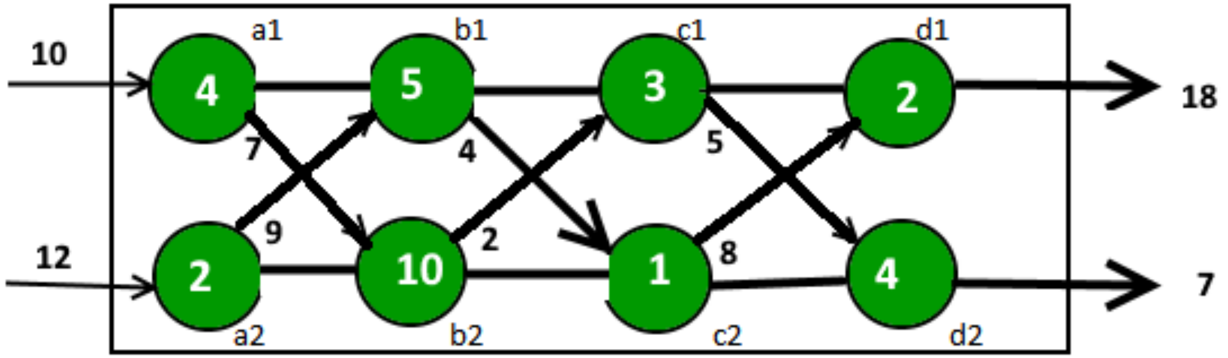
# Design and Analysis of Algorithm

Lecture-15:
Dynamic Programming

# Contents

# Question

Find the shortest path for the assembly line given in the figure




What is the time complexity of Assembly line scheduling?

# Question

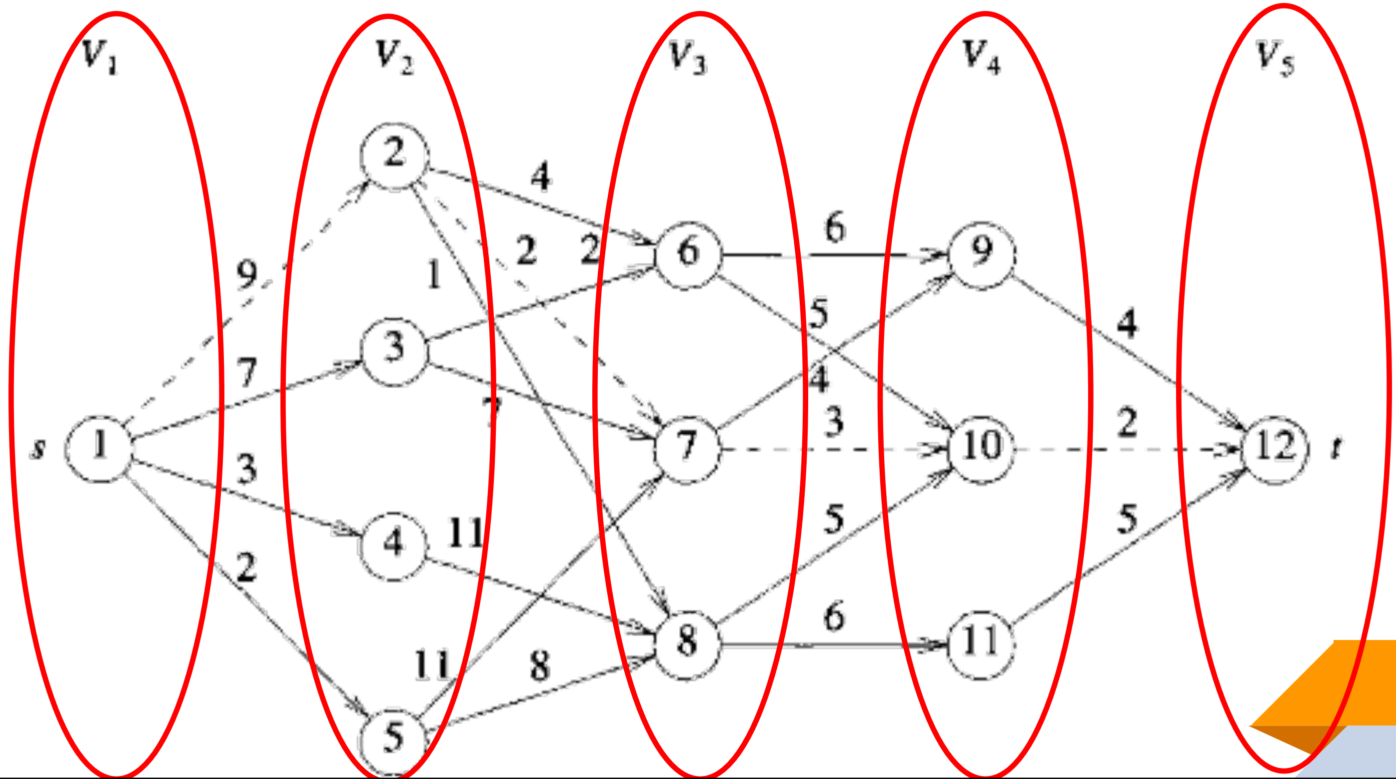
Consider the following assembly line problem:

time_to_reach[2][3] = {{17, 2, 7}, {19, 4, 9}}
time_spent[2][4] = {{6, 5, 15, 7}, {5, 10, 11, 4}}
entry_time[2] = {8, 10}
exit_time[2] = {10, 7}
num_of_stations = 4

What is the minimum time required to build the car chassis?

# The shortest path in multistage graphs

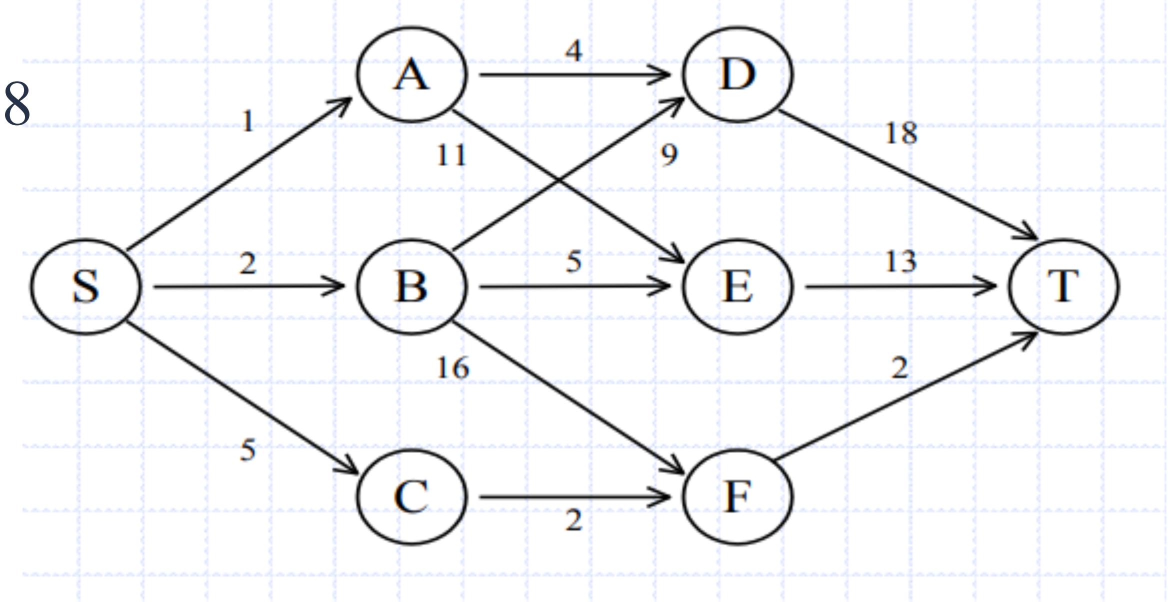
A multi stage graph $G = (V, E)$ is a directed graph in which the vertices are partitioned into $k > 2$ disjoint sets $V_i$, $1 \le i < k$. In addition, if $\{u, v\}$ is an edge in $E$, then $u \in V_i$ and $v \in V_{i+1}$ for some $i$, $1 \le i < k$.



# The shortest path in multistage graphs

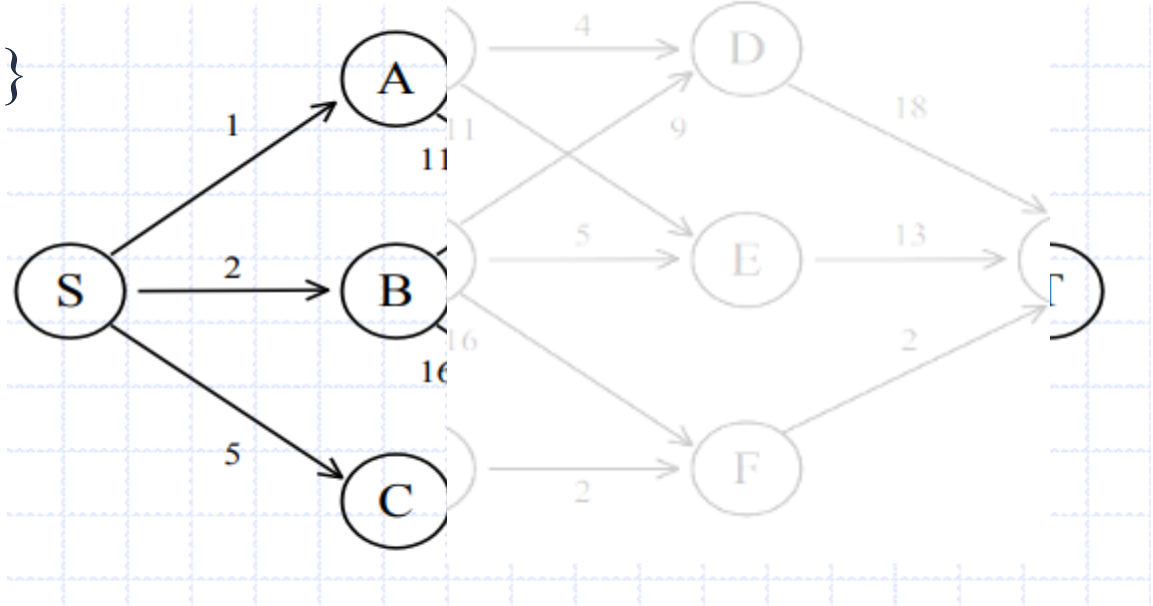
Solution as per greedy is: $S\ A\ D\ T \Rightarrow 1{+}4{+}18$

Minimum cost is: $S\ C\ F\ T \Rightarrow 5{+}2{+}2$



# The shortest path in multistage graphs

d(S, T) = min{1+d(A, T), 2+d(B, T), 5+d(C, T)}




d(A, T) = min{4+d(D, T), 11+d(E, T)}

d(D,T)= 18
d(E,T)= 13

$$d(A, T) = \min(4 + 18), 11 + 13) = 22$$

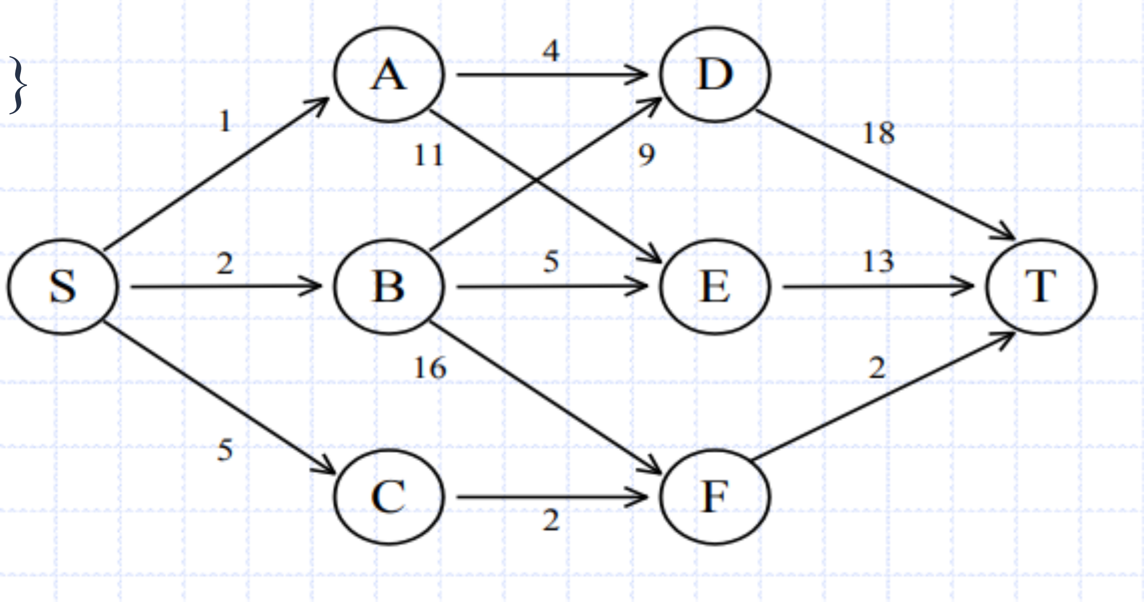
# The shortest path in multistage graphs

d(B, T) = min{(9+d(D, T), 5+d(E, T),16+d(F,T) }

$$d(B,T) = \min(9 + 18)\,, 5 + 13{,}16 + 2) = 18$$

d(C, T) = 2+d(F,T)

$$d(C,T) = 2 + 2 = 4$$



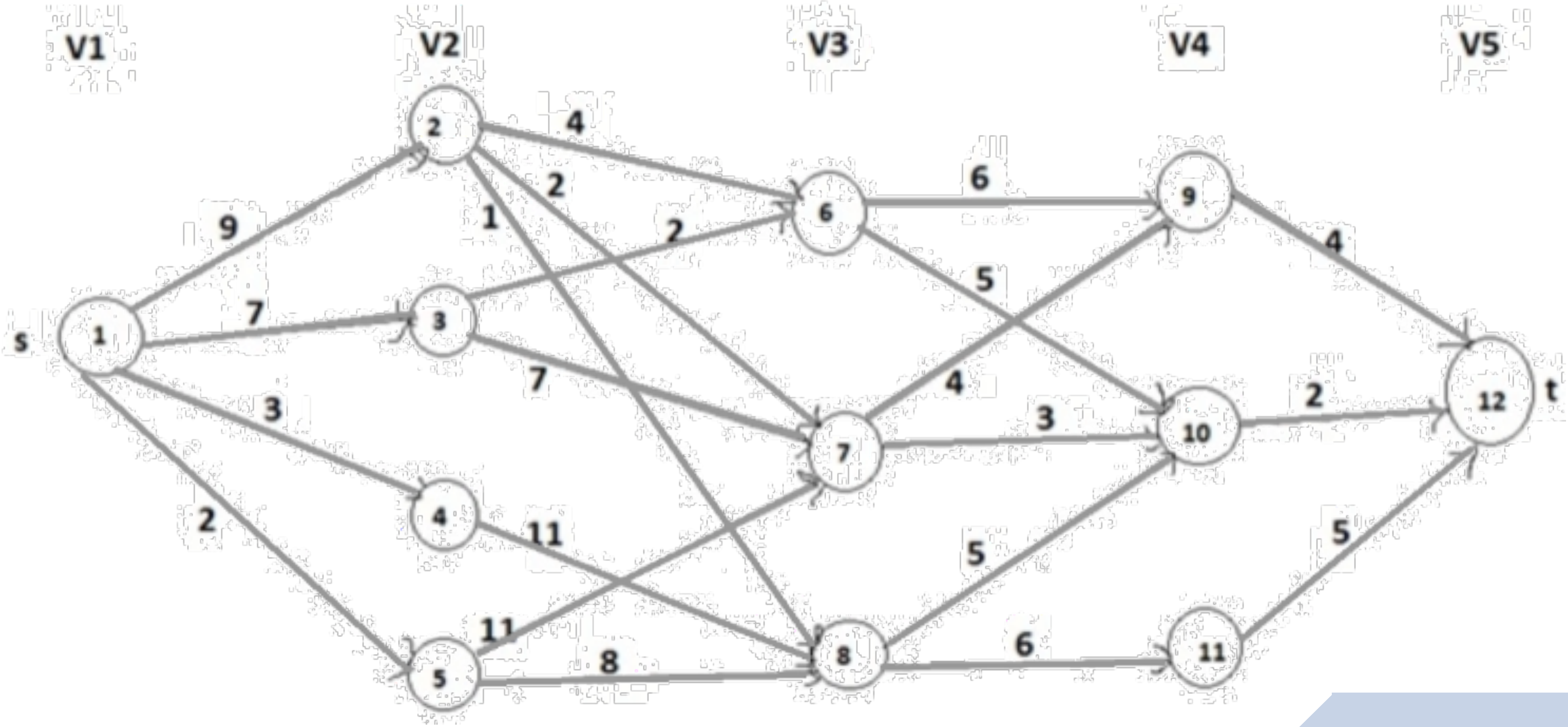

d(S, T) = min{1+d(A, T), 2+d(B, T), 5+d(C, T)}

d(S, T) = min{1+22, 2+18, 5+4}=9

- Backword Reasoning
- Forward Approach

# Question 1

What is the shortest path from source $s$ to destination $t$.




1-2-7-10-12

# Forward Reasoning (Backward Approach)

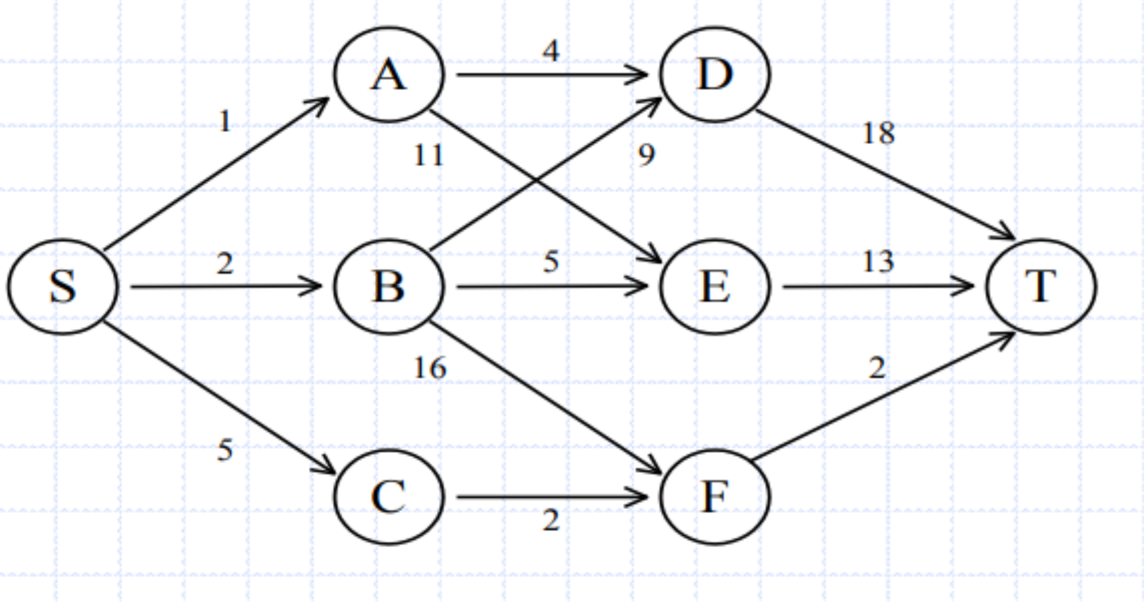
$d(S, A) = 1$ $d(S, B) = 2$ $d(S, C) = 5$

$d(S,D)=\min\{d(S, A)+d(A, D),d(S, B)+d(B, D)\}$

$= \min\{ 1+4, 2+9 \} = 5$

$d(S,E)=\min\{d(S, A)+d(A, E),d(S, B)+d(B, E)\}$

$= \min\{ 1+11, 2+5 \} = 7$




$d(S,F)=\min\{d(S, A)+d(A, F),d(S, B)+d(B, F)\}$

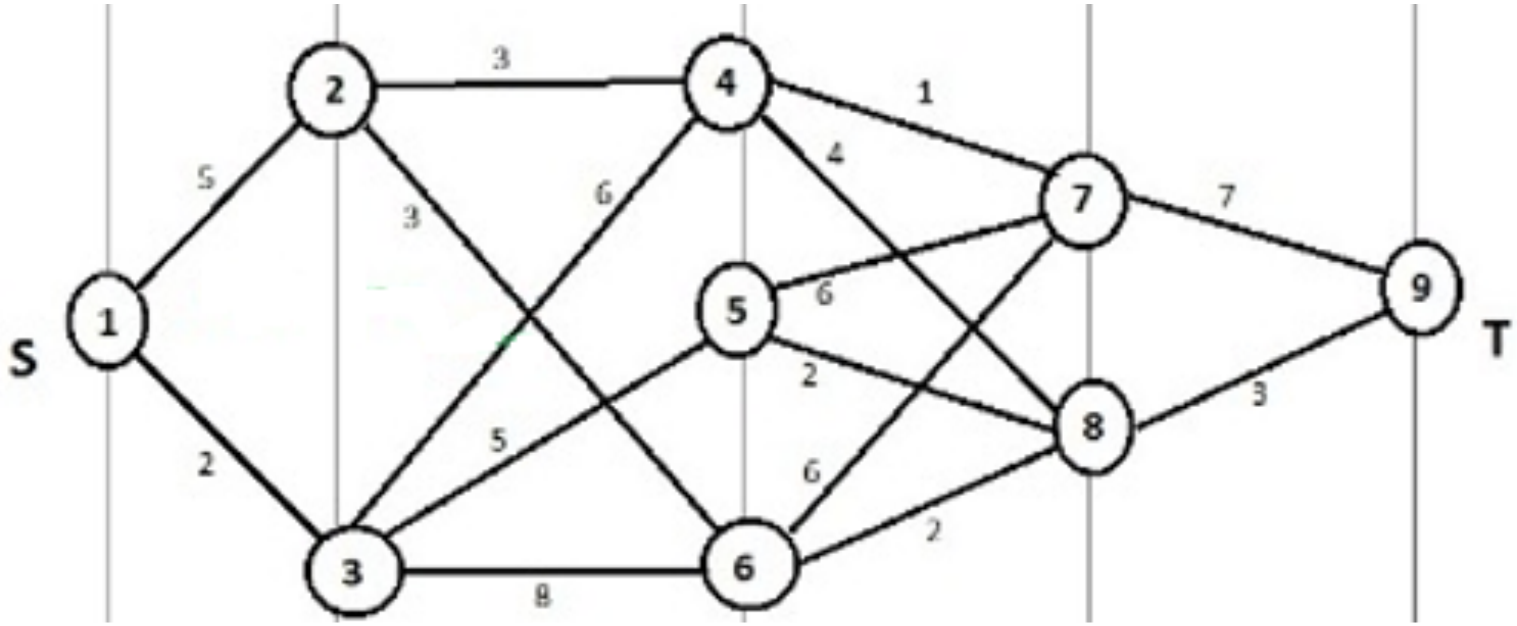
$= \min\{ 2+16, 5+2 \} = 7$

$d(S,T) = \min\{d(S, D)+d(D, T),d(S,E)+ d(E,T), d(S, F)+d(F, T)\}$

$= \min\{ 5+18, 7+13, 7+2 \} = 9$

# Question 2

What is the shortest path from source $s$ to destination $t$.



# All Pair Shortest Path

## Problem

Given a directed graph $G=(V, E)$, and a weight function $w: E \rightarrow \mathbb{R}$, for each pair of vertices $u$, $v$, compute the shortest path weight $\delta(u, v)$, and a shortest path if exists.

**Goal:** create an $n \times n$ matrix $D = (d_{ij})$ of shortest path distances

i.e., $d_{ij} = \delta(v_i, v_j)$

**Possible solution:**

Run a SSSP algorithm *n* times, one for each vertex as the source.

# Adjacency Matrix Representation of Graphs

- $n$ x $n$ matrix W = ($\omega_{ij}$) of edge weights :

$$\omega_{ij} = \begin{cases} \omega(v_i, v_j) & \text{if } (v_i, v_j) \in E \\ \infty & \text{if } (v_i, v_j) \notin E \end{cases}$$

- assume $\omega_{ii} = 0$ for all $v_i \in$ V, because
  - no negative weight cycle

    $\Rightarrow$ shortest path to itself has no edge,

    i.e., $\delta(v_i, v_i) = 0$

# Floyd Warshall Algorithm

We examine a shortest path $i$ to $j$ in G. This path originates at vertex $i$ and goes through some intermediate vertices (possibly none) and terminates at vertex $j$.

If k is an intermediate vertex on this shortest path, then the sub paths from $i$ to $k$ and from $k$ to $j$ must be shortest paths from i to k and k to j, respectively. Otherwise, the $i$ to $j$ path is not of minimum length

$$A(i,j) = \min \{ \min_{1 \le k \le n} \{A^{k-1}(i,k) + A^{k-1}(k,j)\}, cost(i,j)\}$$

# Steps to find all pair shortest path

Remove all the self loops and parallel edges (keeping the lowest weight edge) from the graph.

Write the initial distance matrix.

- For diagonal elements (representing self-loops), distance value = 0
- For vertices having a direct edge between them, distance value = weight of that edge.
- For vertices having no direct edge between them, distance value = ∞.

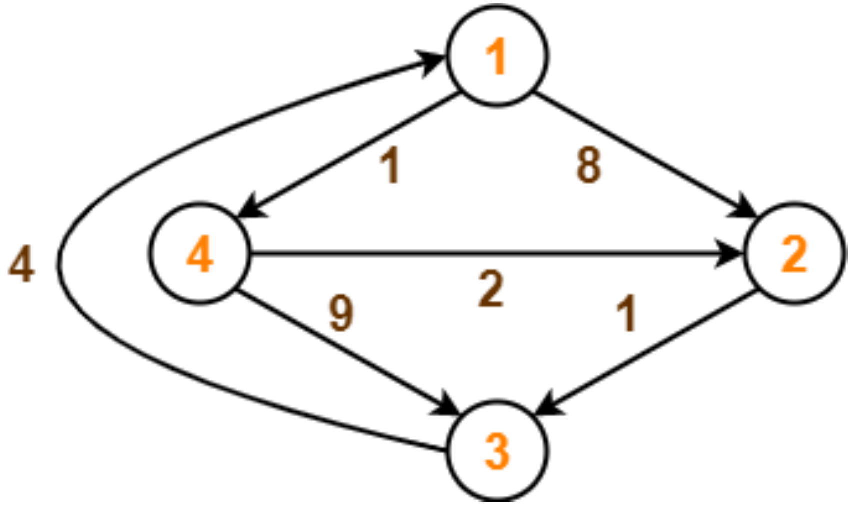
# Example

Initial distance matrix

$$A^k(i,j) = \min\ \{A^{k-1}(i,j),\ A^{k-1}(i,k) + A^{k-1}(k,j)\},\quad k \geq 1$$

$$D_0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 8 & \infty & 1 \\ \infty & 0 & 1 & \infty \\ 4 & \infty & 0 & \infty \\ \infty & 2 & 9 & 0 \end{bmatrix} \end{array}$$

$$D_2 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 8 & 9 & 1 \\ \infty & 0 & 1 & \infty \\ 4 & 12 & 0 & 5 \\ \infty & 2 & 3 & 0 \end{bmatrix} \end{array}$$




$$D_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 8 & \infty & 1 \\ \infty & 0 & 1 & \infty \\ 4 & 12 & 0 & 5 \\ \infty & 2 & 9 & 0 \end{bmatrix} \end{array}$$

$$D_3 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 8 & 9 & 1 \\ 5 & 0 & 1 & 6 \\ 4 & 12 & 0 & 5 \\ 7 & 2 & 3 & 0 \end{bmatrix} \end{array}$$

$$D_4 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & 4 & 1 \\ 5 & 0 & 1 & 6 \\ 4 & 7 & 0 & 5 \\ 7 & 2 & 3 & 0 \end{bmatrix} \end{array}$$

# Example

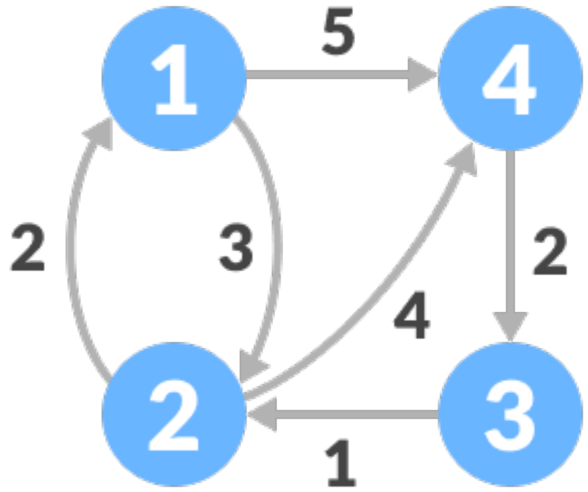
$$A^0 = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 3 & \infty & 5 \\ 2 & 0 & \infty & 4 \\ \infty & 1 & 0 & \infty \\ \infty & \infty & 2 & 0 \end{bmatrix} \end{array}$$

$$A^1 = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 3 & \infty & 5 \\ 2 & 0 & 9 & 4 \\ \infty & 1 & 0 & 8 \\ \infty & \infty & 2 & 0 \end{bmatrix} \end{array}$$

$$A^2 = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 3 & 9 & 5 \\ 2 & 0 & 9 & 4 \\ 3 & 1 & 0 & 5 \\ \infty & \infty & 2 & 0 \end{bmatrix} \end{array}$$

$$A^3 = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 3 & 9 & 5 \\ 2 & 0 & 9 & 4 \\ 3 & 1 & 0 & 5 \\ 5 & 3 & 2 & 0 \end{bmatrix} \end{array}$$

$$A^4 = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 3 & 7 & 5 \\ 2 & 0 & 6 & 4 \\ 3 & 1 & 0 & 5 \\ 5 & 3 & 2 & 0 \end{bmatrix} \end{array}$$



# Floyd Warshall **Algorithm**

$$A^k(i,j) = \min\ \{A^{k-1}(i,j),\ A^{k-1}(i,k) + A^{k-1}(k,j)\},\quad k \geq 1$$

```
Algorithm AllPaths(cost, A, n)
// cost[1 : n, 1 : n] is the cost adjacency matrix of a graph with
// n vertices; A[i, j] is the cost of a shortest path from vertex
// i to vertex j. cost[i, i] = 0.0, for 1 ≤ i ≤ n.
{
    for i := 1 to n do
        for j := 1 to n do
            A[i, j] := cost[i, j]; // Copy cost into A.
    for k := 1 to n do
        for i := 1 to n do
            for j := 1 to n do
                A[i, j] := min(A[i, j], A[i, k] + A[k, j]);
}
```

Complexity of Floyd Warshall algorithm is $O(n^3)$.