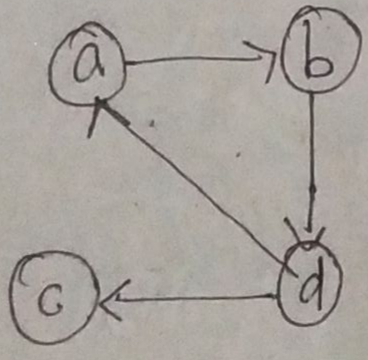
2) Warshalls Algorithm :

* Warshalls Algorithm is used to find the Transitive closure of a directed graph.

* The graph in which all the edges are directed are called digraph or directed graph.

1) digraph
2) Adjacent Matrix
3) Transitive closure.



→ Digraph.

Vertex U $\quad$ Vertex V.

1 → path
0 → No path.

$$R[0] = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 0 & 0 \\ b & 0 & 0 & 0 & 1 \\ c & 0 & 0 & 0 & 0 \\ d & 1 & 0 & 1 & 0 \end{array}$$

Intermediate

$\boxed{a}$

$$R[1] = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 0 & 0 \\ b & 0 & 0 & 0 & 1 \\ c & 0 & 0 & 0 & 0 \\ d & 1 & \boxed{1} & 1 & 0 \end{array}$$

Intermediate

$\boxed{a, b}$

$$R[2] = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 0 & \boxed{1} \\ b & 0 & 0 & 0 & 1 \\ c & 0 & 0 & 0 & 0 \\ d & 1 & 1 & 1 & \boxed{1} \end{array}$$

Intermediate

$\boxed{a, b, c}$

$$R[3] = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Intermediate

$\boxed{a, b, c, d.}$

$$R[4] = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} \boxed{1} & 1 & \boxed{1} & 1 \\ \boxed{1} & \boxed{1} & \boxed{1} & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Vertex u Vertex V.

Transitive closure.

solution :-
$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$
using warshalls Algorithm.

Algorithm :

Algorithm Warshall (Matrix [1...n, 1..n])

// Problem Description : This algorithm is for computing Transitive closure using warshall's Algorithm..

// Input : The adjacency matrix given by Matrix [1...n, 1..n].

// output : the Transitive closure of digraph.

```
R(0) ← Matrix.
for (k ← 1 to n) do.
{
    for (i ← 1 to n) do.
    { for (j ← 1 to n) do.
        { R(k) [i, j] ← R^(k-1) [i, j] AND
                        R(k-1) [k, j].
    } }
}
return R(n).
```

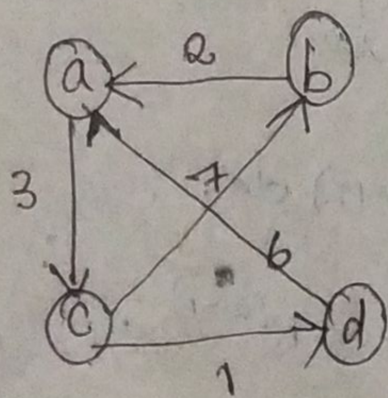
Floyds Algorithm:

> given a weighted connected graph that all pairs shortest path problem finds the distances v length of the shortest path from each vertex to all other vertices.

> weighted graph are represented using weight Matrix.

> Floyds Algorithm computes distance D of the matrix.

Eg:



$$W = D[0] = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \infty & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

$I \to a.$

$$D[1] = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \boxed{5} & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

$I \to a, b.$

$$D[2] = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ \boxed{9} & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{bmatrix} \end{array}$$

$I \to a, b, c.$

$$D[3] = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \boxed{10} & 3 & \boxed{4} \\ 2 & 0 & 5 & 6 \\ 9 & 7 & 0 & 1 \\ 6 & \boxed{16} & 9 & 0 \end{bmatrix} \end{array}$$

$$D[0] = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \infty & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

$$D[1] = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \boxed{5} & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \boxed{9} & 0 \end{bmatrix} \end{array}$$

$(b, c) = \min(5, \infty) = 5$

$(b, d) = \min(\infty, \infty) = \infty$

$(c, b) = \min(\infty, 7) = 7.$

$(c, d) = \min(1, \infty) = 1.$

$(d, b) = \min(\infty, \infty) = \infty$

$(d, c) = \min(\infty, 9) = 9.$

$$D(2) = \begin{array}{c} \\ a \\ b \\ c \\ d \end{array} \begin{array}{c} \begin{array}{cccc} a & b & c & d \end{array} \\ \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ \boxed{9} & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{bmatrix} \end{array}$$

$I \to a, b, c, d.$

$$D[4] = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 10 & 3 & 4 \\ b & 2 & 0 & 5 & 6 \\ c & \boxed{7} & 7 & 0 & 1 \\ d & 6 & 16 & 9 & 0 \end{array}$$

Solution.

All pair shortest path.

$$D = \begin{bmatrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ 7 & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{bmatrix}$$

$$D[3] = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & \boxed{10} & 3 & \boxed{4} \\ b & 2 & 0 & 5 & \boxed{6} \\ c & 9 & 7 & 0 & 1 \\ d & 6 & \boxed{16} & 9 & 0 \end{array}$$

$$D[4] = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 10 & 3 & 4 \\ b & 2 & 0 & 5 & 6 \\ c & \boxed{7} & 7 & 0 & 1 \\ d & 6 & 16 & 9 & 0 \end{array}$$

$\Downarrow$

Solution = ??

Algorithm :

Algorithm Floyd_shortest_path (wt[1...n, 1...n])

// Pr. res. : this algorithm is for computing. shortest path between all pair of vertices. .

// I/P : the weighted matrix, wt[1...n, 1...n] For. given graph.

// O/P : the distance matrix D containing shortest. paths .

D ← wt.
for k ← 1 to n do.
{ for i ← 1 to n do
{ for j ← 1 to n do.
{
D(i,j) ← min { D(i,j), D(i,k)+D(k,j)) }.
}
} }

return D.