Warshall's algorithm

→ used to compute transitive closure of directed graph.

$R^{(0)}$ $R^{(1)}$ $R^{(2)}$ .... $R^{(K-1)}$ R^{K}

↓

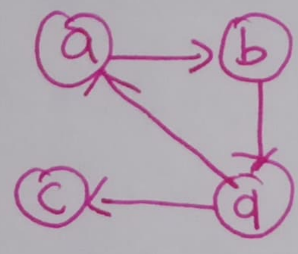
matrices of order 'n'.

Vertices are numbered from 1 to n.

i) digraph

ii) adjacency matrix

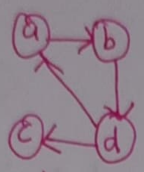
iii) Transitive closure.



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

No intermediate

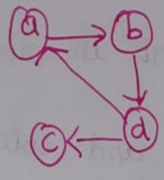
To find R(1).



$$R(1) = \begin{array}{cc} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & \boxed{1} & 1 & 0 \end{bmatrix} \end{array}$$

with intermediate 'a'

To find R_2



$$R[2] = \begin{array}{cc} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & \boxed{1} \end{bmatrix} \end{array}$$

with intermediate a and b

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

with intermediate a, b and c.

$$R(4) = \begin{array}{cc} & \begin{matrix} a & b & c & d \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{bmatrix} \boxed{1} & 1 & \boxed{1} & 1 \\ \boxed{1} & \boxed{1} & \boxed{1} & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

with intermediate a, b, c, d.

Warshalls algorithm:

Computation of warshall's algorithm is given by

$R^{(0)}, R^{(1)}, R^{(2)} \ldots R^{(k-1)}, R^{(k)}$.

$R^{(0)}$ - Intermediate vertices are not allowed. hence it is adjacency matrix.

$R^{(1)}$ calculated using $R^{(0)}$

$$r_{ij}^{K} = r_{ij}^{K-1} \text{ or } r_{iK}^{K-1} \text{ and } r_{Kj}^{K-1}$$

Conditions for solving warshall algorithm:

1 -> If an element r_{ij} is 1 in r_{ij}^{K-1}, it remains in '1'

2 -> If an element r_{ij} is 0 in r^{K-1} it has to changed to '1' in r^{K} iff the element in row 'i' and column 'k' and element in its row 'k' and column 'j'. are both 1 in r^{K}.

Algo:

```
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) or R^(k-1)(i,k) AND R^(k-1)(k,j)
    }
   }
  }.
return R(n).
```

Time complexity $\Theta(n^3)$