5.3 SHORTEST DISTANCES

5.3.1. Introduction.

In this chapter we present two powerful techniques of systemetically traversing the edges of a graph such that every edge and every vertex of the graph is visited, technically called *Searched* or *Processed*.

In previous Chapter we introduced the conception of path between two vertices of a graph. In a Remark we mentioned about the path in Di-graph also. Though there exist several possible paths between two vertices the path having shortest length has most importance in Computer science and Operation Research. There are several method of finding this shortest path. These methods are designed as Algorithm. Among these Disjkart's Algorithm is discussed in this chapter.

5.3.2. Weight of an edge and Weighted Graph.

Sometimes a real number is associated with each edge of a graph or di-graph ; this number represents weight or distance of the corresponding edge. A graph (or Di-graph) each of whose edges bears a weight is called weighted graph (Di-graph).

Illustration. The graph shown in Fig. 5.3.1 is a weighted graph.Here the vertices A, B, C, ... etc. represent the communication centre of a company.

The edge say (AB) represents the communication line (may be telephone line, Fax line etc.) from the centre A to centre B. Let the cost of communication from A to B be Rs. 100 per hour.

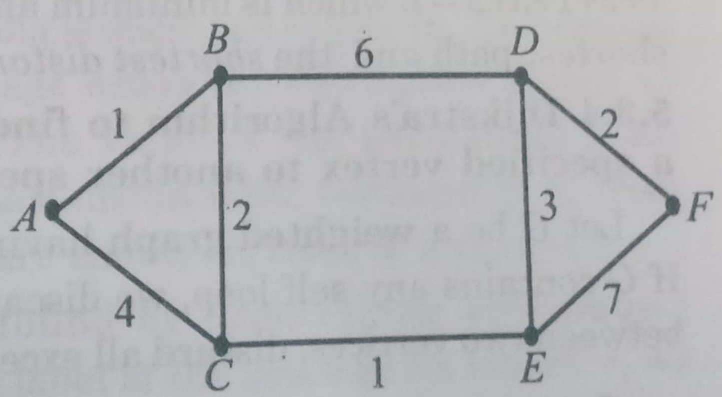


Fig. 5.3.1

We suppose the edge (AB) bears a weight 1. Similarly other edges bear the corresponding communication cost as their weight.

5.3.3. Shortest path between two vertices.

For Unweighted Graph : Let G be a graph ; u, v be two vertices in G. There may exists several paths connecting u and v.Among those the path containing minimum number of edges is called shortest path between u and v in G.

In Fig.5.3.2 we see there are many paths connecting the two vertices B and E. Among those paths the path $\{BG, GE\}$ has minimum numbers of edges. So this path can be treated as shortest path (of length 2) between B and E.

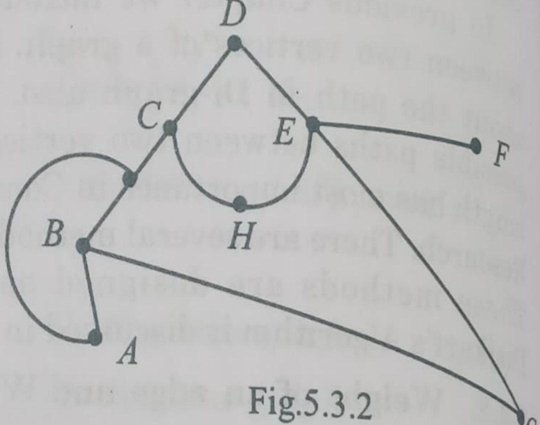


Fig.5.3.2

For Weighted Graph :

Let G be a weighted Graph; u, v be two vertices in G. The sum of the weights of all the edges in a path is called *weight of the path*. Among all paths connecting u and v, the path having, the minimum weight is called shortest path between u and v in the graph. In Fig 5.3.1 there exist several paths from the vertex A to F having several weights. Among these the path $\{A, B, C, E, D, F\}$ has weight 1+2+1+3+2 = 9 which is minimum among all. So this path is the shortest path and the *shortest distance* between A and F is 9.

5.3.4 Dijkstra's Algorithm to find the shortest path from a specified vertex to another specified vertex.

Let G be a **weighted graph** having vertices $v_1, v_2, v_3, \cdots, v_n$ If G contains any self loop, we discard them. If G has parallels between two vertices, discard all except that having least weight.

Let w_{ij} = weight of the edge $(v_i, v_j) \geq 0$

We suppose w_{ii} = weight of the edge $(v_i, v_i) = 0$

$w_{ij} = \infty$ if there is no edge connecting v_i and v_j (e.g. $w_{24} = \infty$ in Fig 5.3.3)

Let we have to find the shortest path from the vertex v_k to v_p. This is a process of iteration, i.e. we find the required path through some consecutive stages. At each stage the vertices of the graph are labelled.

We assign a permanent label to a vertex and temporary label to other vertices. Parmanent label is not changed afterward whereas temporary label goes on changing from stage to stage. The algorithm begins with the first stage where the starting vertex v_k is parmanently labelled 0 and the remaining $n-1$ vertices are temporarily labelled ∞. From then on, at each subsequent stage a new vertex is permanently labelled and the temporary label of others are changed according to the following rule :

Each vertex say v_i which is not yet parmanently labelled gets a new temporary label which is equal to min [label of v_i at the precceding stage, permanent label of the vertex $v_j + w_{ij}$] where v_j is the vertex which was permanently labelled at the preceeding stage.

The smallest value among all the temporary labels is found. Let the vertex v_r has this smallest value. Then this smallest value becomes the permanent label of v_r. This is the shortest distance of v_r from the starting vertex v_k. In case of a tie we select any one of the vertices to be permanently labelled.

The stage to stage labelling is displayed in a table. This process of labelling stops at the stage where the destination vertex v_p gets a permanent label. The value of the permanent label obtained by v_p is the shortest distance (or, distance) of v_p from v_k. The permanent labels are shown enclosed in a square.

The shortest path is found by backtraking technique : Starting at the permanent label of the destination vertex v_p we go back along the previously assigned temporary label of v_p until we get a change. Next we move to the vertex which is newly labelled permanently. Then do a similar backtrack along its previously assigned temporary labels until we find a change, and so on. The vertices found in this way give us the shortest path.

To be more precise see the following Illustration.

Illustrative Example.

Ex. 1. *By "Dijkstra's procedure" find the shorted path and the length of the shortest path from the vertex v_2 to v_5 in the following graph :* [WBUT 2012]

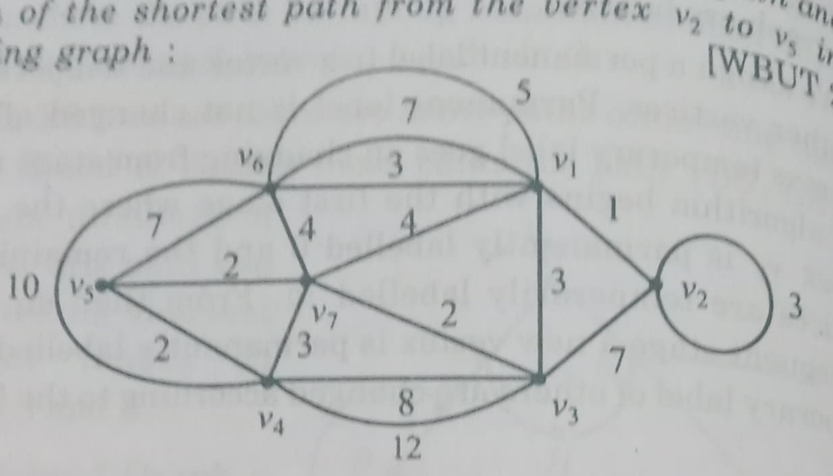


Fig.5.3.3

The given graph has one loop (at v_2) and 3 parallels (connecting v_1, v_6) and 2 parallels connecting v_3 and v_4. We discard the loop and the two parallels having weight 5 and 7 (because the least weight among those of the parallels is 3). Similarly the edge (v_3v_4) having weight 12 is deleted.

The graph becomes :

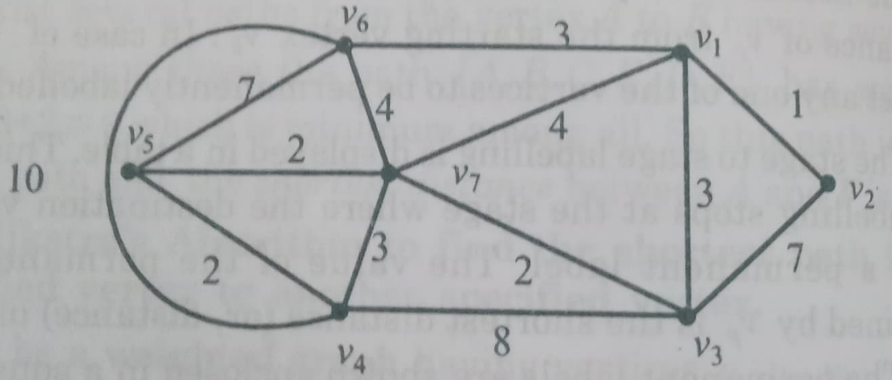


Fig.5.3.4

Here v_2 is the starting vertex and v_5 is the terminating vertex.

The algorithm is displayed in the following table. [Explanations of action taken on each row is given in 'Remark' column]

Dijkstra's Algorithm of finding shortest path from v_2 to v_5.

v_1	v_2	v_3	v_4	v_5	v_6	v_7	Remark
∞	$\boxed{0}\ \checkmark$	∞	∞	∞	∞	∞	Starting vertex v_2 is permanently labelled 0 ; others are temporarily labelled ∞
$\min(\infty, 0+1) = \boxed{1}\ \checkmark$	$\boxed{0}$	$\min(\infty, 0+7) = 7$	$\min(\infty, 0+\infty) = \infty$	$\min(\infty, 0+\infty) = \infty$	$\min(\infty, 0+\infty) = \infty$	$\min(\infty, 0+\infty) = \infty$	v_1 is permanently labelled because it gets minimum of all in this row.
$\boxed{1}$	$\boxed{0}$	$\min(7, 1+3) = 4$	$\min(\infty, 1+\infty) = \infty$	$\min(\infty, 1+\infty) = \infty$	$\min(\infty, 1+3) = \boxed{4}\ \checkmark$	$\min(\infty, 1+4) = 5$	A tie between the temporary label of v_3 and v_6 occurs Select any one.
$\boxed{1}$	$\boxed{0}$	$\min(4, 4+\infty) = \boxed{4}\ \checkmark$	$\min(\infty, 4+10) = 14$	$\min(\infty, 4+7) = 11$	$\boxed{4}$	$\min(5, 4+4) = 5$	v_3 is permanently labelled 4, all others are labelled temporarily
$\boxed{1}$	$\boxed{0}$	$\boxed{4}$	$\min(14, 4+8) = 12$	$\min(11, 4+\infty) = 11$	$\boxed{4}$	$\min(5, 4+2) = \boxed{5}\ \checkmark$	v_7 is permanently labelled
$\boxed{1}$	$\boxed{0}$	$\boxed{4}$	$\min(12, 5+3) = 8$	$\min(11, 5+2) = \boxed{7}\ \checkmark$	$\boxed{4}$	$\boxed{5}$	We stop here because the destination vertex v_5 is permanently labelled.

Since the permanent label of the destination vertex v_5 is 7, so the shortest distance (i.e. distance) from v_2 to v_5 is 7.

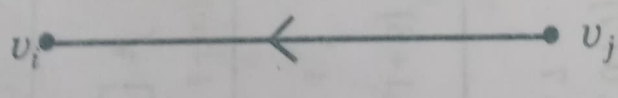
Now we are going to find the shortest path. This is done by the following backtracking technique :

Starting at the permanent label of v_5 (i.e.[7]) we go back along the previously assigned temporary label of v_5 until we get a change.

So we reach the temporary label 11 (at 5th row,). In that row the newly permanent labelled vertex is v_7. Again we go back along the previously assigned temporary label of v_7 until we get a change. We reach the temporary label ∞ (at 2nd row). In that row the recent permanent labelled vertex is v_1 .Similarly tracking back we reach the vertex v_2. So the shortest path is $\{v_2, v_1, v_7, v_5\}$.

Note. (1) *If there were no loop or parallels in the given graph we would start from the graph given in Fig.4.1.4.*

(2) *Dijkstra's Algorithm is applicable for digraph also. There we consider weight $w_{ij} = \infty$ if the edge (v_i, v_j) is directed from v_j to v_i only, as shown in the following figure. In case of a Di-graph $w_{ij} \neq w_{ji}$ whereas in the case of a un-directed graph $w_{ij} = w_{ji}$.*



(3) *If the graph is not weighted assume all the weights $w_{ij} = 1$ and proceed as usual.*

(4) *There may exist more than one shortest path between two vertices in a graph. At the time of permanently labelling if there occur a tie among some temporary label, more than one shortest path may occur.*

(5) *The validity of Dijkstra's Algorithm can be proved in the following two theorems.*

Theorem 1.

In Dijkstra's Algorithm if at some stage the permanent label of a vertex v_i is finite then there is a path from the starting vertex to v_i whose length is equal to the value of that permanent label.

Proof: Beyond the scope of the book.

Theorem 2.

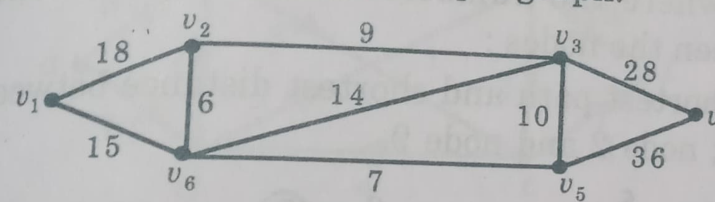
In Dijkstra's Algorithm when a vertex v_i is chosen at the time of permanently lebelling the permanent label has value $\delta(v_i)$.

Proof: Beyond the scope of the book.

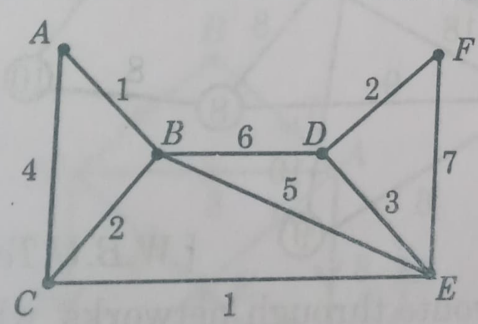
Exercise

[I] Long Answer Questions

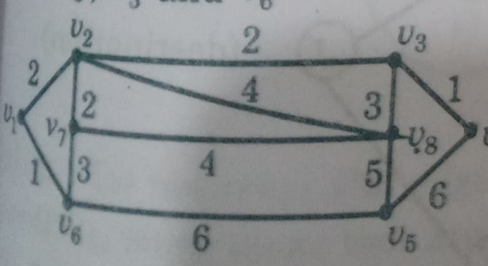
1. Applying Dijkstra's Algorithm find the shortest path from the vertex v_1 to v_4 in the following simple graph.



2. Applying Dijkstra's Algorithm find the shortest path and shortest distance from A to F in the following graph.



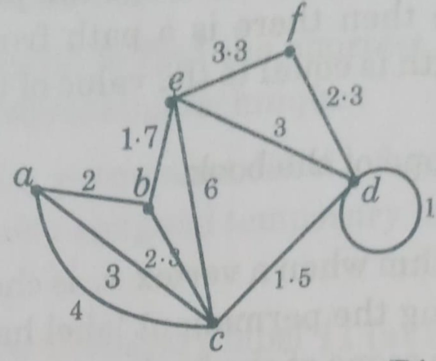
3. Use Dijkstra's Algorithm to find the shortest path between the vertices v_1 and v_4 in the graph (ii) Floyd Alogorithm to find the shortest distance and shortest path between v_1 and v_4; v_2 and v_5; v_3 and v_6 [W.B.U.T. 2013]



Can it have other shortest path.

[Hints : For 2nd part : Follow Note (4) in Art 4.1.4]

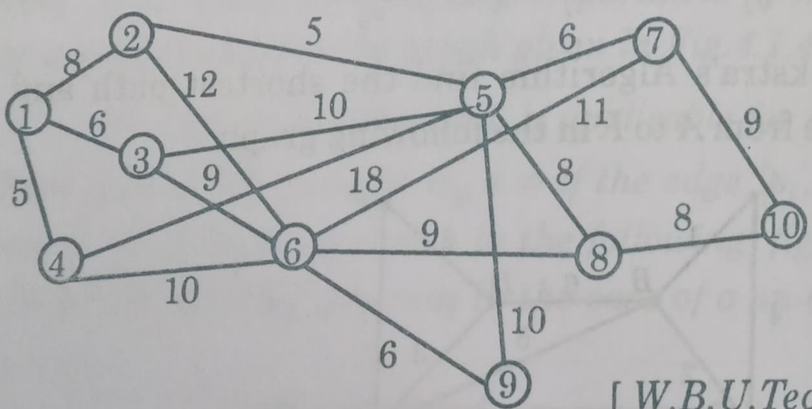
4. Applying Dijkstra's method find the shortest path between the two vertices a and f in the graph : [W.B.U.T. 2005]



[Hint : This has parallels and loops. Discard the edge (a, c) having weight 4 and loop having weight 1.]

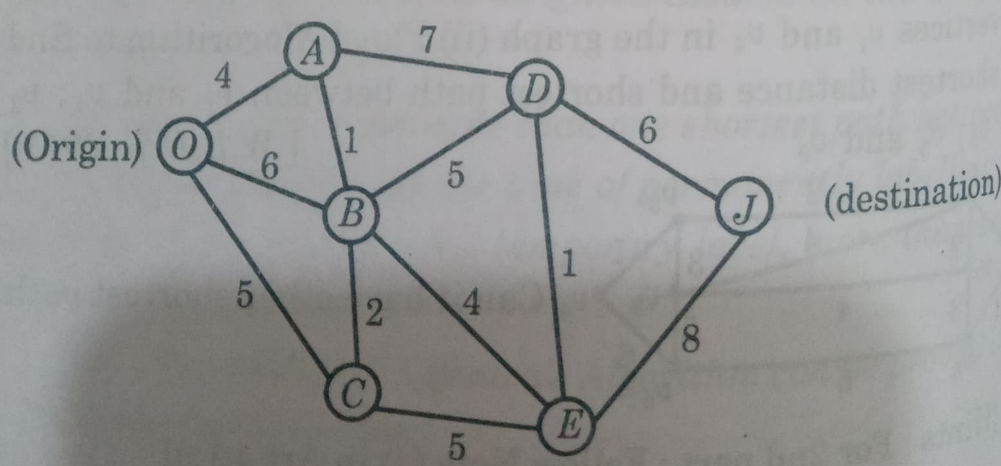
5. (i) Find the shortest path, by Dijkstra's Algorithm, and the minimum route length in the following network form node 1 to the node 10, where the numbers beside the line denote the distance between the nodes :

(ii) Find the shortest path and shortest distance between node 2 and node 10; node 2 and node 9.



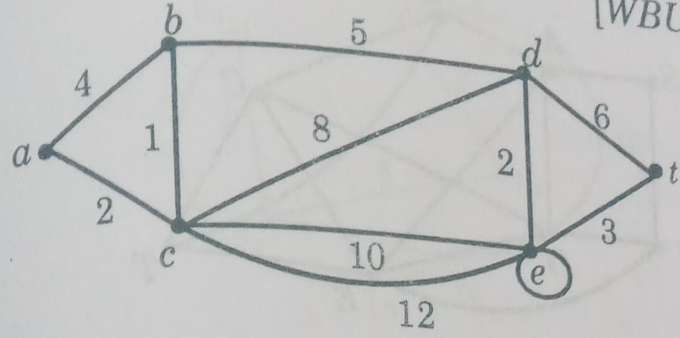
[W.B.U.Tech 2004 MCA]

6. Find the shortest route through networks, where the weight represents the actual distance between the corresponding nodes :

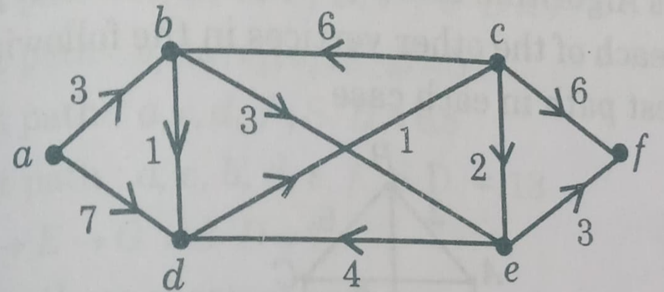


7. Using Dijkstra's Algorithm find the length of the shortest path of the following graph from the vertex a to t :

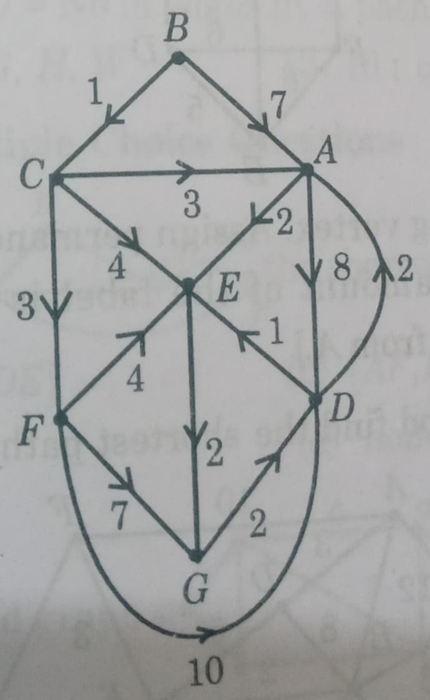
[WBUT 2011, 2006]



8. Using Dijkstra's Algorithm find the shortest path from the vertex a to f in the following Di-graph :

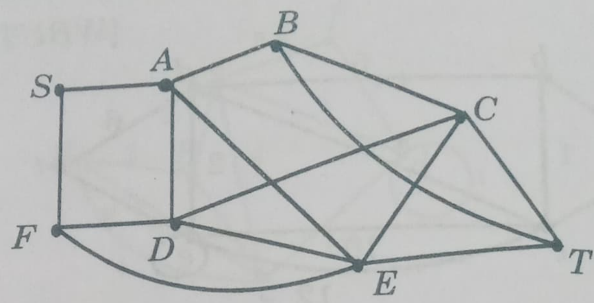


9. Using Dijkstra's Algorithm find the shortest path from the vertex B to G in the following Di-graph :



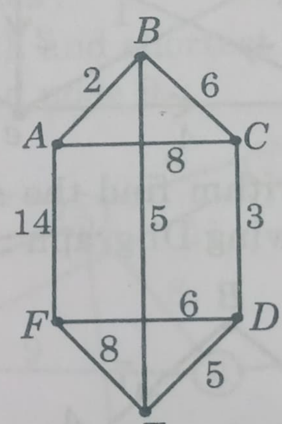
[Hint : Do not treat the edges AD of wt 8 and DA of wt 2 as parallels, since these two are different in a di-graph. see Note (2), in Art 4.1.4]

10. Use Dijkstra's Algorithm to find the shortest path between the two vertices S and T in the following un-weighted graph :



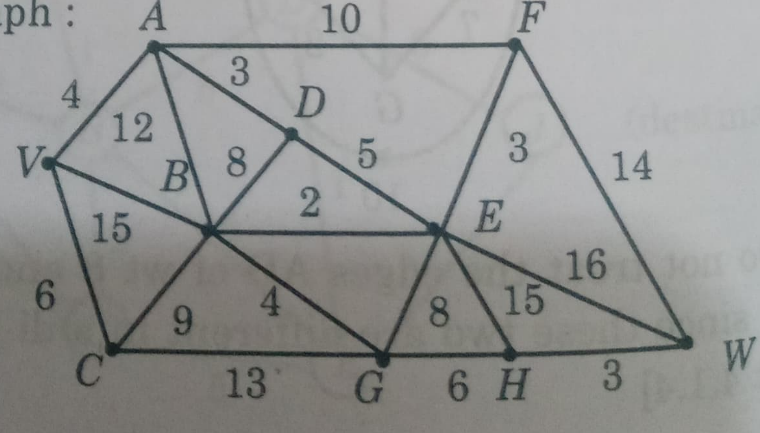
[Hint: Assume all the weights, $w_{ij} = 1$ and proceed as prescribed in Dijkstra's Algorithm]

11. By Dijkstra's Algorithm find the length of shortest path from the vertex A to each of the other vertices in the following graph. Show the shortest path in each case

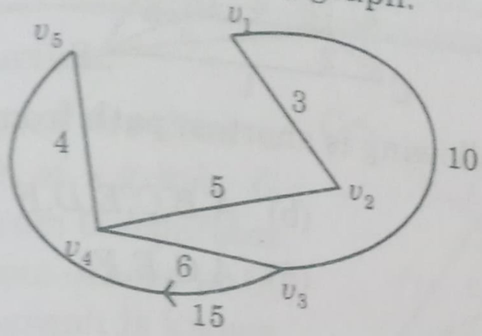


[Hints : A is starting vertex. Assign permanent label to each of the vertices. The amount of the label is the $S.D$ of the corresponding vertices from A.]

12. By Dijkstra's method find the shortest path from V to W in the following Graph :



13. By Dijkart's Algorithm find the shortest distance and shortest path from v_5 to v_3 for the following graph.

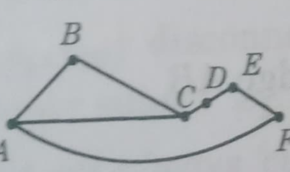


Answers

1. shortest path is v_1, v_2, v_3, v_4 ; shortest length $= 55$
2. shortest path : A, B, C, E, D, F ; $S \cdot D = 9$
3. shortest path : v_1, v_2, v_3, v_4 ; $S \cdot D = 5$
4. shortest path : a, c, d, f ; $S \cdot D = 6.8$
7. shortest path : a, c, b, d, e, f. S.D. = 13
9. $B \to C \to E \to G$: $S \cdot D = 7$.
10. there are three shortest paths :
    (1) S, A, E, T (2) S, A, B, T
    (3) S, F, E, T. $S.D$ = No of edges in a path = 3
12. V, A, D, E, B, G, H, W.
13. 10 ; $v_5 - v_4 - v_3$

[II] Multiple Choice Questions

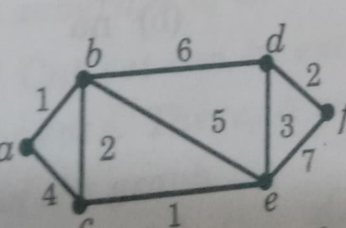
1. For the graph



the shortest path from A to E is

(a) $\{AB, BC, CD, DE\}$
(b) $\{AF, FE\}$
(c) $\{AC, CD, DE\}$
(d) none of these

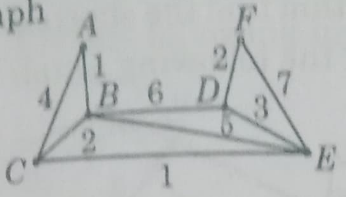
2. In the weighted graph



which one of the following is the shortast path from a to f

(a) $\{a, b, c, e, d, f\}$
(b) $\{a, b, d, f\}$
(c) $\{a, c, e, f\}$
(d) $\{a, b, e, f\}$

3. In the weighted graph



which one of the following is shortest path from A to F

(a) A,B,D,F (b) A,B,C,E,D,F

(c) A,B,E,F (d) A,C,E,F

4. Dijkstra's algorithm is used to

(a) find maximum flow in a net work

(b) to scan all vertices of a graph

(c) find the shortest path from a specified vertex to another

(d) none

5. Dijkstra's algorithm is applicable for digraph.

(a) yes (b) no

6. In Dijskrat's algorithm for a digraph if the edge AB is directed from A to B only then we take weight on the edge BA

(a) 0

(b) ∞

(c) – weight on the directed edge AB

(d) none

7. Dijskrat's algorithm can be applied for unweighted graph

(a) yes (b) no

Answers

1.b 2.a 3.b 4.c 5.a 6.b 7.a