CSE 2103
(Data Structures)

Lecture on
Chapter-9: Graph

By

Dr. M. Golam Rashed

(golamrashed@ru.ac.bd)

VARENDRA UNIVERSITY

বরেন্দ্র বিশ্ববিদ্যালয়

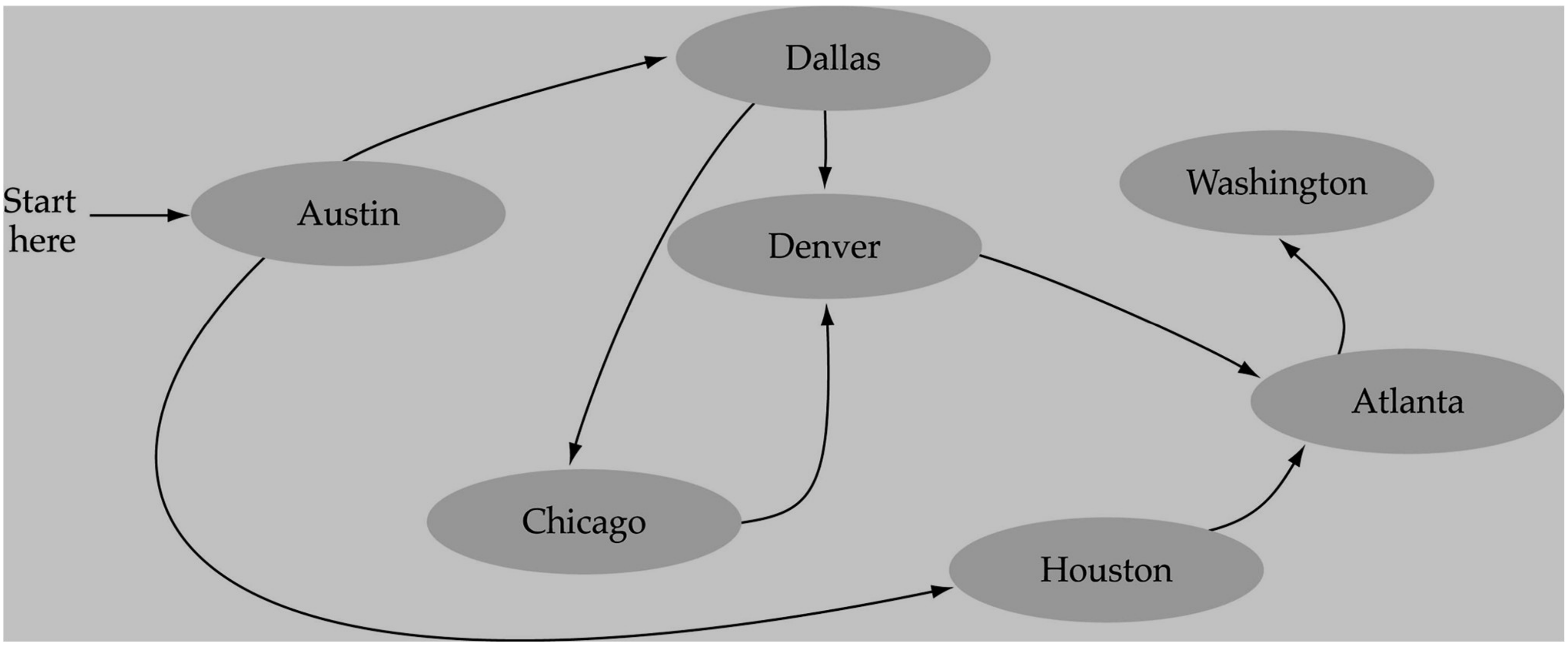
Department of Computer Science and Engineering(CSE)
Varendra University, Rajshahi, Bangladesh

What is Graph?

CSE 2103

- A data structure that consists of a set of nodes (*vertices*) and a set of edges that relate the nodes to each other.

- The set of edges describes relationships among the vertices

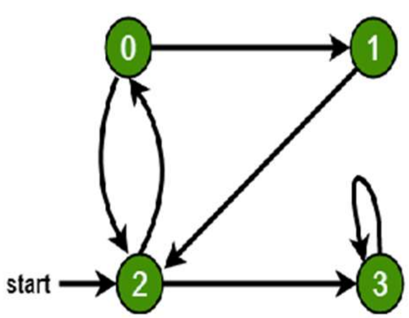


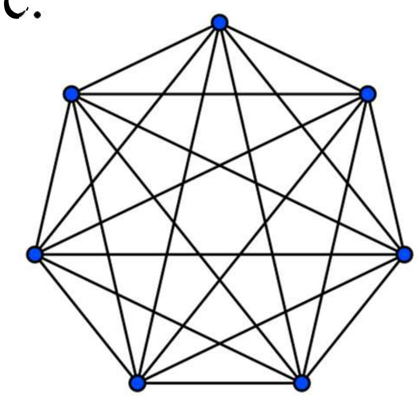
What is Graph?

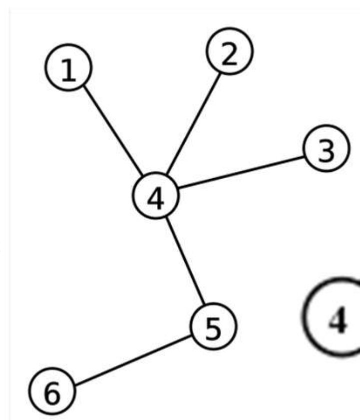
- A graph **G** consists of two things:
  - A set **V** of elements called nodes (or points or vertices)
  - A set **E** of edges such that each edge **e** in **E** is identified with a unique (unorder) pair [**u,v**] of nodes in **V**, denoted by e= [**u,v**].
- Sometimes the parts of a graph is indicated by writing **G=(V, E)**
  - *V(G)*: a finite, nonempty set of vertices
  - *E(G)*: a set of edges (pairs of vertices)

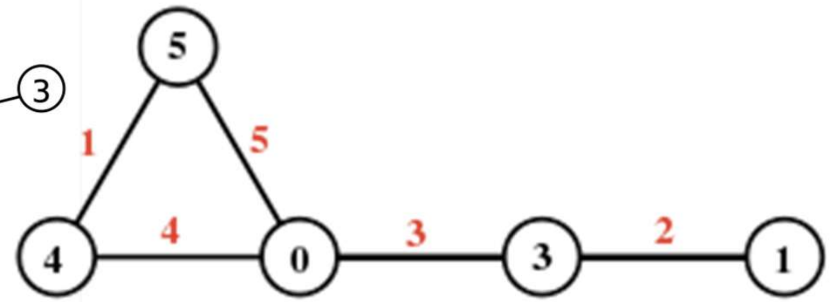
Connected, Complete, Tree, Labeled Graph

- A graph G is Connected if and only if there is a simple path between any two nodes in G.
- A graph G is said to be Complete if every node u in G is adjacent to every other node v in G.
    - A complete graph with n nodes will have (n-1)/2 edges.
- A connected graph T without any circle is called a Tree graph or free tree or, simply, a tree.
- A graph G is said to be Labeled if its edges are assigned data,
    - In particular, G is said to be weighted if each edge e in G is assigned a nonnegative numerical value w(e) called the weight or length of e.





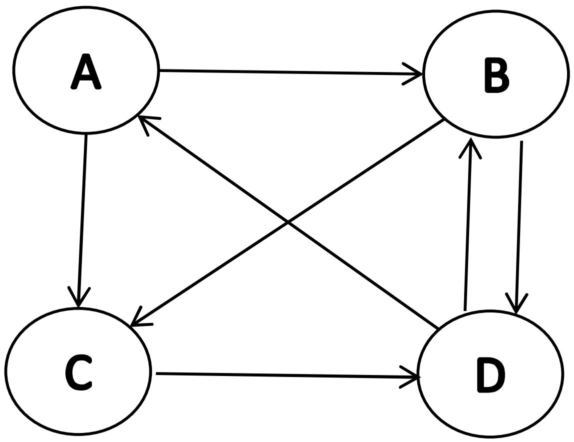


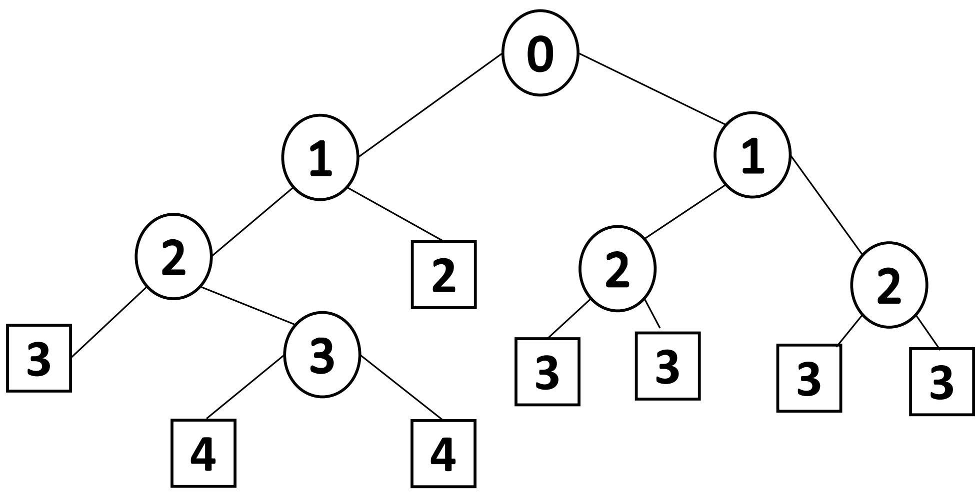


Assignement

CSE 2103

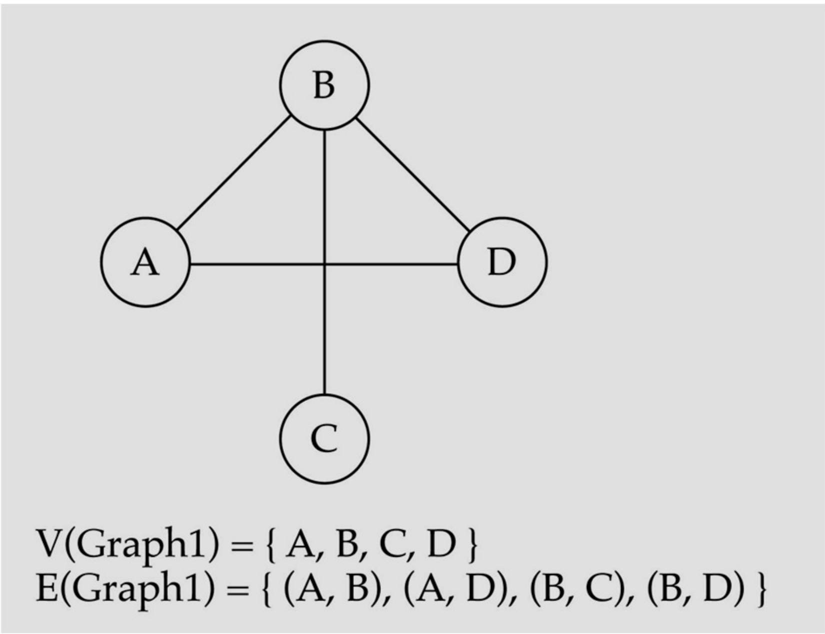
Presentation on Example: 8.1

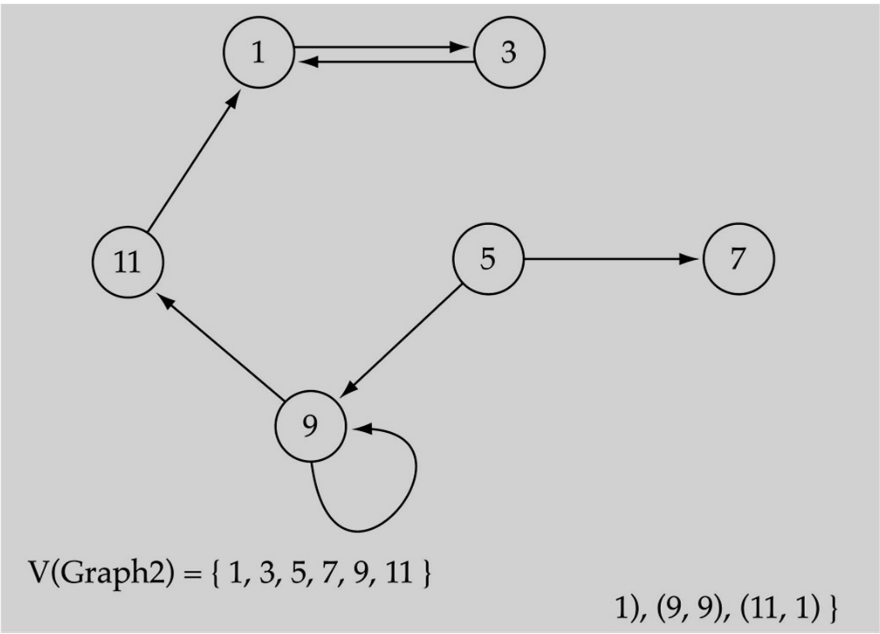




Undirected vs. Directed Graphs

- When the edges in a graph have no direction, the graph is called *undirected*
- When the edges in a graph have a direction, the graph is called *directed* (or *digraph*)





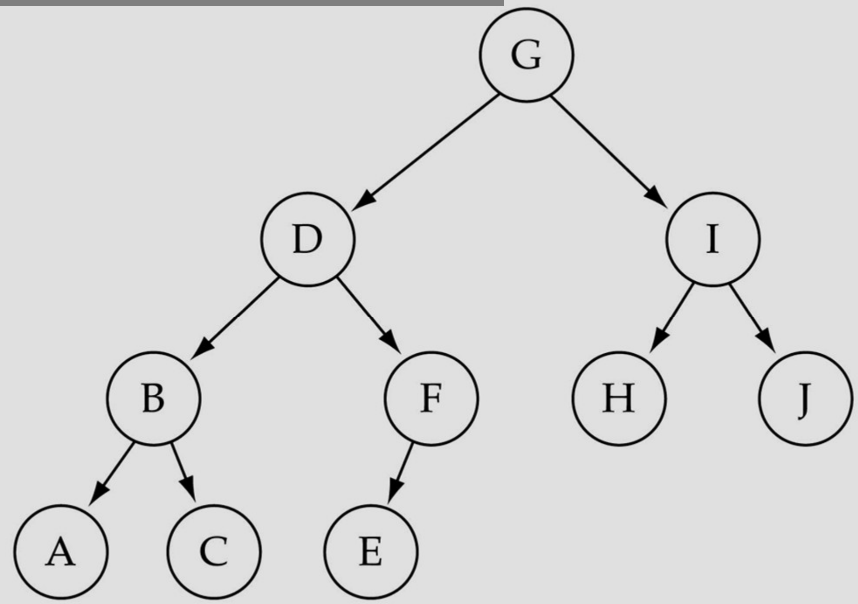
Directed Graphs

- Suppose G is a directed graph with a directed edge e=(u,v).
  - E begins at u and ends at v
  - u is the origin or initial point of e, and v is the destination or terminal point of e
  - u is a predecessor of v, and v is a successor or neighbor of u.
  - u is adjacent to v, and v is adjacent to u.
- A directed graph G is said to be connected, or strongly connected, if for each pair u, v of nodes in G there is a path from u to v, and there is also a path from v to u.

Tree vs. Graph

- Trees are special cases of graphs!!

Tree is a directed graph



V(Graph3) = { A, B, C, D, E, F, G, H, I, J }
E(Graph3) = { (G, D), (G, J), (D, B), (D, F) (I, H), (I, J), (B, A), (B, C), (F, E) }

Graph Representation

- There are two standard ways of maintaining a graph G in the computer memory:
  - The Sequential Representation of G, is by means of its adjacency matrix, A
  - The Linked representation of G, is by means of linked lists of neighbors.

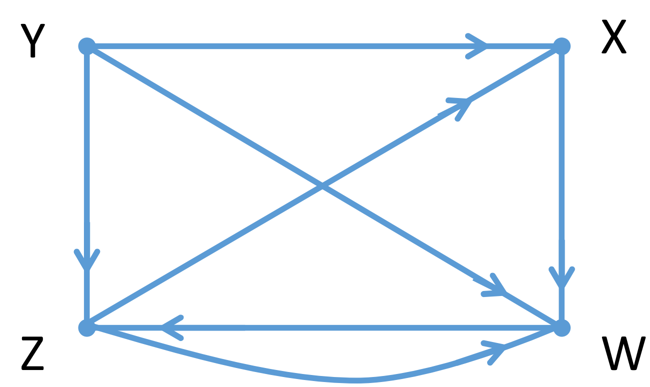
Graph Representation: Adjacency Matrix

- G is a simple directed graph with $v_1, v_2, \ldots, v_m$ vertices
- $A = (a_{ij})$ is the adjacency matrix of G where,

$$a_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases}$$

Example 8.3

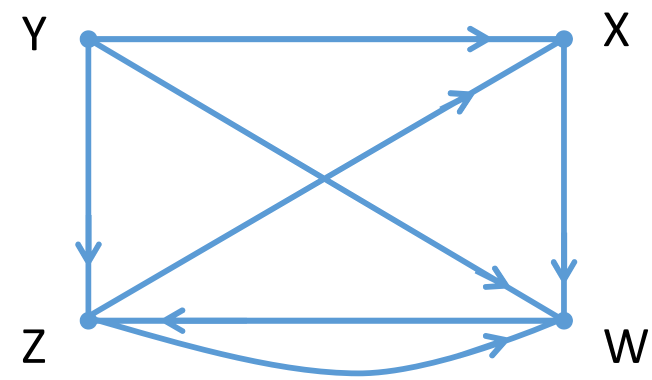
$$A = \begin{array}{c} \begin{matrix} X & Y & Z & W \end{matrix} \\ \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$



Graph Representation: Adjacency Matrix

- Let A is the adjacency matrix of G. Then $a_k(i,j)$ is the ij entry in the matrix A^K, gives the number of path of length K from v_i to v_j.

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$



$$A^2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 2 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix} \quad A^3 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 2 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \quad A^4 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 2 & 0 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

Graph Representation: Path Matrix

CSE 2103

- G is simple directed graph with m nodes v_1, v_2, …., v_m. Then the path matrix P = (p_{ij}) defined as

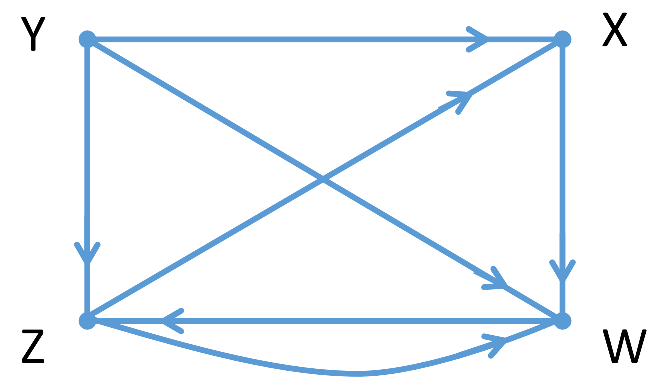
$$p_{ij} = \begin{cases} 1 & if \;\; there \; is \; a \;\; path \;\; from \; v_i \;\; to \; v_j \\ 0 & otherwise \end{cases}$$

- Simple Path: Path from v_i to v_j and $v_i \neq v_j$
- Cycle: Path from v_i to v_j and $v_i = v_j$
- p_{ij} = 1 if and only if there is a nonzero number in ij entry of the matrix $B_m = A + A^2 + A^3 + \ldots + A^m$

Graph Representation: Path Matrix

CSE 2103

$$A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \qquad A^2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 2 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$$



$$A^3 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 2 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \qquad A^4 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 2 & 0 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

Thus,

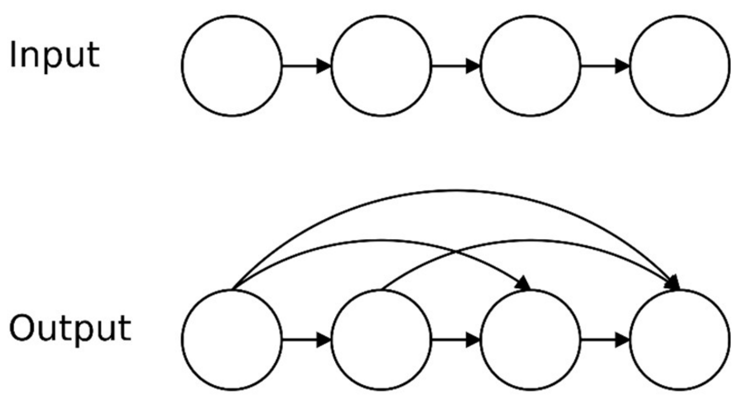
$$B_4 = \begin{pmatrix} 1 & 0 & 2 & 3 \\ 5 & 0 & 6 & 8 \\ 3 & 0 & 3 & 5 \\ 2 & 0 & 3 & 3 \end{pmatrix}$$

and

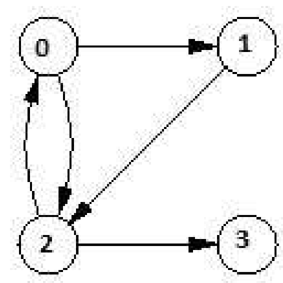
$$P = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

Transitive Closure

- The transitive closure of a graph G is defined to be the graph G' such that G' has the same node as G and there is an edge (v_i,v_j) in G' whenever there is a path from v_i to v_j in G.
- 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.
- For example, consider below graph



$$\begin{matrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{matrix}$$



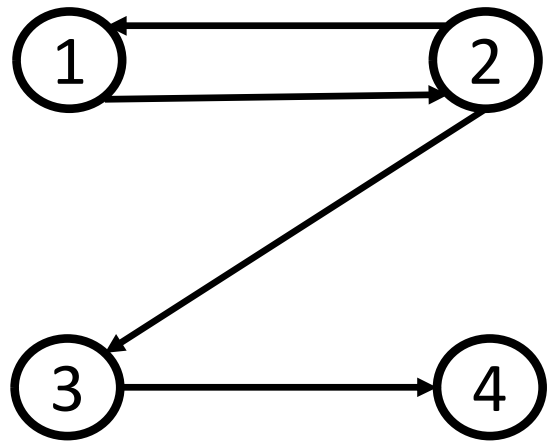

```
Transitive closure of above graphs is
     1 1 1 1
     1 1 1 1
     1 1 1 1
     0 0 0 1
```

Examples of Transitive Closure

CSE 2103

Let $A = \{1, 2, 3, 4\}$, and let $R = \{(1,2), (2,3), (3,4), (2,1)\}$.

Find the transitive closure of R.



Transitive Closure

CSE 2103

Let ***R*** be a relation on a set A. Let R^{∞} be the transitive closure of ***R.***

Three methods for finding R^{∞}:

a) Digraph Approach

b) Adjacency Matrix method

c) Warshall's Algorithm

Warshall's Algorithm for Path Matrix

CSE 2103

- Let G be a directed graph with m nodes $v_1, v_2, v_3, \ldots v_m$.
- Suppose we want to find the path matrix P of the graph G.
- Warshall's algorithm is much more efficient than calculating the powers of the adjacency matrix A.
- First we define m-square Boolean matrices $P_0, P_1, P_2, \ldots P_m$ as follows.
- Let $P_k[i,j]$ denote ij entry of the matrix P_k. Then

$$P_k[i,j] = \begin{cases} 1 & \text{If there is a simple path from } v_i \text{ to } v_j \text{ which does not use any other nodes except possible } v_1, v_2, \ldots, v_k \\ 0 & otherwise \end{cases}$$

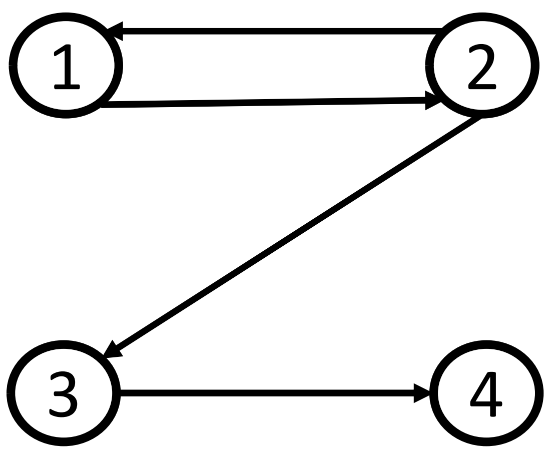
Warshall's Algorithm for Path Matrix: Example

Let $A = \{1, 2, 3, 4\}$, and let $R = \{(1,2), (2,3), (3,4), (2,1)\}$.

Find the transitive closure of R.

Warshall's Algorithm convert R to M_R

$R = \{(1,2), (2,3), (3,4), (2,1)\}$



$$\rightarrow M_R = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

Warshall's Algorithm for Path Matrix: Example

CSE 2103

Step 1:

$$W_0 = M_R = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

Observe value 1's in column 1 : 2

Observe value 1's in row 1 : 2

So we have new 1 in position (2,2).

$$W_1 = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

Warshall's Algorithm for Path Matrix

CSE 2103

Step 2:

$$W_1 = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

Observe value 1's in column 2 : 1,2

Observe value 1's in row 2 : 1,2,3

So we have new 1 in position:

(1,1),(1,2),(1,3)(2,1),(2,2),(2,3)

(if 1's are not already there)

$$W_2 = \begin{vmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

Warshall's Algorithm for Path Matrix

CSE 2103

Step 3:

$$W_2 = \begin{vmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

Observe value 1's in column 3 : 1,2

Observe value 1's in row 3 : 4

So we have new 1 in position: (1,4),(2,4)
(if 1's are not already there)

$$W_3 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

Warshall's Algorithm for Path Matrix

CSE 2103

Step 4:

$$W_3 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

Observe value 1's in column 4 : 1,2,3

Observe value 1's in row 4 : N/A

So no new 1 to add.

$$W_4 = W_3 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

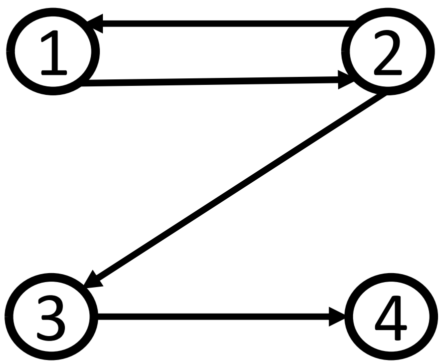
$$\therefore \; R^{\infty} = \{(1,1), (1,2), (1,3), (1,4), (2,1), (2,2), (2,3), (2,4), (3,4)\}$$

Warshall's Algorithm for Path Matrix

CSE 2103

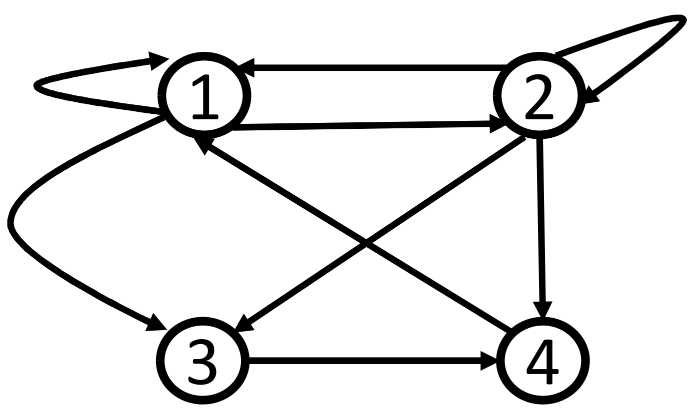
Let A = {1, 2, 3, 4}, and let
R = {(1,2), (2,3), (3,4), (2,1)}.

$$M_R = \begin{vmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$



$$W_4 = W_3 = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

$$\therefore \quad R^{\infty} = \{(1,1),\ (1,2),\ (1,3),\ (1,4),\ (2,1),\ (2,2),\ (2,3),\ (2,4),\ (3,4)\}$$



Warshall's Algorithm for Path Matrix

CSE 2103

- $P_0[i,j] = 1$ if there is an edge from v_i to v_j
- $P_1[i,j] = 1$ if there is a simple path from v_i to v_j which does not use any other nodes except possibly v_1.
- $P_2[i,j] = 1$ if there is a simple path from v_i to v_j which does not use any other nodes except possibly v_1, v_2.
- The element of P_k can be obtained as

$$P_k[i,j] = P_{k-1}[i,j] \vee (P_{k-1}[i,k] \wedge P_{k-1}[k,j])$$

Warshall's Algorithm for Path Matrix

CSE 2103

- G: Directed graph, A: Adjacency matrix, M: Nodes

1. Repeat for I,J = 1,2,….,M

   If A[I,J] == 0 then set P[I,J] = 0

   Else set P[I,J] = 1

2. Repeat steps 3 and 4 for K = 1,2,…,M
3. Repeat step 4 for I = 1,2,…., M
4. Repeat for J = 1,2,…..,M

   Set $P[I,J] = P[I,J] \vee (P[I,K] \wedge P[K,J])$

5. Exit

Shortest Path Algorithm

- Let G be a directed graph with m nodes, $v_1, v_2, \ldots\ldots, v_m$.
- Suppose G is weighted, and $w(e)$ is called the weight or length of the edge e.
- Then the weight matrix $W = (w_{ij})$ is defined as:

$$w_{ij} = \begin{cases} w(e) & \text{if there is an edge } e \text{ from } v_i \text{ to } v_j \\ 0 & \text{if there is no edge from } v_i \text{ to } v_j \end{cases}$$

- The path matrix P tells us whether or not there are paths between the nodes.
- Now we find a matrix Q which tell us the lengths of the shortest paths between the nodes, or more exactly, $Q = (q_{ij})$:

q_{ij} = length of the shortest path from v_i to v_j

Shortest Path Algorithm

CSE 2103

- We define sequence of matrices $Q_0, Q_1, \ldots\ldots, Q_m$ (like $P_0, P_1, \ldots\ldots P_m$), whose entries are:

$$Q_k[i.j] = MIN(Q_{k-1}[i,j], Q_{k-1}[i,k] + Q_{k-1}[k,j])$$

- Q_0 is same as the weight matrix W where the 0 is replaced by the infinity (∞)

- The final matrix Q_m will be the desired matrix Q

Shortest Path Algorithm

- G: directed weighted graph, W: weight matrix, M: Nodes

1. Repeat for I,J = 1,2,….,M

   If W[I,J] == 0 then set Q[I,J] = ∞

   Else set Q[I,J] = W[I,J]

2. Repeat steps 3 and 4 for K = 1,2,…,M

   3. Repeat step 4 for I = 1,2,…., M

      4. Repeat for J = 1,2,…..,M

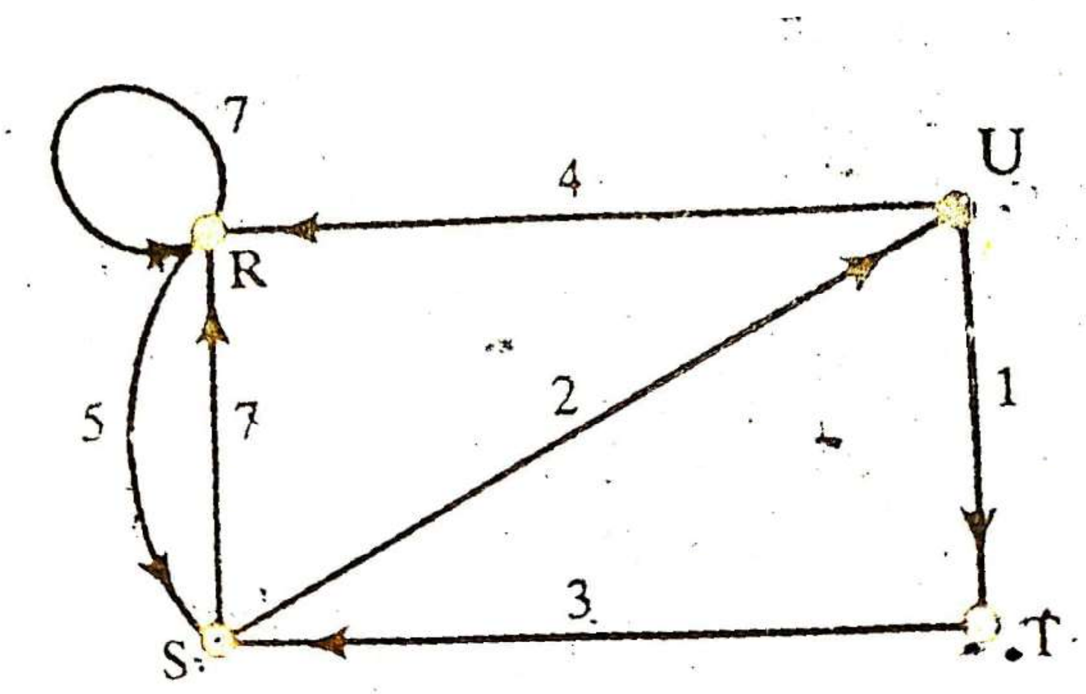
         Set Q[I,J] = MIN(Q[I,J], Q[I,K] + Q[K,J])

5. Exit

Shortest Path Algorithm: Example

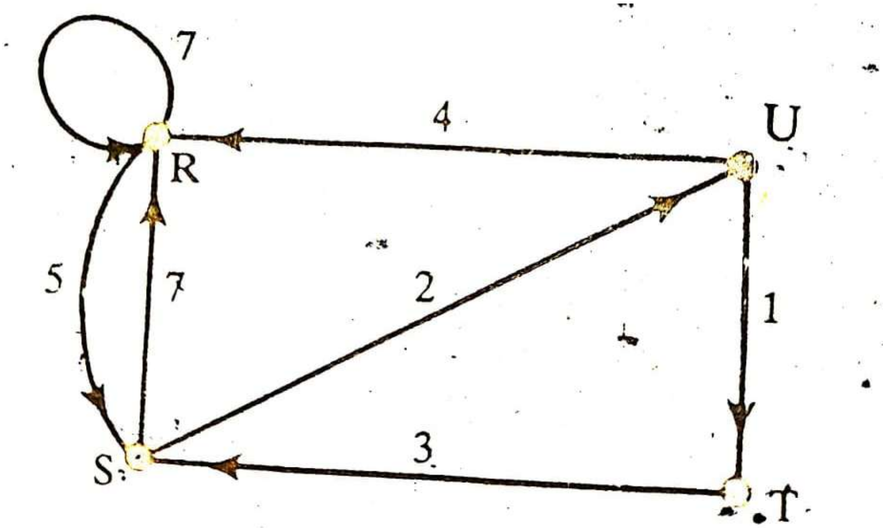
CSE 2103

- Consider the following weighted graph:
- Assume v_1=R, v_2=S, v_3=T, and v_4=U
- Then the weighted matrix W of G is as follows:



$$\begin{array}{cc} & \begin{array}{cccc} R & S & T & U \end{array} \\ W = & \begin{pmatrix} 7 & 5 & 0 & 0 \\ 7 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 4 & 0 & 1 & 0 \end{pmatrix} \begin{array}{c} R \\ S \\ T \\ U \end{array} \end{array}$$

Shortest Path Algorithm: Example



$$
\begin{array}{cc}
 & \begin{array}{cccc} R & S & T & U \end{array} \\
W = & \begin{pmatrix} 7 & 5 & 0 & 0 \\ 7 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 4 & 0 & 1 & 0 \end{pmatrix} \begin{array}{c} R \\ S \\ T \\ U \end{array}
\end{array}
$$

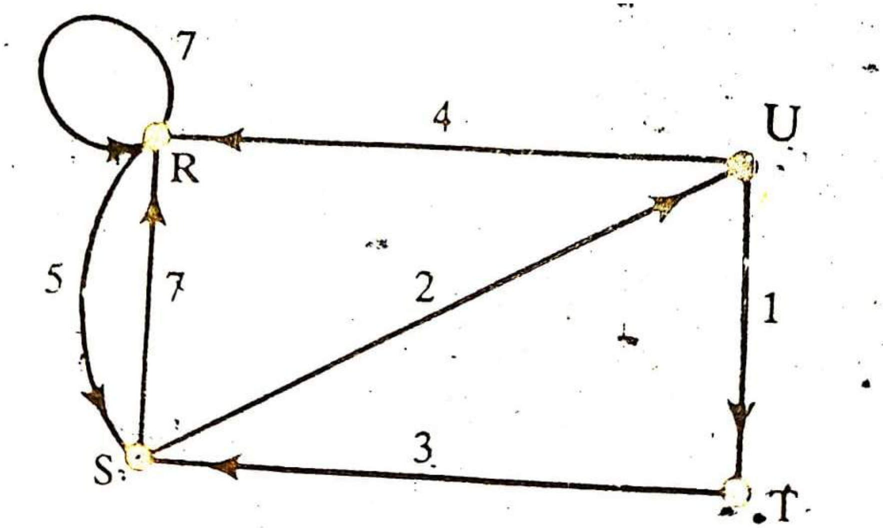
•Applying the modified Warshall's algorithm, we obtain the following matrices:

$$
Q_0 = \begin{pmatrix} 7 & 5 & \infty & \infty \\ 7 & \infty & \infty & 2 \\ \infty & 3 & \infty & \infty \\ 4 & \infty & 1 & \infty \end{pmatrix}
\qquad
\begin{pmatrix} RR & RS & - & - \\ SR & - & - & SU \\ - & TS & - & - \\ UR & - & UT & - \end{pmatrix}
$$

$$
Q_1 = \begin{pmatrix} 7 & 5 & \infty & \infty \\ 7 & 12 & \infty & 2 \\ \infty & 3 & \infty & \infty \\ 4 & 9 & 1 & \infty \end{pmatrix}
\qquad
\begin{pmatrix} RR & RS & - & - \\ SR & SRS & - & SU \\ - & TS & - & - \\ UR & URS & UT & - \end{pmatrix}
$$

$$
Q_1[4, 2] = \mathrm{MIN}(Q_0[4, 2],\ Q_0[4, 1] + Q_0[1, 2]) = \mathrm{MIN}(\infty, 4 + 5) = 9
$$

Shortest Path Algorithm: Example



$$W = \begin{array}{c} \\ \\ \\ \\ \\ \end{array}\!\!\begin{array}{c} \begin{array}{cccc} R & S & T & U \end{array} \\ \begin{pmatrix} 7 & 5 & 0 & 0 \\ 7 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 4 & 0 & 1 & 0 \end{pmatrix} \end{array} \begin{array}{c} \\ R \\ S \\ T \\ U \end{array}$$

•Applying the modified Warshall's algorithm, we obtain the following matrices

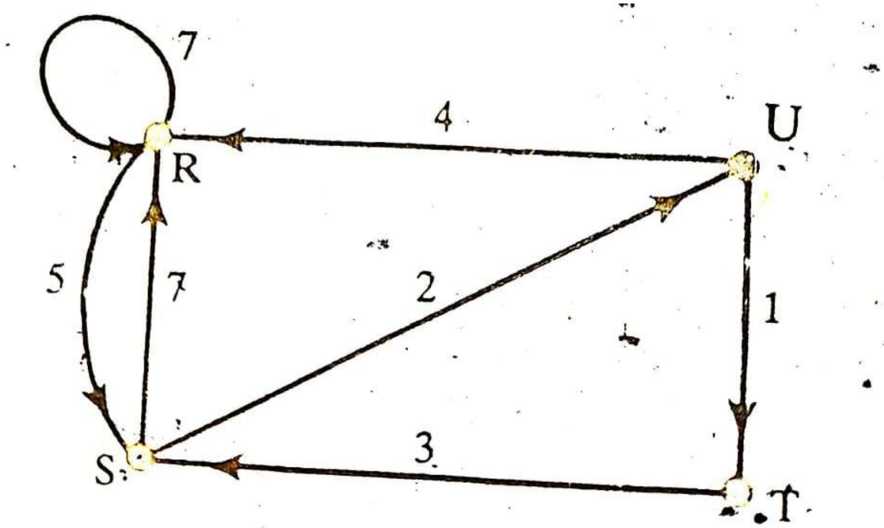
$$Q_1 = \begin{pmatrix} 7 & 5 & \infty & \infty \\ 7 & 12 & \infty & 2 \\ \infty & 3 & \infty & \infty \\ 4 & 9 & 1 & \infty \end{pmatrix} \qquad \begin{pmatrix} RR & RS & — & — \\ SR & SRS & — & SU \\ — & TS & — & — \\ UR & URS & UT & — \end{pmatrix}$$

$$Q_2 = \begin{pmatrix} 7 & 5 & \infty & 7 \\ 7 & 12 & \infty & 2 \\ 10 & 3 & \infty & 5 \\ 4 & 9 & 1 & 11 \end{pmatrix} \qquad \begin{pmatrix} RR & RS & — & RSU \\ SR & SRS & — & SU \\ TSR & TS & — & TSU \\ UR & URS & UT & URS \end{pmatrix}$$

$$Q_2[1, 3] = \text{MIN}(Q_1[1, 3],\ Q_1[1, 2] + Q_1[2, 3]) = \text{MIN}(\infty, 5 + \infty) = \infty$$

Shortest Path Algorithm: Example

CSE 2103



$$W = \begin{array}{c} \\ \\ \end{array}\begin{array}{cccc} R & S & T & U \end{array}$$

$$\begin{array}{c|c} & \begin{array}{cccc} R & S & T & U \end{array} \\ W = & \begin{pmatrix} 7 & 5 & 0 & 0 \\ 7 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 4 & 0 & 1 & 0 \end{pmatrix} \begin{array}{c} R \\ S \\ T \\ U \end{array} \end{array}$$

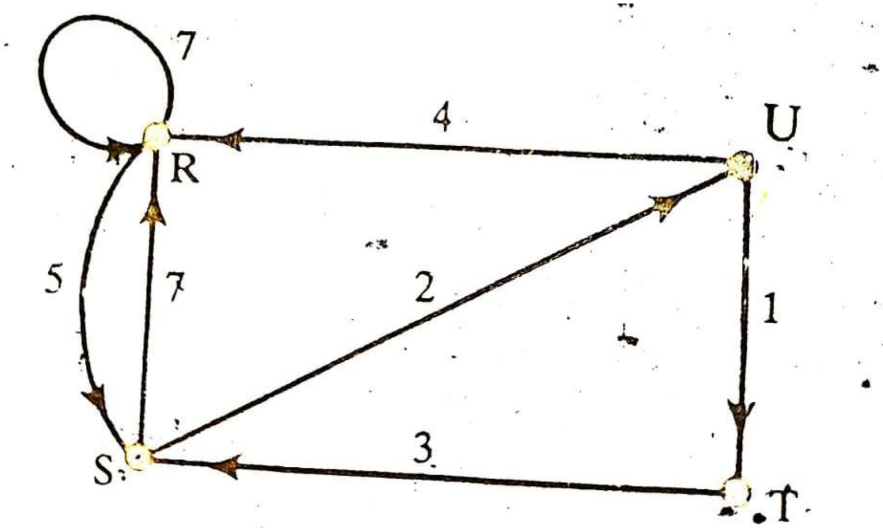
•Applying the modified Warshall's algorithm, we obtain the following matrices:

$$Q_2 = \begin{pmatrix} 7 & 5 & \infty & 7 \\ 7 & 12 & \infty & 2 \\ 10 & 3 & \infty & 5 \\ 4 & 9 & 1 & 11 \end{pmatrix} \quad \begin{pmatrix} RR & RS & - & RSU \\ SR & SRS & - & SU \\ TSR & TS & - & TSU \\ UR & URS & UT & URS \end{pmatrix}$$

$$Q_3 = \begin{pmatrix} 7 & 5 & \infty & 7 \\ 7 & 12 & \infty & 2 \\ 10 & 3 & \infty & 5 \\ 4 & 4 & 1 & 6 \end{pmatrix} \quad \begin{pmatrix} RR & RS & - & RSU \\ SR & SRS & - & SU \\ TSR & TS & - & TSU \\ UR & UTS & UT & UTSU \end{pmatrix}$$

$$Q_3[4, 2] = \text{MIN}(Q_2[4, 2],\ Q_2[4, 3] + Q_2[3, 2]) = \text{MIN}(9, 3 + 1) = 4$$

Shortest Path Algorithm: Example



$$W = \begin{array}{c} \\ \\ \end{array}\begin{array}{cccc} R & S & T & U \end{array}$$

$$W = \begin{pmatrix} 7 & 5 & 0 & 0 \\ 7 & 0 & 0 & 2 \\ 0 & 3 & 0 & 0 \\ 4 & 0 & 1 & 0 \end{pmatrix}\begin{array}{c} R \\ S \\ T \\ U \end{array}$$

•Applying the modified Warshall's algorithm, we obtain the following matrices:

$$Q_3 = \begin{pmatrix} 7 & 5 & \infty & 7 \\ 7 & 12 & \infty & 2 \\ 10 & 3 & \infty & 5 \\ 4 & 4 & 1 & 6 \end{pmatrix} \qquad \begin{pmatrix} RR & RS & — & RSU \\ SR & SRS & — & SU \\ TSR & TS & — & TSU \\ UR & UTS & UT & UTSU \end{pmatrix}$$

$$Q_4 = \begin{array}{c} R \\ S \\ T \\ U \end{array}\begin{pmatrix} 7 & 5 & 8 & 7 \\ 7 & 11 & 3 & 2 \\ 9 & 3 & 6 & 5 \\ 4 & 4 & 1 & 6 \end{pmatrix} \qquad \begin{pmatrix} RR & RS & RSUT & RSU \\ SR & SURS & SUT & SU \\ TSUR & TS & TSUT & TSU \\ UR & UTS & UT & UTSU \end{pmatrix}$$

$$Q_4[3, 1] = \mathrm{MIN}(Q_3[3, 1],\ Q_3[3, 4] + Q_3[4, 1]) = \mathrm{MIN}(10, 5 + 4) = 9$$