# UNIVERSITY INSTITUTE OF COMPUTING

## MASTER OF COMPUTER APPLICATIONS

## Design and Analysis of Algorithms

24CAT-611

UNIT-3

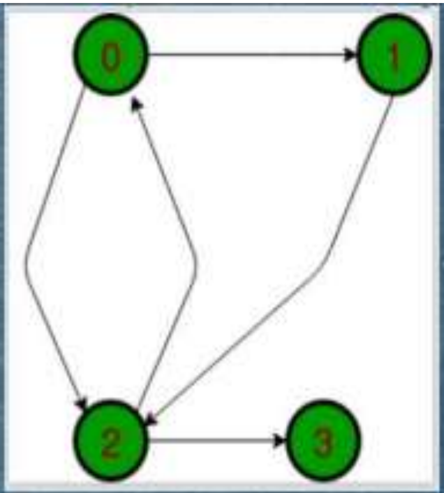
DISCOVER . LEARN . EMPOWER

# Transitive Closure

- A vertex j is reachable from another matrix i for all matrix pairs(i, j) in the given graph.
- Reachable (there is a path from vertex i to j).
- Example:

  Set, A= {0,1,2,3}

  Relation, R = {(1,2),(2,3),(3,4)}

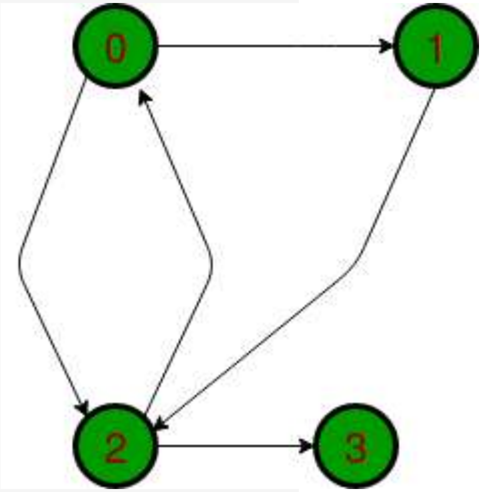



(1,3) is the transitive closure which must belongs to R.

- Obtained by repeatedly adding the new value using union operation with previous Relation.

# Transitive Closure

- Given a directed graph, find out if a vertex j is reachable from another vertex i for all vertex pairs (i, j) in the given graph. Here reachable mean that there is a path from vertex i to j. The reach-ability matrix is called the transitive closure of a graph




```
Transitive closure of above graphs is
     1 1 1 1
     1 1 1 1
     1 1 1 1
     0 0 0 1
```

# Transitive Closure

- The graph is given in the form of adjacency matrix say 'graph[V][V]' where graph[i][j] is 1 if there is an edge from vertex i to vertex j or i is equal to j, otherwise graph[i][j] is 0.
Floyd Warshall Algorithm can be used, we can calculate the distance matrix dist[V][V] using Floyd Warshall, if dist[i][j] is infinite, then j is not reachable from i. Otherwise, j is reachable and the value of dist[i][j] will be less than V.

# Warshall Algorithm

- Main Idea:

Suppose two nodes 1 and 4

a path exists between two vertices 1, 4, if

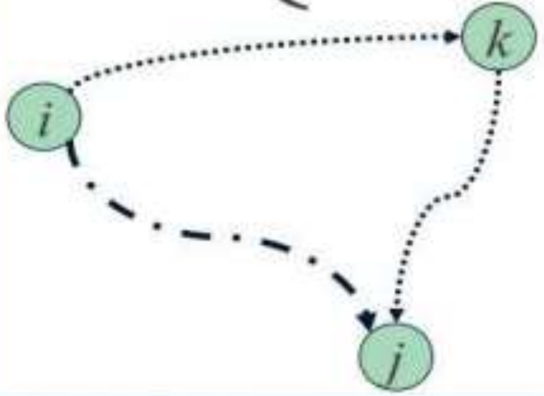
- there is an edge from 1 to 4; or
- there is a path from 1 to 4 going through vertex 2; or
- there is a path from 1 to 4 going through vertex 2 and/or 3; or
- ...

So, (1,4) is a transitive closure

# Warshall Algorithm

- On the kth iteration the algorithm determine if a path exists between two vertices i,j using just vertices among 1,….,k allowed as intermediate

$$R^{(k)}[i,j] = \begin{cases} R^{(k-1)}[i,j] & \text{path using just } 1,\ldots,k\text{-}1 \\ \text{or} & \\ (R^{(k-1)}[i,k] \ \text{ and } \ R^{(k-1)}[k,j]) & \text{path from i to k and from k to j using just } 1,\ldots,k\text{-}1 \end{cases}$$




$k^{th}$ iteration

# Warshall Algorithm : Transitive Closure

## ❖ Algorithm to find Transitive Closure

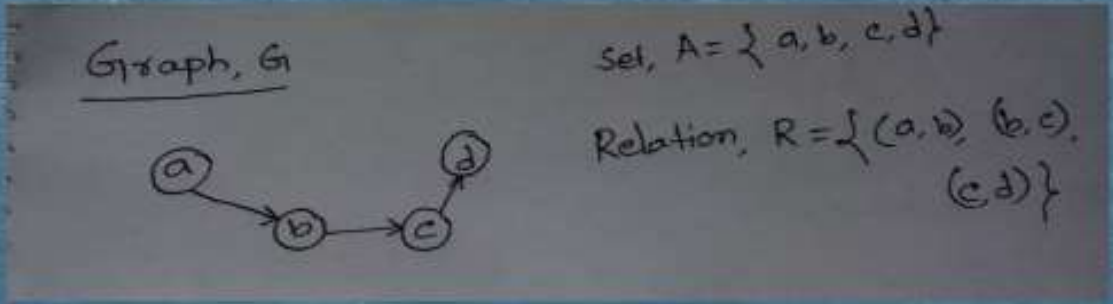
Input: The given graph.

Output: Transitive Closure matrix.

```
TransitiveClosure {
        R(0) ⟵ A        //copy adjacency matrix into another 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]);
}
```

# Working Procedure

- First consider a Graph(G), a Set(A, contains 4 elements) and a Relation Set(R).




- It's respective Adjacency Matrix ($M_R$) is:

Adjacency Matrix, $M_R$

$$
\begin{array}{c c}
 & \begin{array}{cccc} a & b & c & d \end{array} \\
\begin{array}{c} a \\ b \\ c \\ d \end{array} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{4 \times 4}
\end{array}
$$

# Working Procedure

- Fetching Transitive Closure

from 1st iteration Adjacency Matrix ($M_R$).

Transitive closure

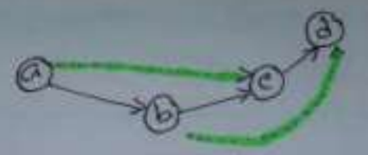
| | I | II | III | IV |
|---|---|---|---|---|
| Column | $\phi$ | {a} | {b} | {c} |
| Row | {b} | {c} | {d} | $\phi$ |
| Cartesian Product | × | (a,c) | (b,d) | × |

- There is found two new element of (a,c) and (b,d).
- Perform union operation between two new element and previous Relation.
- The graph has got two new edge.

Relation $R = \{(a,b), (b,c), (c,d)\} \cup \{(a,c), (b,d)\}$

$R = \{(a,b), (b,c), (c,d), (a,c), (b,d)\}$

Graph G:



# Working Procedure

- Adjacency Matrix($M_R^2$) by
  updating previous $M_R$(2nd iteration).
- there have paths of (a and c, b and d),
  that's why value is updated from 0 to 1.

- Lets see through Warshall Algorithm
- There is a edge between a and c


Adjacency Matrix $M_R^2$

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

$$\begin{aligned} M_R^2[a,c] &= M_R[a,c] \text{ OR } (M_R[a,b] \text{ AND } M_R[b,c]) \\ &= 0 \text{ OR } (1 \text{ AND } 1) \\ &= 0 \text{ OR } 1 \end{aligned}$$

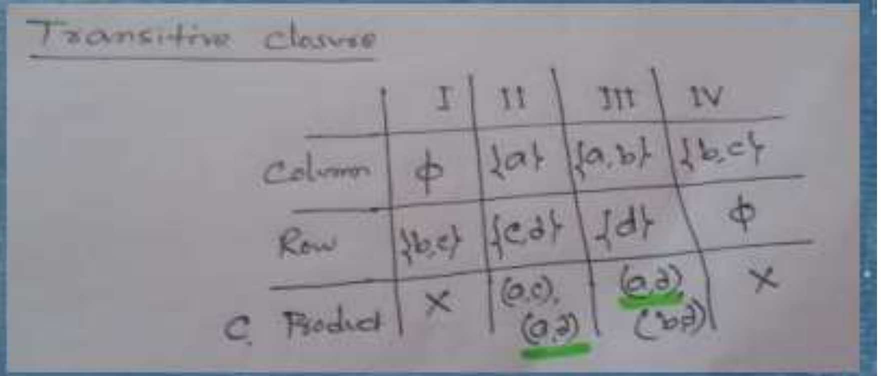
$$M_R^2[a,c] = 1$$

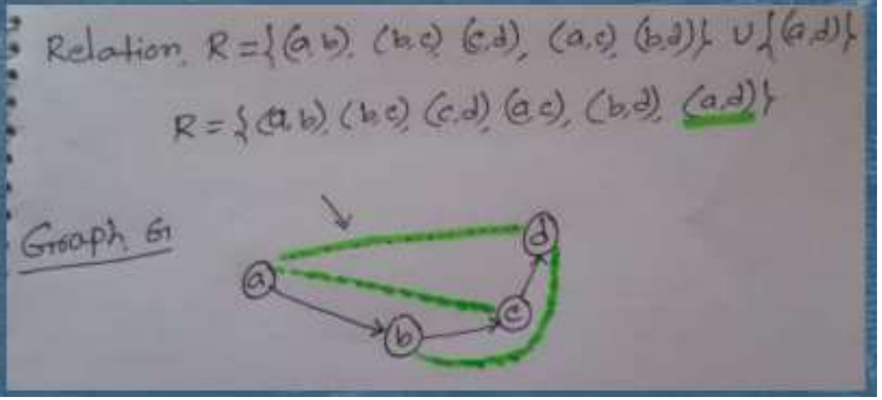
# Working Procedure

- Fetching Transitive Closure

  from Adjacency Matrix ($M_R^2$).

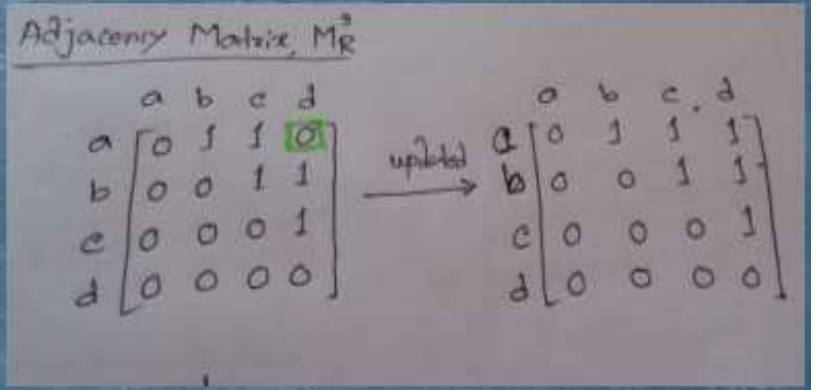
- Found a new element of (a,d)
- Perform again union operation.
- new edge of a to d



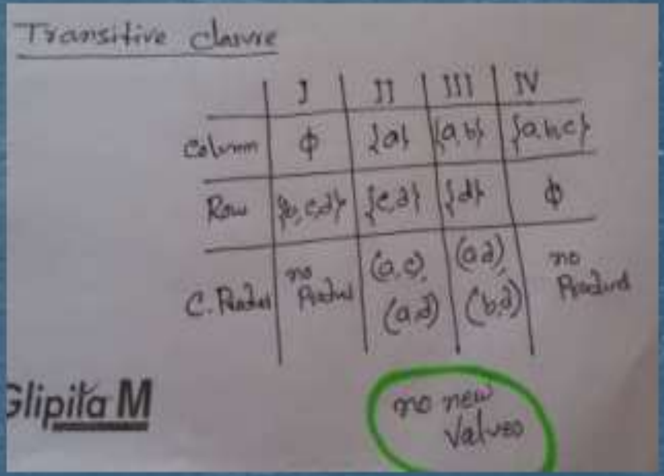



# Working Procedure

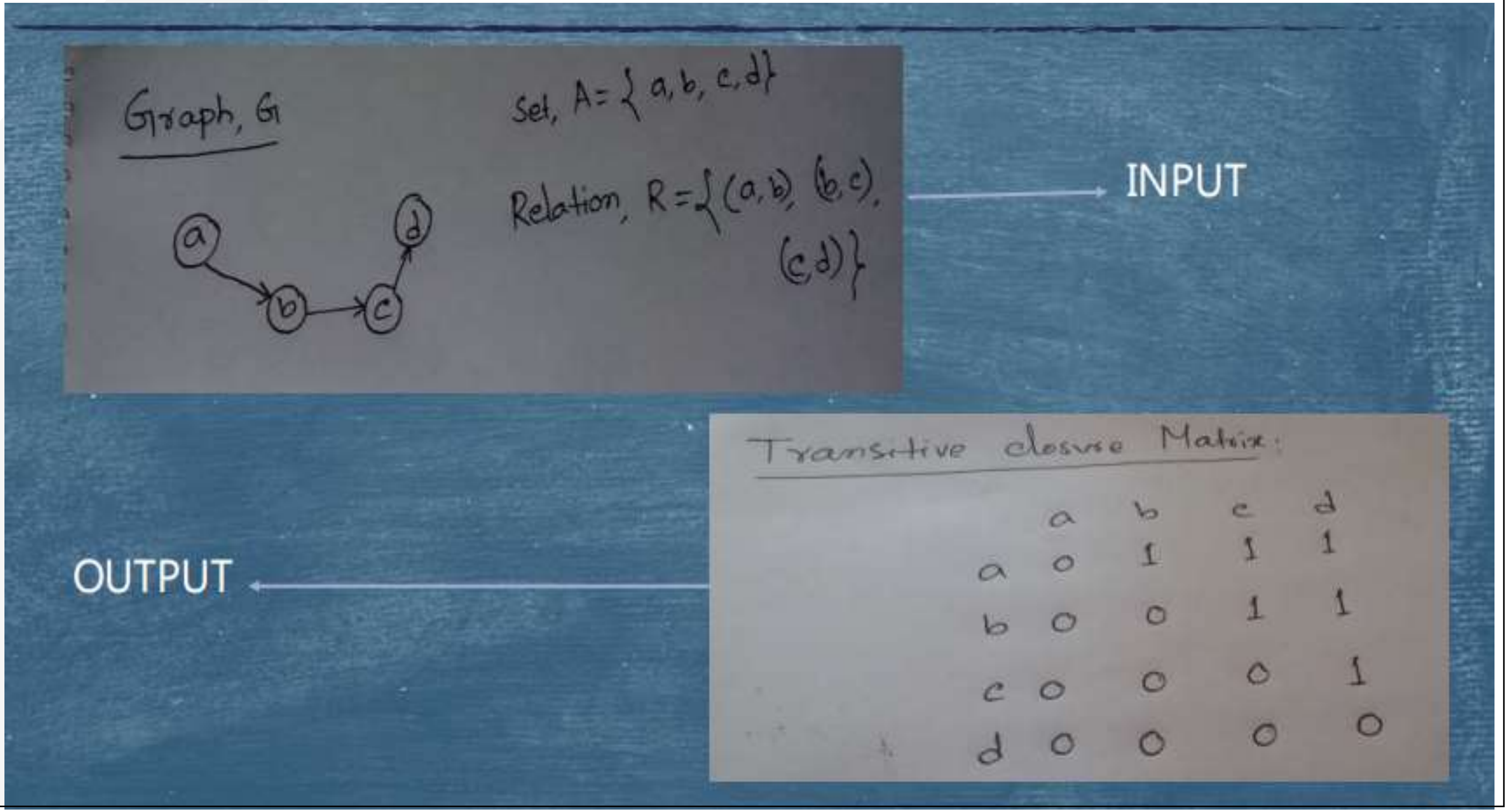
- Update matrix in a repeating way.




- Transitive closure from Adjacency Matrix ($M_R^3$)
- No new valus are found
- So, Adjacency Matrix ($M_R^3$) is the Transitive Closure Matrix.
- Going forward cause of new value.



# Result

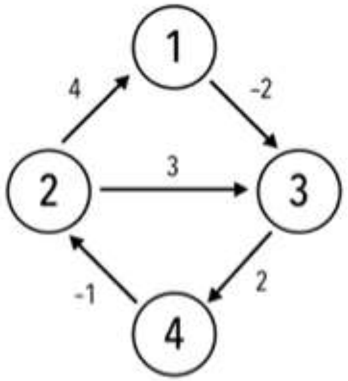

Graph, G
Set, A = {a, b, c, d}
Relation, R = {(a,b), (b,c), (c,d)}
a
b
c
d
INPUT
Transitive closure Matrix:
a b c d
a 0 1 1 1
b 0 0 1 1
c 0 0 0 1
d 0 0 0 0
OUTPUT

# Complexity

- Time Complexity: $\Theta(n^3)$ as there are 3 nested loops(i,j,k).
- Space Complexity: Requires extra space for separate matrices for recording intermediate results of the algorithm.

# Pseudocode Warshall's Algorithm

```
let V = number of vertices in graph
let dist = V × V array of minimum distances
for each vertex v
  dist [v][v] ← 0
for each edge (u,v)
  dist [u][v] ← weight(u,v)
for k from 1 to V
  for i from 1 to V
    for j from 1 to V
      if dist [i][j] > dist [i][k] + dist [k][j]
        dist [i][j] ← dist [i][k] + dist [k][j]
      end if
```




| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0 | | -2 | |
| 2 | 4 | 0 | 3 | |
| 3 | | | 0 | 2 |
| 4 | | -1 | | 0 |

# References


1) https://www.tutorialspoint.com/data_structures_algorithms/divide_and_conquer.htm

2) Data Structures and Algorithms made easy By Narasimha Karumanchi.

3) The Algorithm Design Manual, 2nd Edition by Steven S Skiena

4) Fundamentals of Computer Algorithms - Horowitz and Sahani

# THANK YOU