# SNS COLLEGE OF ENGINEERING

# Coimbatore-107

## COURSE NAME: ANALYSIS OF ALGORITHM

## II YEAR/ IV SEMESTER

## UNIT – III

## DYNAMIC PROGRAMMING

**Topic**

**Dynamic Programming: Warshall's Algorithm**

Date
Page

## Floyd Algorithm:

*. It finds shortest path between every pair of vertices of graph

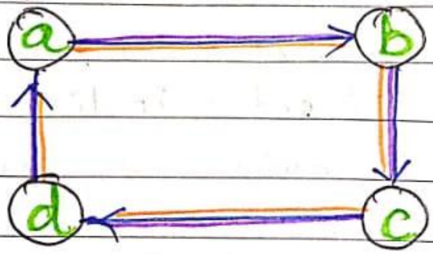
*. No negative cycles is allowed

*. It needs weighted graph

## Warshall's Algorithm

### Digraph:

The graph in which all the edges are directed ~~gra~~ then it is called digraph/Directed graph.



### Adjacency Matrix:

It is a representation of a graph by using matrix. If there exists an direct edge from vertices $V_i$ to $V_j$; entry can be represented as 1.

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

a→b : 1
b→c : 1
c→d : 1
d→a : 1

## Transitive closure:

$$\begin{array}{c|cccc} & a & b & c & d \\ \hline a & 1 & 1 & 1 & 1 \\ b & 1 & 1 & 1 & 1 \\ c & 1 & 1 & 1 & 1 \\ d & 1 & 1 & 1 & 1 \end{array}$$

From examp
Since all v
Can reach all
vertices throu
direeit/ indirec
So considered '1

⇒ It is a Boolean matrix (Matr with 0 & 1 values). It uses Depth First Search (DFS) or (BFS) Breadth First search to traverse from any vertex.

*⇒ If there exist any path from one vertex to another vertex [Direct / indirect path] then it should be represented a '1' in matric.

## Warshall's Algorithm:

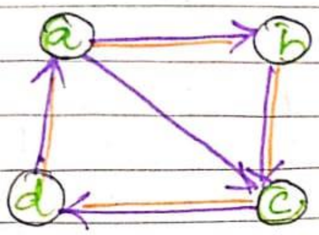
Warshall Algorithm constructs transitive closure of given digrap with 'n' vertices through a Series of n-by-n Boolean vertic by avoiding repeated traversing.

Computation:

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

(Building of Boolean matrices).

Example:



Step 1: Construct an Adjacency Matrix `$R_0$´

$$R_0 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 1 & 0 \\ b & 0 & 0 & 1 & 0 \\ c & 0 & 0 & 0 & 1 \\ d & 1 & 0 & 0 & 0 \end{array}$$

Self loop
a -> a
b -> b
c -> c
d -> d
} -> `0´

1) Self loop is denoted as `0´ in matrix
2) Direct edges is represented as `1´
(3) Indirect edges is denoted by `0`.

—— × —— × —— × —— × —— × ——

Step 2: construct matrix `$R_1$´ with `a´ as intermediate node.

$$R_1 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 1 & 0 \\ b & 0 & 0 & 1 & 0 \\ c & 0 & 0 & 0 & 1 \\ d & 1 & 1 & 1 & 0 \end{array}$$

self loop
a -> a
b -> b
c -> c
d -> d

(i) Self loop is represented with `0´
(ii) Direct edges given as `1´
(iii) Indirect edges with `a´ as intermediat only is given as `1´.

Step 3 : construct 'R2' Matrix with a 'b' as intermediate node.

$$R_2 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 1 & 0 \\ b & 0 & 0 & 1 & 0 \\ c & 0 & 0 & 0 & 1 \\ d & 1 & 1 & 1 & 0 \end{array}$$

*. Self loop is denoted by '0'.

*. Direct edges are given '1' in

*. Indirect edges with only 'a' &

as intermediate is given '1'.

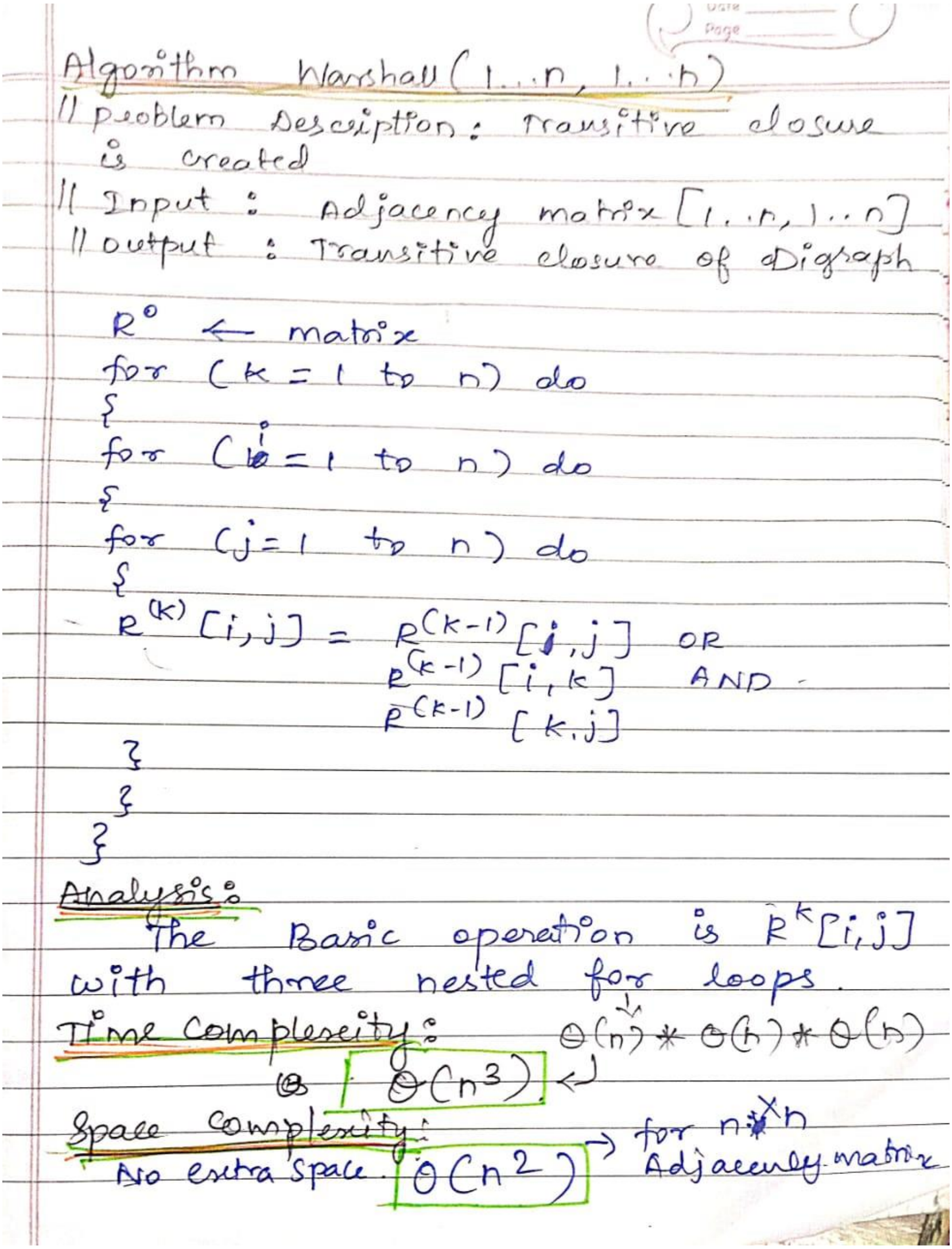
Step 4 : construct 'R3' Matrix with a, 'c' as in nod

$$R_3 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 1 & 1 \\ b & 0 & 0 & 1 & 1 \\ c & 0 & 0 & 0 & 1 \\ d & a & 1 & 1 & \cancel{0}1 \end{array}$$

Step 5 : construct with a, b, c, d a intermi

$$R_4 = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 1 & 1 & 1 & 1 \\ b & 1 & 1 & 1 & 1 \\ c & 1 & 1 & 1 & 1 \\ d & 1 & 1 & 1 & 1 \end{array}$$

Every node is reachable from every other node.



Algorithm Warshall (1..n, 1..n)

// problem Description : Transitive closure is created

// Input : Adjacency matrix [1..n, 1..n]

// output : 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
{
$R^{(k)}[i,j] = R^{(k-1)}[i,j]$ OR
                $R^{(k-1)}[i,k]$ AND
                $R^{(k-1)}[k,j]$
}
}
}
```

Analysis:

The Basic operation is $R^k[i,j]$ with three nested for loops.

Time Complexity: $\Theta(n) * \Theta(n) * \Theta(n)$

$= \boxed{\Theta(n^3)}$

Space Complexity:

No extra space. $\boxed{O(n^2)}$ → for $n \times n$ Adjacency matrix