UNIT-3

ALL PAIR SHORTEST PATH

- Let G=<N,A> be a directed graph 'N' is a set of nodes and 'A' is the set of edges.
- Each edge has an associated non-negative length.
- We want to calculate the length of the shortest path between each pair of nodes.
- Suppose the nodes of G are numbered from 1 to n, so N={1,2,...N},and suppose G matrix L gives the length of each edge, with L(i,j)=0 for i=1,2...n,L(i,j)>=for all i & j, and L(i,j)=infinity, if the edge (i,j) does not exist.
- The principle of optimality applies: if k is the node on the shortest path from i to j then the part of the path from i to k and the part from k to j must also be optimal, that is shorter.
- First, create a cost adjacency matrix for the given graph.
- Copy the above matrix-to-matrix D, which will give the direct distance between nodes.
- We have to perform N iteration after iteration k.the matrix D will give you the distance between nodes with only (1,2...,k)as intermediate nodes.
- At the iteration k, we have to check for each pair of nodes (i,j) whether or not there exists a path from i to j passing through node k.

COST ADJACENCY MATRIX:

$$D0 = L = \begin{vmatrix} 0 & 5 & \propto & \propto \\ 50 & 0 & 15 & 5 \\ 30 & \propto & 0 & 15 \\ 15 & \propto & 5 & 0 \end{vmatrix}$$

$$\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{vmatrix} 7 & 5 & \propto & \propto \\ 7 & \propto & \propto & 2 \\ \propto & 3 & \propto & \propto \\ 4 & \propto & 1 & \propto \end{vmatrix} \quad \begin{vmatrix} 11 & 12 & - & - \\ 21 & - & - & 24 \\ - & 32 & - & - \\ 41 & - & 43 & - \end{vmatrix}$$

vertex 1:

$$\begin{vmatrix} 7 & 5 & \propto & \propto \\ 7 & \mathbf{12} & \propto & 2 \\ \propto & 3 & \propto & \propto \\ 4 & \mathbf{9} & 1 & \propto \end{vmatrix} \begin{vmatrix} 11 & 12 & - & - \\ 21 & \mathbf{212} & - & 24 \\ - & 32 & - & - \\ 41 & \mathbf{412} & 43 & - \end{vmatrix}$$

vertex 2:

$$\begin{vmatrix} 7 & 5 & \propto & \mathbf{7} \\ 7 & 12 & \propto & 2 \\ \mathbf{10} & 3 & \propto & \mathbf{5} \\ 4 & 9 & 1 & \mathbf{11} \end{vmatrix} \begin{vmatrix} 11 & 12 & - & \mathbf{124} \\ 21 & 212 & - & 24 \\ \mathbf{321} & 32 & - & \mathbf{324} \\ 41 & 412 & 43 & \mathbf{4124} \end{vmatrix}$$

vertex 3:

$$\begin{vmatrix} 7 & 5 & \propto & 7 \\ 7 & 12 & \propto & 2 \\ 10 & 3 & \propto & 5 \\ 4 & \mathbf{4} & 1 & \mathbf{6} \end{vmatrix} \begin{vmatrix} 11 & 12 & - & 124 \\ 21 & 212 & - & 24 \\ 321 & 32 & - & 324 \\ 41 & \mathbf{432} & 43 & \mathbf{4324} \end{vmatrix}$$

vertex 4:

$$\begin{vmatrix} 7 & 5 & \mathbf{8} & 7 \\ \mathbf{6} & \mathbf{6} & \mathbf{3} & 2 \\ \mathbf{9} & 3 & \mathbf{6} & 5 \\ 4 & 4 & 1 & 6 \end{vmatrix} \begin{vmatrix} 11 & 12 & \mathbf{1243} & 124 \\ \mathbf{241} & \mathbf{2432} & \mathbf{243} & 24 \\ \mathbf{3241} & 32 & \mathbf{3243} & 324 \\ 41 & 432 & 43 & 4324 \end{vmatrix}$$

- At 0^{th} iteration it nil give you the direct distances between any 2 nodes

$$D0=\begin{vmatrix} 0 & 5 & \propto & \propto \\ 50 & 0 & 15 & 5 \\ 30 & \propto & 0 & 15 \\ 15 & \propto & 5 & 0 \end{vmatrix}$$

- At 1^{st} iteration we have to check the each pair(i,j) whether there is a path through node 1.if so we have to check whether it is minimum than the previous value and if I is so than the distance through 1 is the value of d1(i,j).at the same time we have to solve the intermediate node in the matrix position p(i,j).

$$D1=\begin{vmatrix} 0 & 5 & \propto & \propto \\ 50 & 0 & 15 & 5 \\ 30 & \mathbf{35} & 0 & 15 \\ & & & \end{vmatrix} \qquad \begin{matrix} p[3,2]= 1 \\ p[4,2]= 1 \end{matrix}$$

15 **20** 5 0

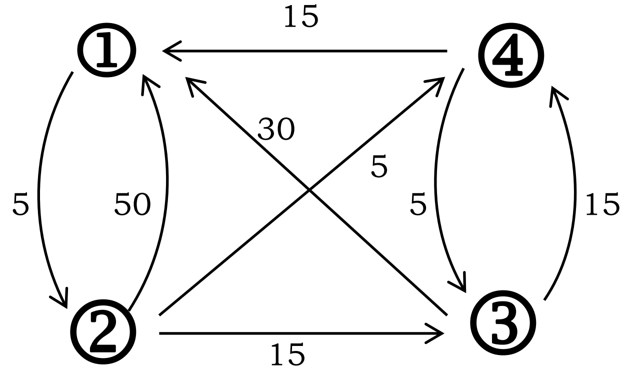

Fig: floyd's algorithm and work

❖ likewise we have to find the value for N iteration (ie) for N nodes.

$$D2=\left|\begin{array}{cccc} 0 & 5 & \mathbf{20} & \mathbf{10} \\ 50 & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{array}\right| \quad \begin{array}{l} P[1,3] = 2 \\ P[1,4] = 2 \end{array}$$

$$D3=\left|\begin{array}{cccc} 0 & 5 & 20 & 10 \\ \mathbf{45} & 0 & 15 & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{array}\right| \quad P[2,1]=3$$

$$D4=\left|\begin{array}{cccc} 0 & 5 & \mathbf{15} & 10 \\ 20 & 0 & \mathbf{10} & 5 \\ 30 & 35 & 0 & 15 \\ 15 & 20 & 5 & 0 \end{array}\right| \quad \begin{array}{l} P[1,3]=4 \\ P[2,3]=4 \end{array}$$

❖ D4 will give the shortest distance between any pair of nodes.

❖ If you want the exact path then we have to refer the matrix p.The matrix will be,

$$\left|\begin{array}{cccc} 0 & 0 & 4 & 2 \\ 3 & 0 & 4 & 0 \end{array}\right| \quad 0 \longrightarrow \text{direct path}$$

P= 0 1 0 0
0 1 0 0

- Since,p[1,3]=4,the shortest path from 1 to3 passes through 4.
- Looking now at p[1,4]&p[4,3] we discover that between 1 & 4, we have to go to node 2 but that from 4 to 3 we proceed directly.
- Finally we see the trips from 1 to 2, & from 2 to 4, are also direct.
- The shortest path from 1 to 3 is 1,2,4,3.

ALGORITHM :

Function Floyd (L[1..r,1..r]):array[1..n,1..n]
array D[1..n,1..n]

D = L
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

ANALYSIS:

This algorithm takes a time of θ (n^3)

Additional Resources which are useful to refer:

https://www.youtube.com/watch?v=4OQeCuLYj-4

https://www.youtube.com/watch?v=4NQ3HnhyNfQ
https://www.youtube.com/watch?v=nV_wOZnhbog [important]
https://www.youtube.com/watch?v=DzfmJoFq1pc
https://www.youtube.com/watch?v=oNI0rf2P9gE