Data Structures & Algorithms Design- SS ZG519
Lecture - 17

BITS Pilani
Pilani Campus

Dr. Padma Murali

Lecture 17 Topics

- Shortest Path Problem

innovate achieve lead

BITS Pilani
Pilani Campus

- **Shortest Paths**

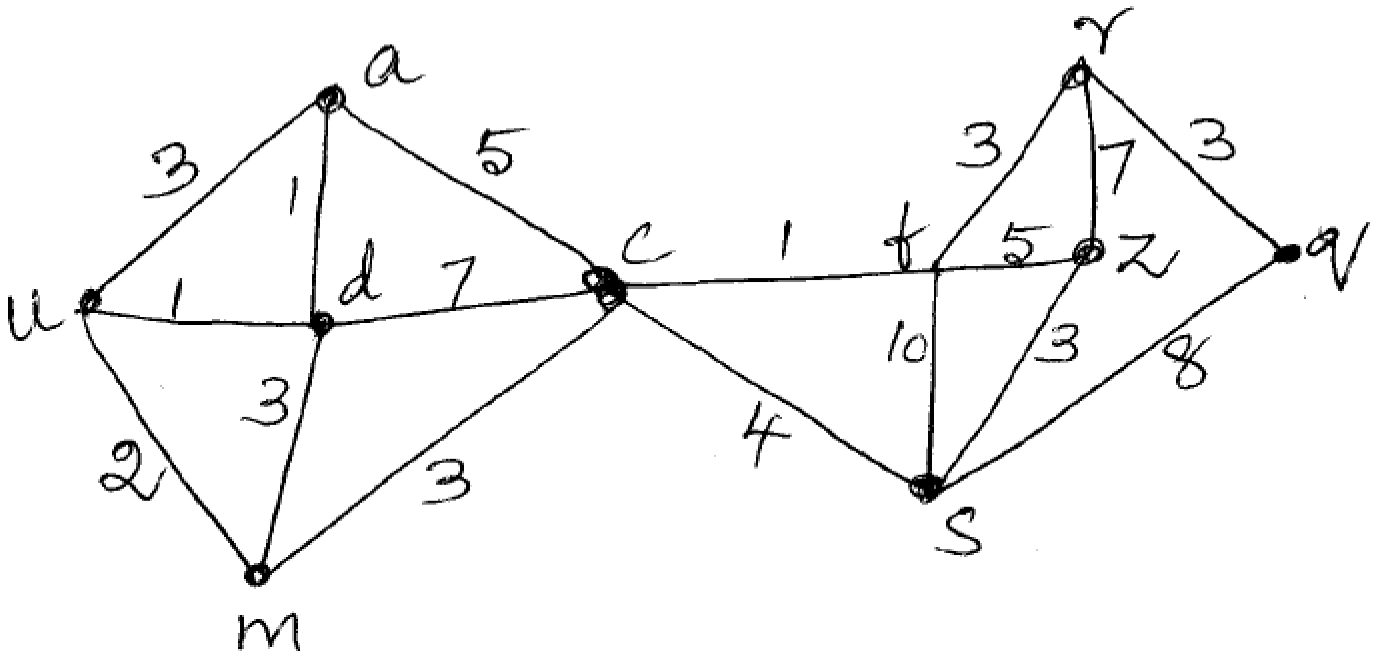
Shortest Paths

- Dijkstra’s algorithm
- Floyd Warshall’s algorithm

Example 2

Example: 2

Find the shortest path from u to every other vertex.



Example 2

Recent permanent labelled vertex	λ(u)	λ(a)	λ(d)	λ(m)	λ(c)	λ(s)	λ(r)	λ(z)	λ(q)	λ(f)
	$\boxed{0}$	∞	∞	∞	∞	∞	∞	∞	∞	∞
u	$\boxed{0}$	3	$\boxed{1}$	2	∞	∞	∞	∞	∞	∞
d	$\boxed{0}$	$\boxed{2}$	$\boxed{1}$	2	8	∞	∞	∞	∞	∞
a	$\boxed{0}$	$\boxed{2}$	$\boxed{1}$	$\boxed{2}$	7	∞	∞	∞	∞	∞
m	$\boxed{0}$	$\boxed{2}$	$\boxed{1}$	$\boxed{2}$	$\boxed{5}$	∞	∞	∞	∞	∞
c	$\boxed{0}$	$\boxed{2}$	$\boxed{1}$	$\boxed{2}$	$\boxed{5}$	9	∞	∞	∞	$\boxed{6}$
f	$\boxed{0}$	$\boxed{2}$	$\boxed{1}$	$\boxed{2}$	$\boxed{5}$	$\boxed{9}$	9	11	∞	$\boxed{6}$
s	$\boxed{0}$	$\boxed{2}$	$\boxed{1}$	$\boxed{2}$	$\boxed{5}$	$\boxed{9}$	$\boxed{9}$	11	17	$\boxed{6}$
r	$\boxed{0}$	$\boxed{2}$	$\boxed{1}$	$\boxed{2}$	$\boxed{5}$	$\boxed{9}$	$\boxed{9}$	$\boxed{11}$	12	$\boxed{6}$
z	$\boxed{0}$	$\boxed{2}$	$\boxed{1}$	$\boxed{2}$	$\boxed{5}$	$\boxed{9}$	$\boxed{9}$	$\boxed{11}$	$\boxed{12}$	$\boxed{6}$
q	same as above row.									

Dijkstra's Algorithm

Single-source shortest path problem:

- No negative-weight edges: $w(u, v) > 0 \ \forall \ (u, v) \in E$

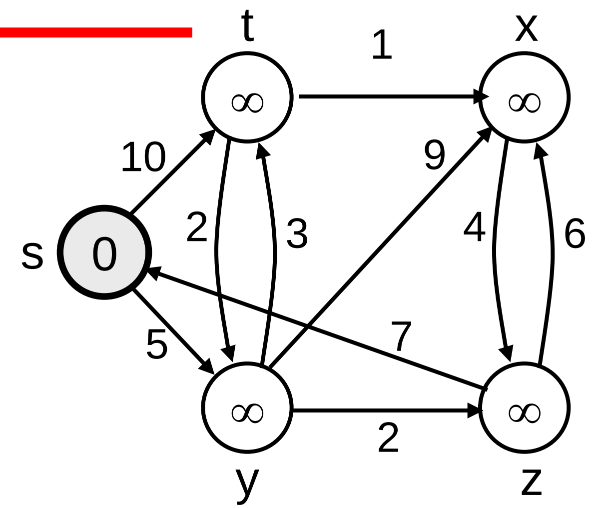
Maintains two sets of vertices:

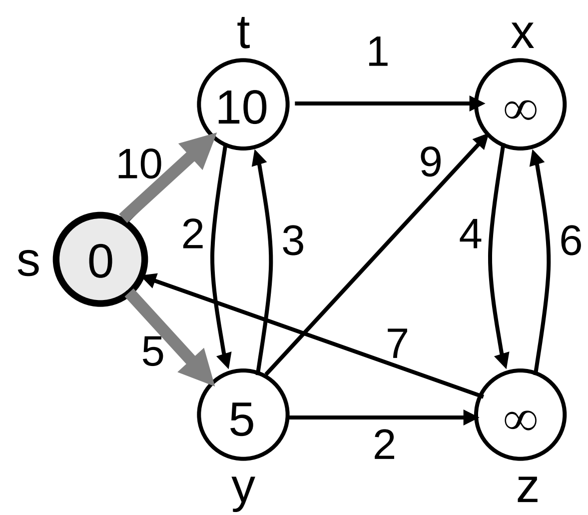
- S = vertices whose final shortest-path weights have already been determined
- Q = vertices in V – S: min-priority queue
  - Keys in Q are estimates of shortest-path weights (d[v])

Repeatedly select a vertex $u \in V - S$, with the minimum shortest-path estimate d[v]

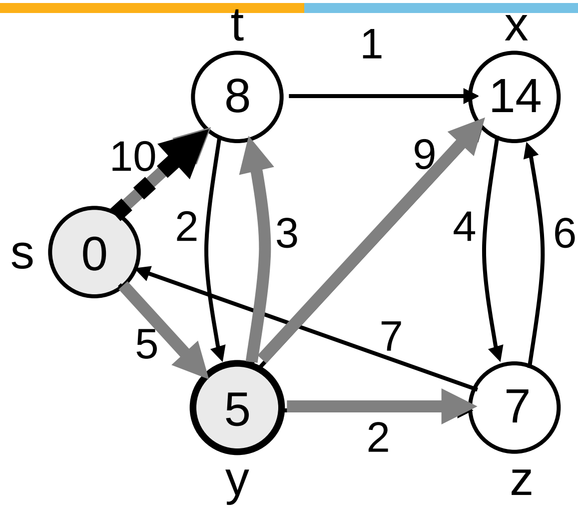
Dijkstra (G, w, s)

```
INITIALIZE-SINGLE-SOURCE(V, s)
S ← ∅
Q ← V[G]
while Q ≠ ∅
    do u ← EXTRACT-MIN(Q)
        S ← S ∪ {u}
        for each vertex v ∈ Adj[u]
            do RELAX(u, v, w)
```

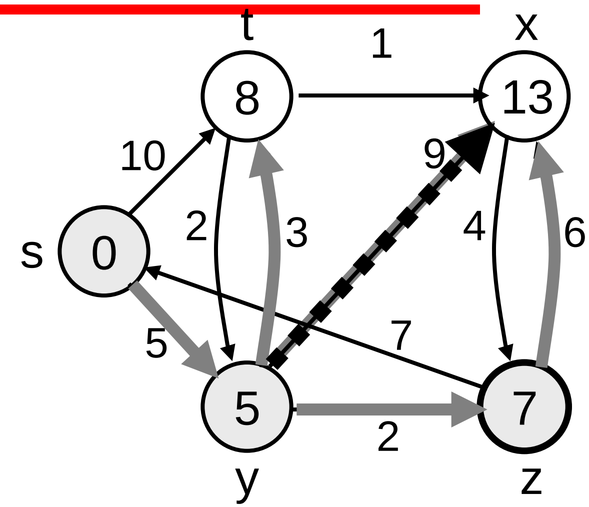




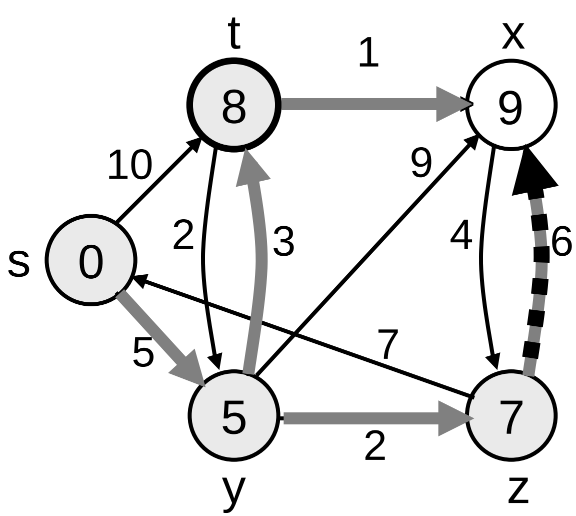
Example



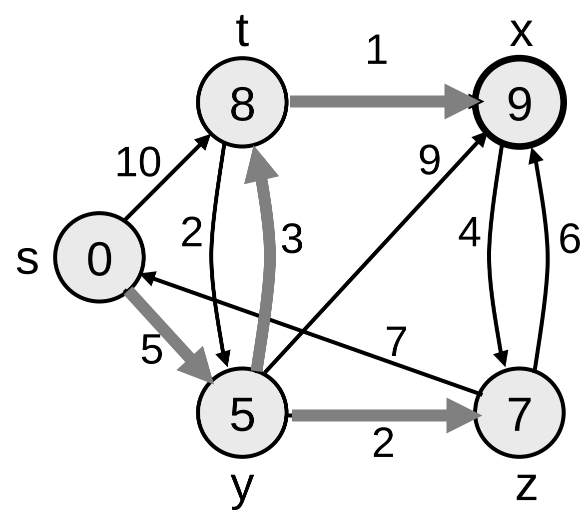
t
x
s
y
z
8
14
0
5
7
1
10
2
3
9
4
6
5
7
2



t
x
s
y
z
8
13
0
5
7
1
10
2
3
9
4
6
5
7
2



t
x
s
y
z
8
9
0
5
7
1
10
2
3
9
4
6
5
7
2



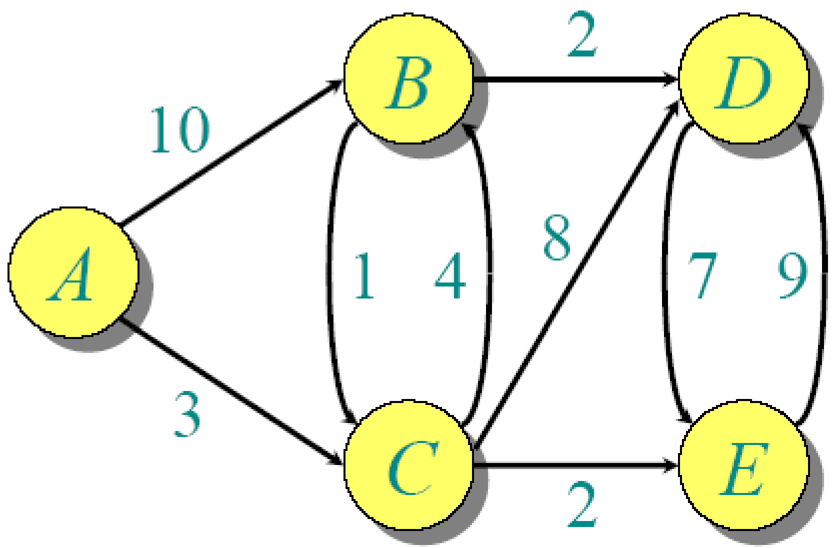
t
x
s
y
z
8
9
0
5
7
1
10
2
3
9
4
6
5
7
2

Dijkstra (G, w, s)

1. INITIALIZE-SINGLE-SOURCE(V, *s*) $\leftarrow$ $\Theta(V)$
2. S $\leftarrow \emptyset$
3. Q $\leftarrow$ V[G] $\leftarrow$ O(V) build min-heap
4. **while** Q $\neq \emptyset$ $\leftarrow$ Executed O(V) times
5. **do** u $\leftarrow$ EXTRACT-MIN(Q) $\leftarrow$ O(lgV)
6. S $\leftarrow$ S $\cup$ {u}
7. **for** each vertex v $\in$ *Adj*[u]
8. **do** RELAX(u, v, w) $\leftarrow$ O(E) times; O(lgV)

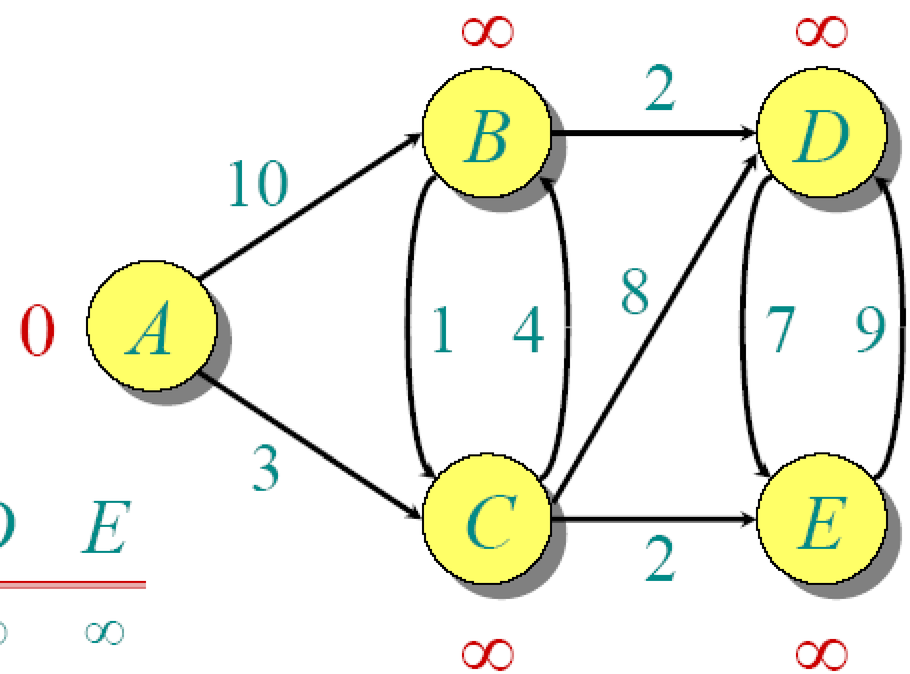
Running time: $O(V\lg V + E\lg V) = O(E\lg V)$

Example of Dijkstra's Algorithm



Example of Dijkstra's Algorithm

Initialize:

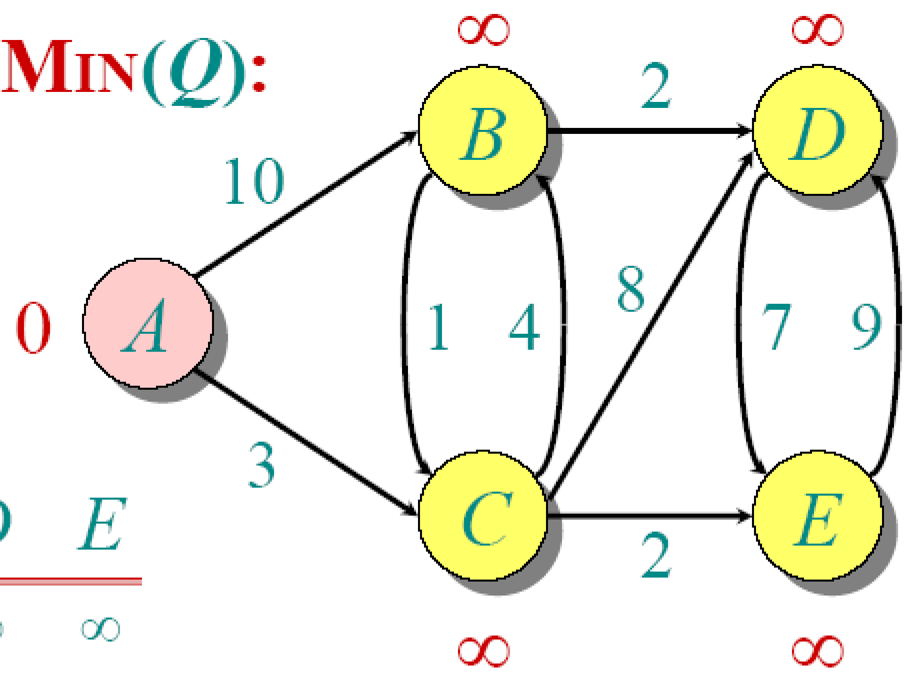


Q:	A	B	C	D	E
	0	∞	∞	∞	∞

S: {}

Example of Dijkstra's Algorithm

"A" ← EXTRACT-MIN(Q):

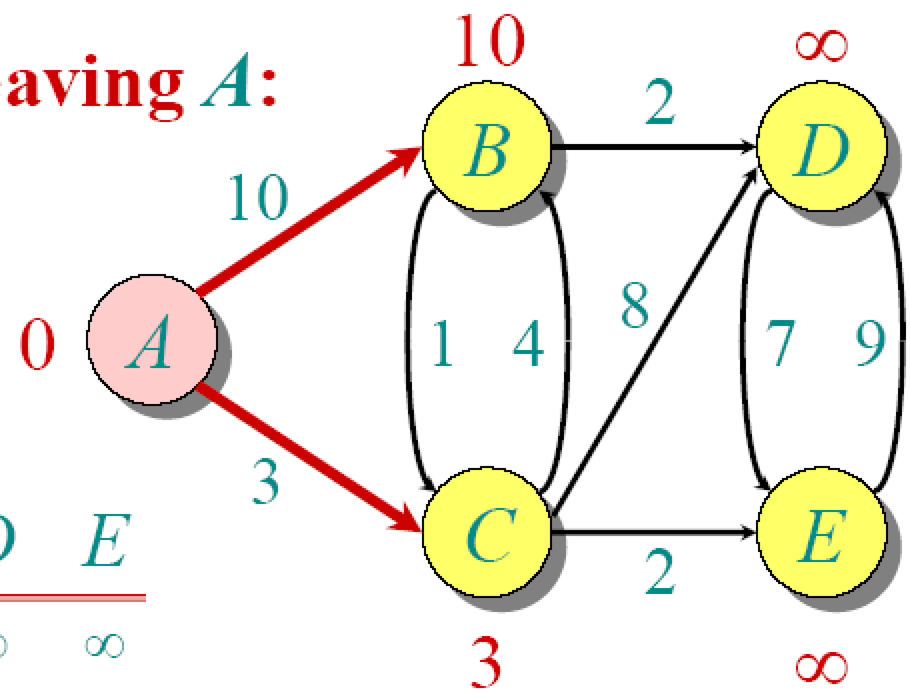


Q:	A	B	C	D	E
	0	∞	∞	∞	∞

S: { A }

Example of Dijkstra's Algorithm

Relax all edges leaving *A*:

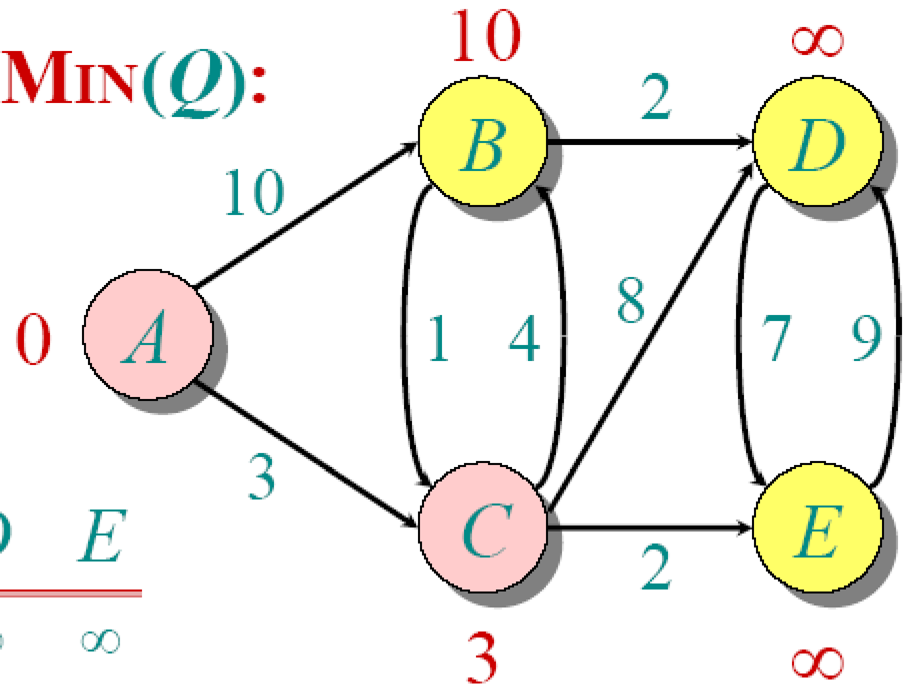


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A }

Example of Dijkstra's Algorithm

"C" $\leftarrow$ EXTRACT-MIN(Q):

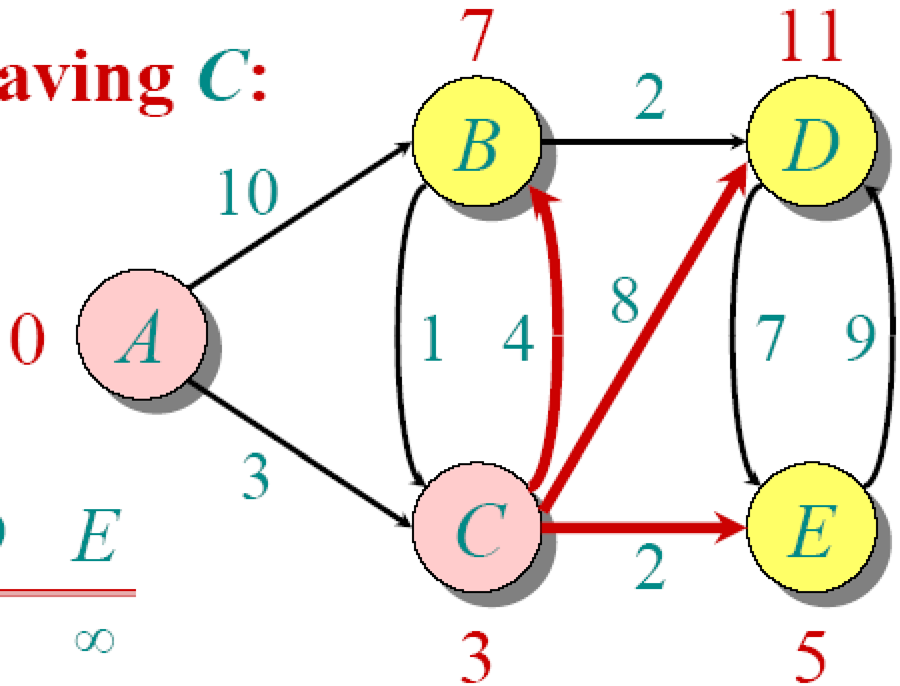


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞

S: { A, C }

Example of Dijkstra's Algorithm

Relax all edges leaving *C*:

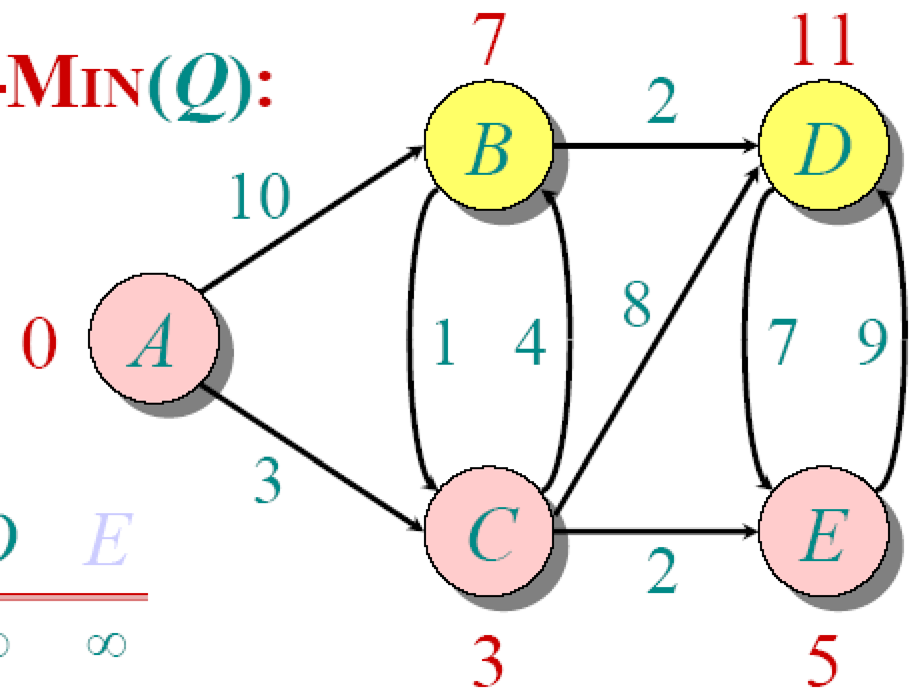


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5

S: { A, C }

Example of Dijkstra's Algorithm

"E" $\leftarrow$ EXTRACT-MIN(Q):

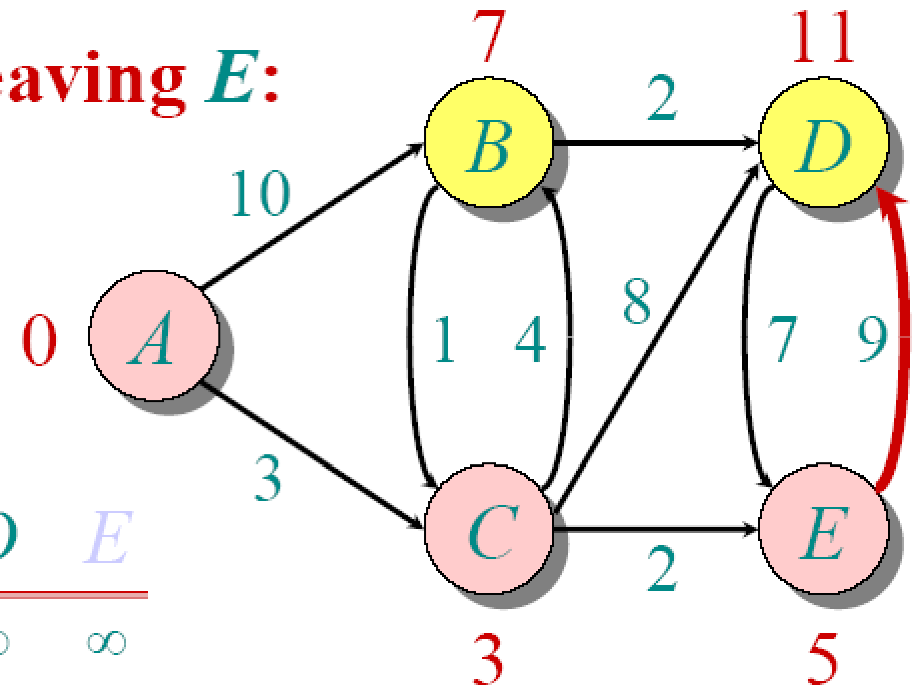


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5

S: { A, C, E }

Example of Dijkstra's Algorithm

Relax all edges leaving *E*:

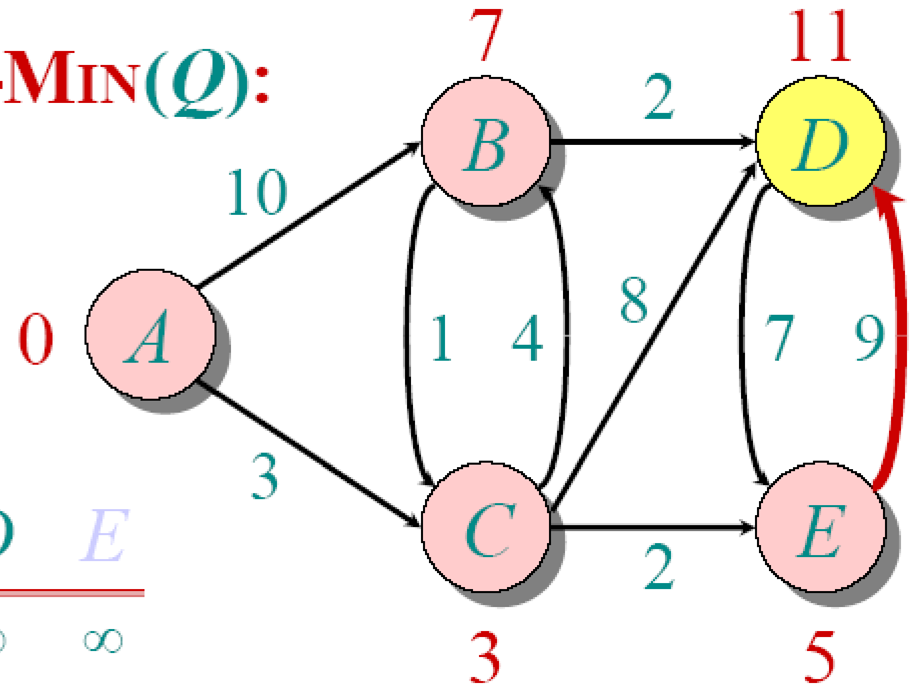


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

S: { A, C, E }

Example of Dijkstra's Algorithm

"B" ← EXTRACT-MIN(Q):

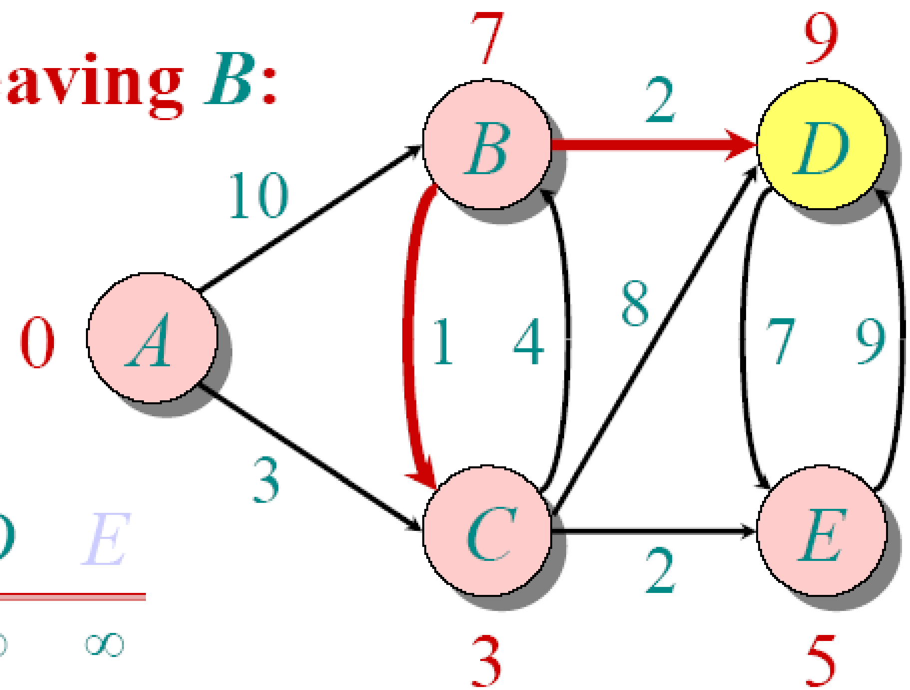


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	

S: { A, C, E, B }

Example of Dijkstra's Algorithm

Relax all edges leaving *B*:

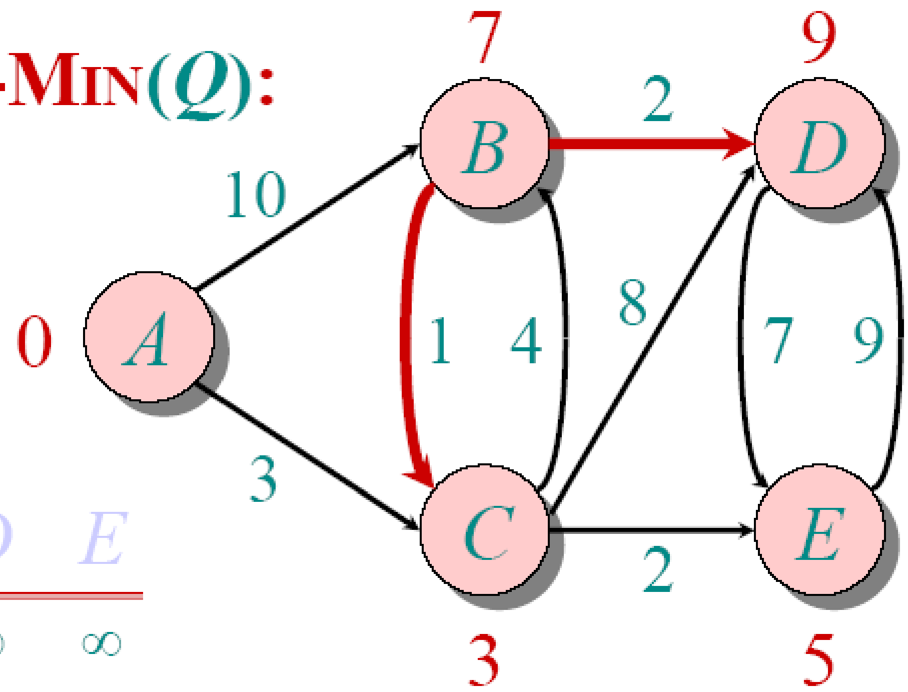


Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	
				9	

S: { A, C, E, B }

Example of Dijkstra's Algorithm

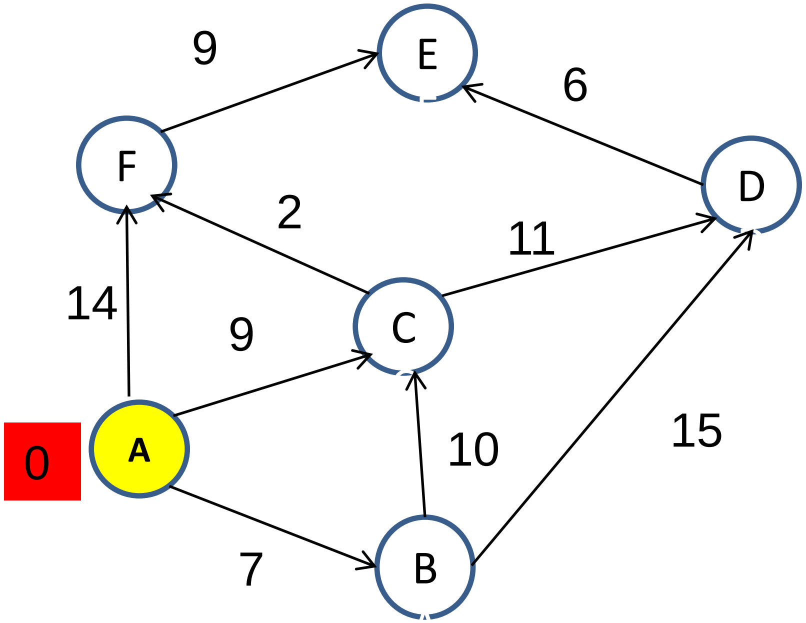
"D" ← EXTRACT-MIN(Q):



Q:	A	B	C	D	E
	0	∞	∞	∞	∞
		10	3	∞	∞
		7		11	5
		7		11	
				9	

S: { A, C, E, B, D }

EXAMPLE

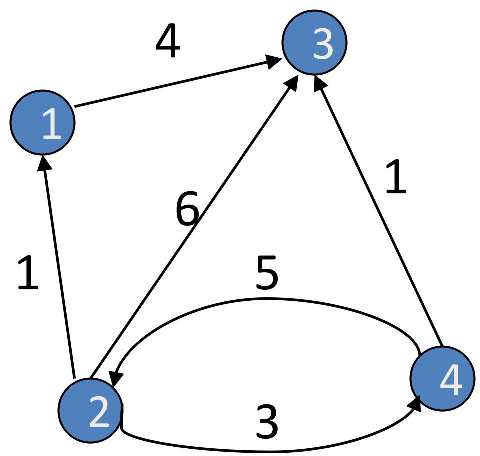


Floyd-Warshall's Algorithm: All pairs shortest paths

Problem: In a weighted (di)graph, find shortest paths between every pair of vertices

Same idea: construct solution through series of matrices $D^{(0)}$, …, $D^{(n)}$ using increasing subsets of the vertices allowed as intermediate

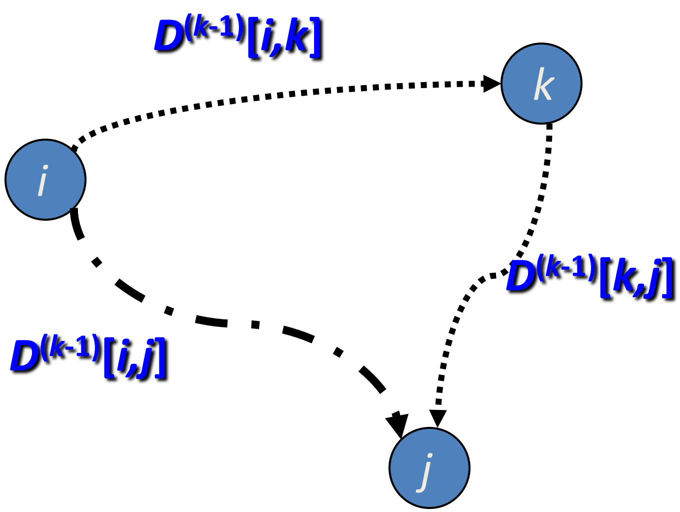
Example:



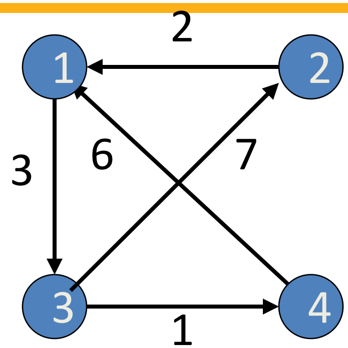
Floyd-Warshall's Algorithm (matrix generation)

On the *k*-th iteration, the algorithm determines shortest paths between every pair of vertices *i, j* that use only vertices among 1,…,*k* as intermediate

$$D^{(k)}[i,j] = \min \{D^{(k-1)}[i,j],\ D^{(k-1)}[i,k] + D^{(k-1)}[k,j]\}$$



Floyd-Warshall's Algorithm (example)



$$D^{(0)} = \begin{matrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \infty & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \infty & 0 \end{matrix}$$

$$D^{(1)} = \begin{matrix} 0 & \infty & 3 & \infty \\ 2 & 0 & \mathbf{5} & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & \mathbf{9} & 0 \end{matrix}$$

$$D^{(2)} = \begin{matrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ \mathbf{9} & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{matrix}$$

$$D^{(3)} = \begin{matrix} 0 & \mathbf{10} & 3 & \mathbf{4} \\ 2 & 0 & 5 & \mathbf{6} \\ 9 & 7 & 0 & 1 \\ 6 & \mathbf{16} & 9 & 0 \end{matrix}$$

$$D^{(4)} = \begin{matrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ \mathbf{7} & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{matrix}$$

Floyd-Warshall's(pseudocode and analysis)

ALGORITHM *Floyd*($W[1..n, 1..n]$)

//Implements Floyd's algorithm for the all-pairs shortest-paths problem
//Input: The weight matrix W of a graph with no negative-length cycle
//Output: The distance matrix of the shortest paths' lengths
$D \leftarrow W$ //is not necessary if W can be overwritten
for $k \leftarrow 1$ **to** n **do**
    for $i \leftarrow 1$ **to** n **do**
        for $j \leftarrow 1$ **to** n **do**
            $D[i, j] \leftarrow \min\{D[i, j], D[i, k] + D[k, j]\}$
return D

Time efficiency: Θ(n^3)

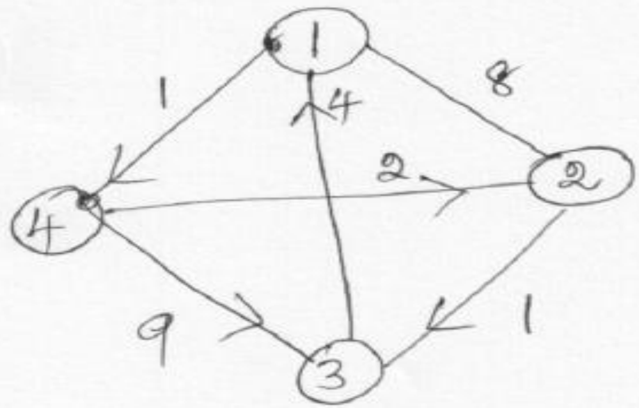
Space efficiency: Matrices can be written over their predecessors

Note: Shortest paths themselves can be found, too

Example-Floyd-Warshall Algorithm

innovate achieve lead

Example - 1



$$D^{(0)} = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 8 & \infty & 1 \\ 2 & \infty & 0 & 1 & \infty \\ 3 & 4 & \infty & 0 & \infty \\ 4 & \infty & 2 & 9 & 0 \end{array}$$

Example-Floyd-Warshall Algorithm

$$D^{(1)} = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 8 & \infty & 1 \\ \infty & 0 & 1 & \infty \\ 4 & 12 & 0 & 5 \\ \infty & 2 & 9 & 0 \end{bmatrix} \end{array}$$

$$D^{(2)} = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 8 & 9 & 1 \\ \infty & 0 & 1 & \infty \\ 4 & 12 & 0 & 5 \\ \infty & 2 & 3 & 0 \end{bmatrix} \end{array}$$

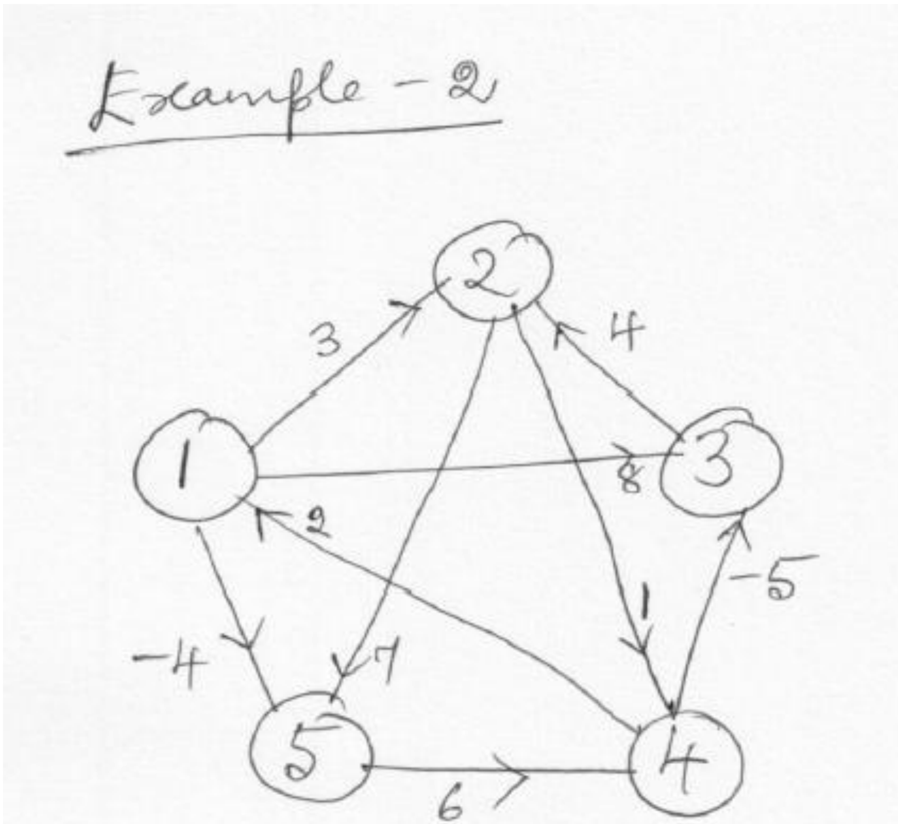
Example-Floyd-Warshall Algorithm

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

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

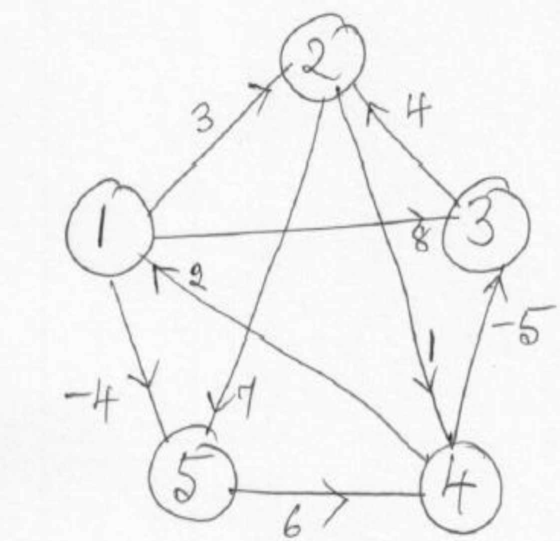
$\hookrightarrow$ final solution.

Example-Floyd-Warshall Algorithm



Example-Floyd-Warshall Algorithm

Example - 2



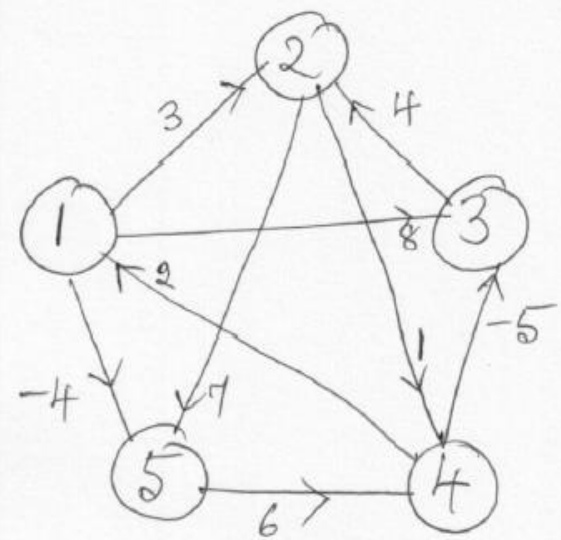
$$D^{(0)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \end{array}$$

$$D^{(1)} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \end{array}$$

Example-Floyd-Warshall Algorithm

$$D^{(2)} = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 3 & 8 & 4 & -4 \\ 2 & \infty & 0 & \infty & 1 & 7 \\ 3 & \infty & 4 & 0 & 5 & 11 \\ 4 & 2 & 5 & -5 & 0 & -2 \\ 5 & \infty & \infty & \infty & 6 & 0 \end{array}$$

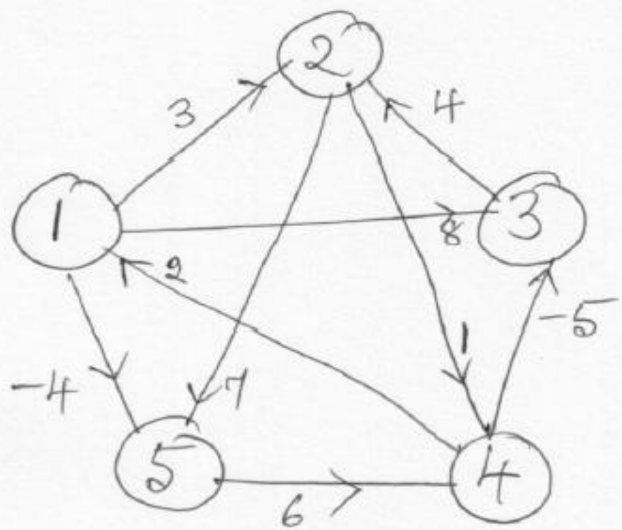
Example - 2



$$D^{(3)} = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 3 & 8 & 4 & -4 \\ 2 & \infty & 0 & \infty & 1 & 7 \\ 3 & \infty & 4 & 0 & 5 & 11 \\ 4 & 2 & -1 & -5 & 0 & -2 \\ 5 & \infty & \infty & \infty & 6 & 0 \end{array}$$

Example-Floyd-Warshall Algorithm

Example - 2



$$D^{(4)} = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 3 & -1 & 4 & -4 \\ 2 & 3 & 0 & -4 & 1 & -1 \\ 3 & 7 & 4 & 0 & 5 & 3 \\ 4 & 2 & -1 & -5 & 0 & -2 \\ 5 & 8 & 5 & 1 & 6 & 0 \end{array}$$

$$D^{(5)} = \begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 1 & -3 & 2 & -4 \\ 2 & 3 & 0 & -4 & 1 & -1 \\ 3 & 7 & 4 & 0 & 5 & 3 \\ 4 & 2 & -1 & -5 & 0 & -2 \\ 5 & 8 & 5 & 1 & 6 & 0 \end{array}$$

$\longrightarrow$ Final answer.