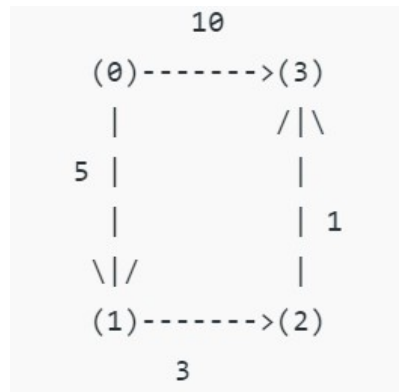


Floyd Warshall Algorithm

- Floyd Warshall Algorithm is for solving the All-Pairs Shortest Path problem.
- The problem is to find shortest distances between every pair of vertices in a given weighted directed Graph.

Example Graph:



Input: adjacency matrix representation of above graph

```
graph[][] = { {0, 5, INF, 10},
               {INF, 0, 3, INF},
               {INF, INF, 0, 1},
               {INF, INF, INF, 0} }
```

Note:

- The value of $\text{graph}[i][j]$ is 0 if i is equal to j .
- The value of $\text{graph}[i][j]$ is INF (infinite) if there is no edge from vertex i to j .

Output:

Shortest distance matrix				
0	5	8	9	
INF	0	3	4	
INF	INF	0	1	
INF	INF	INF	0	

Example:

Floyd warshall Algorithm ①

Input :

	0	1	2	3
0	0	5	∞	10
1	∞	0	3	∞
2	∞	∞	0	1
3	∞	∞	∞	0

O/P :

	0	1	2	3
0	0	5	8	9
1	∞	0	3	4
2	∞	∞	0	1
3	∞	∞	∞	0

3 nested loops $\Rightarrow$ values from 0 to no. of vertices (V)-1

$\therefore k = 0$ to $3 \rightarrow$ intermediate vertex

$i = 0$ to $3 \rightarrow$ source vertex

$j = 0$ to $3 \rightarrow$ destination vertex.

Condition to check: $\text{dist}[i,k] \neq \infty$ and $\text{dist}[k,j] \neq \infty$

if condition is true, then

$$\text{dist}[i,j] = \min \{ \text{dist}[i,j], \text{dist}[i,k] + \text{dist}[k,j] \}$$

Iteration-1 :

$k=0$ $i=0$ $j=0$ $\text{dist}[0,0] \neq \infty \checkmark$ $\text{dist}[0,0] = \min(0, 0+0) = 0$
 $i=0$ $j=1$ $\text{dist}[0,1] \neq \infty \checkmark$ $\text{dist}[0,1] = \min(5, 0+5) = 5$
 $i=0$ $j=2$ $\text{dist}[0,2] \neq \infty \times$
 $i=0$ $j=3$ $\text{dist}[0,3] \neq \infty \checkmark$ $\text{dist}[0,3] = \min(10, 0+10) = 10$

$i=1$ $j=0$ $\text{dist}[1,0] \neq \infty \times$

$i=2$ $j=0$ $\text{dist}[2,0] \neq \infty \times$

$i=3$ $j=0$ $\text{dist}[3,0] \neq \infty \times$

	0	1	2	3
0	0	5	∞	10
1	∞	0	3	∞
2	∞	∞	0	1
3	∞	∞	∞	0

Iteration - 2 :

(2)

$$k=1 \quad i=0 \quad j=0 \quad \text{dist}[1,0] = \infty \times$$

$$\text{dist}[0,1] = \infty \quad 1 \quad \text{dist}[1,1] = \infty \quad \text{dist}[0,1] = \min(5, 5+0) = 5$$

$$2 \quad \text{dist}[1,2] = \infty \quad \text{dist}[0,2] = \min(\infty, 5+3) = 8$$

$$3 \quad \text{dist}[1,3] = \infty \times$$

$$k=1 \quad i=1 \quad j=0 \times$$

$$1 \quad \text{dist}[1,1] = \min(0, 0+0) = 0$$

$$\text{dist}[1,1] = \infty \quad 2 \quad \text{dist}[1,2] = \min(3, 0+3) = 3$$

$$3 \times$$

$$i=2 \times \quad j=0 \quad \text{dist}[2,1] = \infty$$

$$1 \times$$

$$2 \times$$

$$3 \times$$

$$i=3 \times \quad j=0 \quad \text{dist}[3,1] = \infty$$

$$1 \times$$

$$2 \times$$

$$3 \times$$

	0	1	2	3
0	0	5	8	10
1	∞	0	3	∞
2	∞	∞	0	1
3	∞	∞	∞	0

Iteration - 3 :

$$k=2 \quad i=0 \quad j=0 \quad \text{dist}[2,0] = \infty \times$$

$$\text{dist}[0,2] = \infty \quad 1 \quad \text{dist}[2,1] = \infty \times$$

$$2 \quad \text{dist}[2,2] = \infty \quad \text{dist}[0,2] = \min(8, 8+0) = 8$$

$$3 \quad \text{dist}[2,3] = \infty \quad \text{dist}[0,3] = \min(10, 8+1) = 9$$

$$i=1 \quad j=0 \times$$

$$1 \times$$

$$\text{dist}[1,2] = \infty \quad 2 \quad \text{dist}[1,2] = \min(3, 3+0) = 3$$

$$3 \quad \text{dist}[1,3] = \min(\infty, 3+1) = 4$$

$$i=2 \quad j=0 \times$$

$$\text{dist}[2,2] = \infty \quad 1 \times$$

$$2 \quad \text{dist}[2,2] = \min(0, 0+0) = 0$$

$$3 \quad \text{dist}[2,3] = \min(1, 0+1) = 1$$

$$i=3 \quad j=0 \quad \text{dist}[3,2] = \infty$$

$$1 \times$$

$$2 \times$$

$$3 \times$$

	0	1	2	3
0	0	5	8	9
1	∞	0	3	4
2	∞	∞	0	1
3	∞	∞	∞	0

(3)

Iteration -4:

$$k=3 \quad i=0 \vee j=0 \quad \text{dist}[3,0] = \infty$$

$$1 \quad \text{dist}[3,1] = \infty$$

$$\text{dist}[0,3] = \infty \quad 2 \quad \text{dist}[3,2] = \infty$$

$$3 \quad \text{dist}[3,3] = \infty \quad \text{dist}[0,3] = \min(9, 9+0) = 9$$

$$i=1 \vee j=0$$

$$1 \quad \times$$

$$\text{dist}[1,3] = \infty \quad 2 \quad \times$$

$$3 \vee \text{dist}[1,3] = \min(4, 4+0) = 4$$

$$i=2 \vee j=0$$

$$1 \quad \times$$

$$\text{dist}[2,3] = \infty \quad 2 \quad \times$$

$$3 \vee \text{dist}[2,3] = \min(1, 1+0) = 1$$

$$i=3 \vee j=0$$

$$1 \quad \times$$

$$\text{dist}[3,3] = \infty \quad 2 \quad \times$$

$$3 \vee \text{dist}[3,3] = \min(0, 0+0) = 0$$

	0	1	2	3
0	0	5	8	9
1	∞	0	3	4
2	∞	∞	0	1
3	∞	∞	∞	0

Algorithm:

Floyd_Warshall(graph[][])

for (k = 0; k < V; k++) // k is intermediate vertex. Pick all vertices as intermediate vertices one by one

 for (i = 0; i < V; i++) // Pick all vertices as source one by one

 for (j = 0; j < V; j++) // Pick all vertices as destination for the above picked source

 If (dist[i][k] != INF && dist[k][j] != INF)

 if (dist[i][j] > (dist[i][k] + dist[k][j]))

 dist[i][j] = dist[i][k] + dist[k][j];

Time Complexity: $O(V^3)$