 & 0 & 6 & 4 & 2 \\ 1 & 3 & 6 & 0 & 1 & 8 \\ 2 & 1 & 4 & 1 & 0 & 2 \\ 7 & 5 & 2 & 8 & 2 & 0 \end{pmatrix}; \quad L_5 = \begin{pmatrix} 0 & 2 & 5 & 1 & 2 & 4 \\ 2 & 0 & 3 & 2 & 1 & 3 \\ 5 & 3 & 0 & 5 & 4 & 2 \\ 1 & 2 & 5 & 0 & 1 & 3 \\ 2 & 1 & 4 & 1 & 0 & 2 \\ 4 & 3 & 2 & 3 & 2 & 0 \end{pmatrix}$$

$$L_6 = \begin{array}{c} \\ A \\ B \\ C \\ D \\ E \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} A & B & C & D & E & F \end{array} \\ \begin{pmatrix} 0 & 2 & 5 & 1 & 2 & 4 \\ 2 & 0 & 3 & 2 & 1 & 3 \\ 5 & 3 & 0 & 5 & 4 & 2 \\ 1 & 2 & 5 & 0 & 1 & 3 \\ 2 & 1 & 4 & 1 & 0 & 2 \\ 4 & 3 & 2 & 3 & 2 & 0 \end{pmatrix} \end{array}$$

L_6 is the required shortest distance matrix that gives the shortest distance between all pairs of vertices of the given graph. The corresponding shortest path matrix is as follows:

$$\begin{array}{c|cccccc} & A & B & C & D & E & F \\ \hline A & — & AB & ABC & AD & ADE & ADEF \\ B & BA & — & BC & BED & BE & BEF \\ C & CBA & CB & — & CFED & CFE & CF \\ D & DA & DEB & DEFC & — & DE & DEF \\ E & EDA & EB & EFC & ED & — & EF \\ F & FEDA & FEB & FC & FED & FE & — \end{array}$$