Data Structure and Algorithms

Session-29

Dr. Subhra Rani Patra
SCOPE, VIT Chennai

What we will learn in 'All Pair Shortest Path' ?

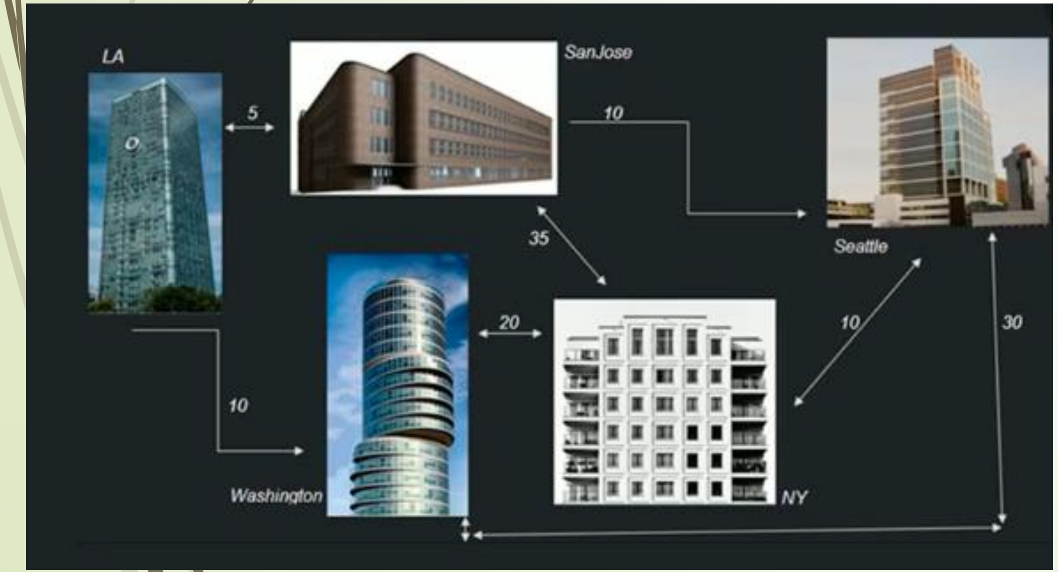
What is All Pair Shortest Path problem

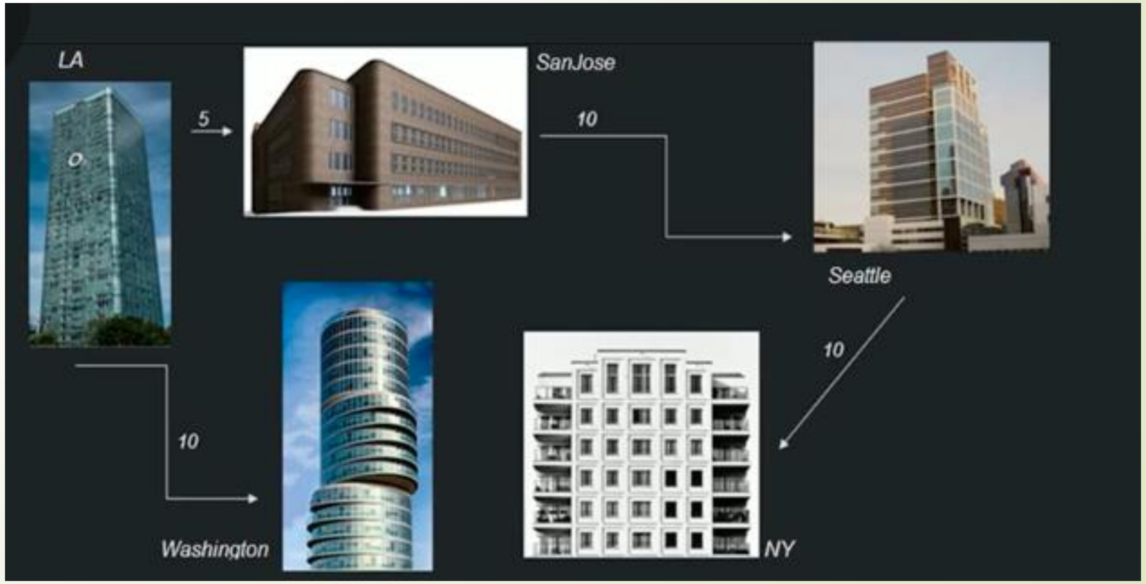
- ✓ *What is Floyd Warshall Algorithm*
- ✓ *How it works*
- ✓ *Why it works*
- ✓ *Why Negative Cycle does not works with it*

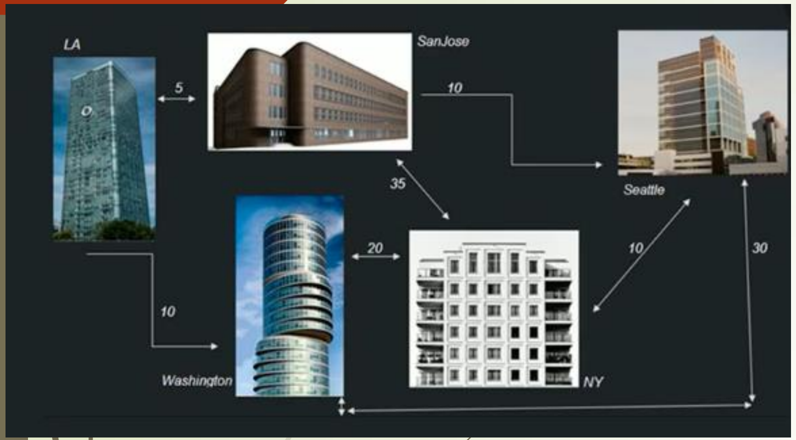
What is 'Single Source Shortest Path' ?

- Definition:
  - Single source shortest path problem is about finding a path between a given vertex (called 'Source') to all other Vertices in a graph such that, the total distance between them (source & Destination) is minimum.

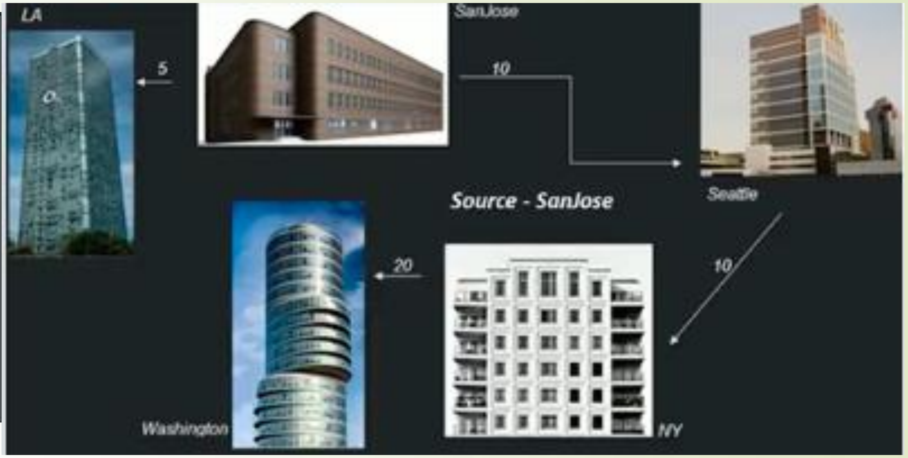
- Example
  - Let's say we have office in 5 different cities and we need to travel from Head office to all other offices.
  - Flight charges between cities are known (as given in below diagram).
  - What is the cheapest way to reach each office from HQ ?

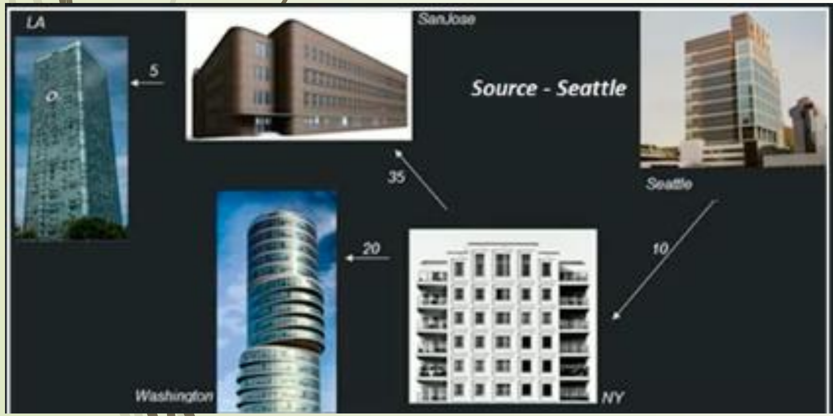





LA
SanJose
Seattle
Washington
NY
5
10
35
20
10
10
30


LA
SanJose
Seattle
Source - LA
Washington
NY
5
10
10
10

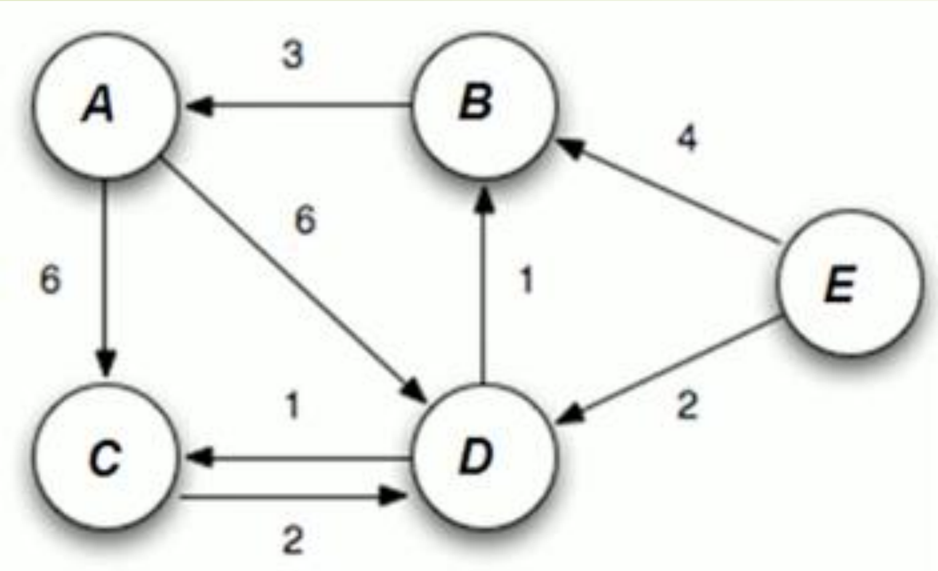

LA
SanJose
Seattle
Source - SanJose
Washington
NY
5
10
20
10

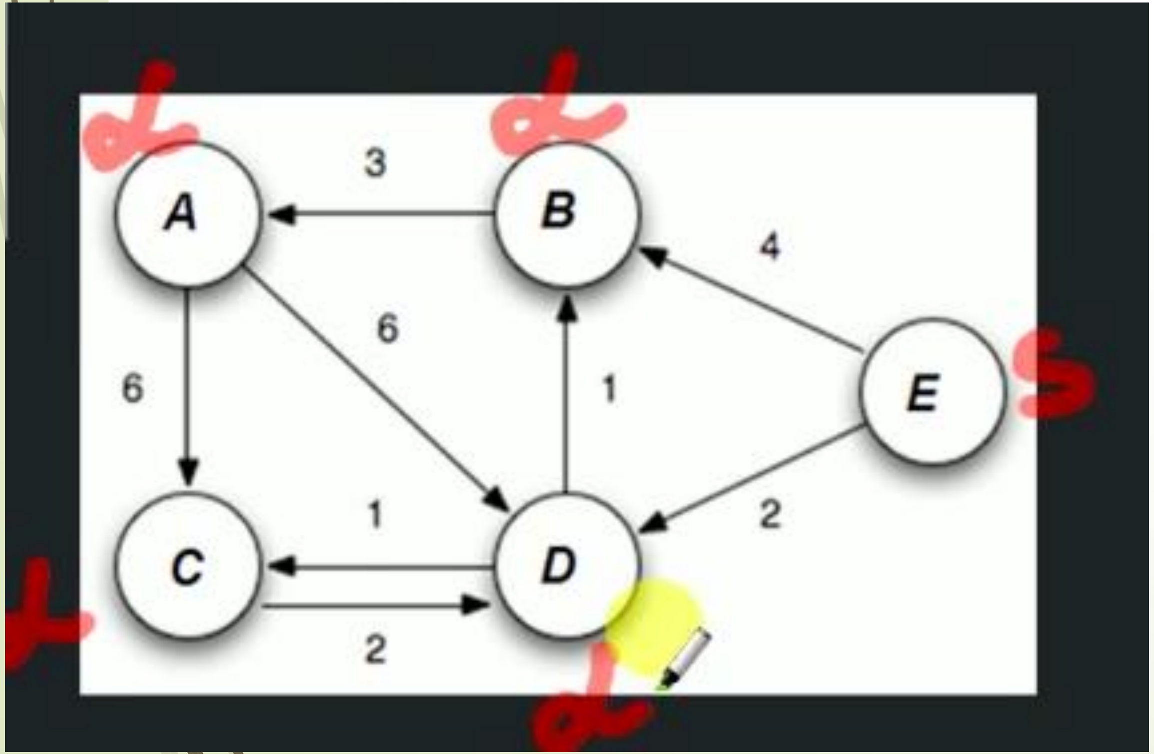

LA
SanJose
Source - Seattle
Seattle
Washington
NY
5
35
20
10

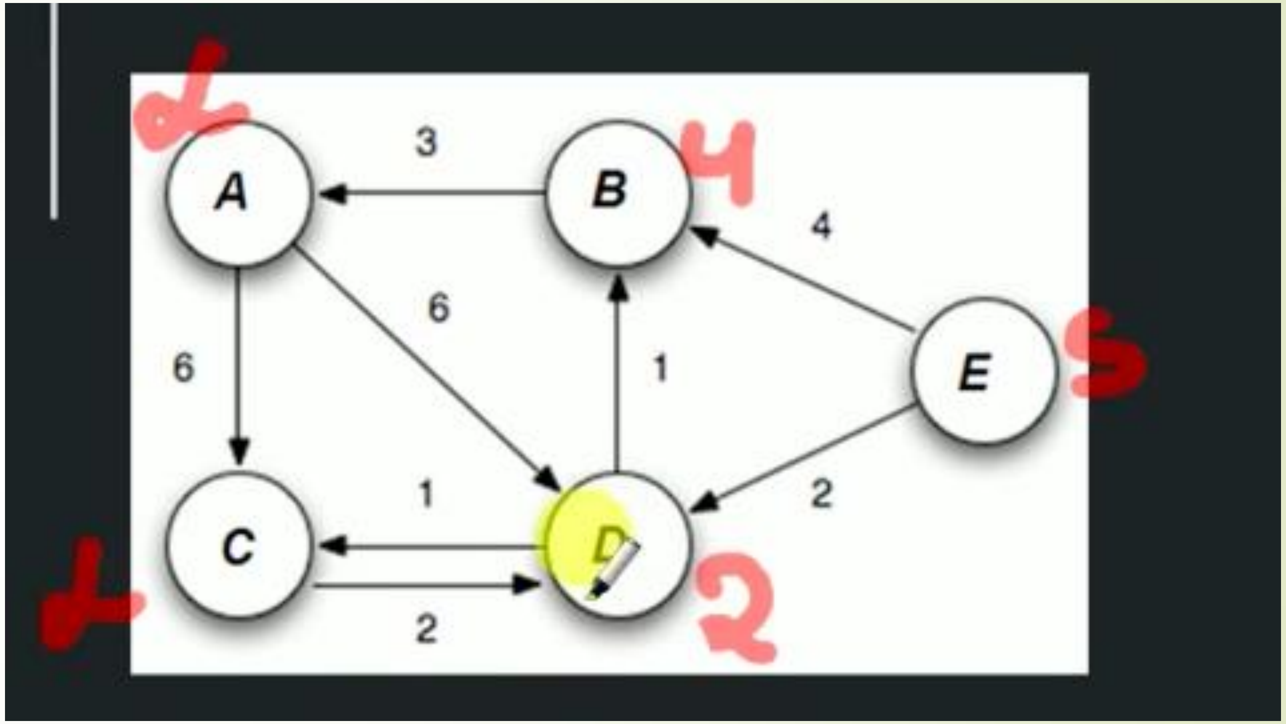

LA
SanJose
Seattle
Source - Washington
Washington
NY
5
35
20
10

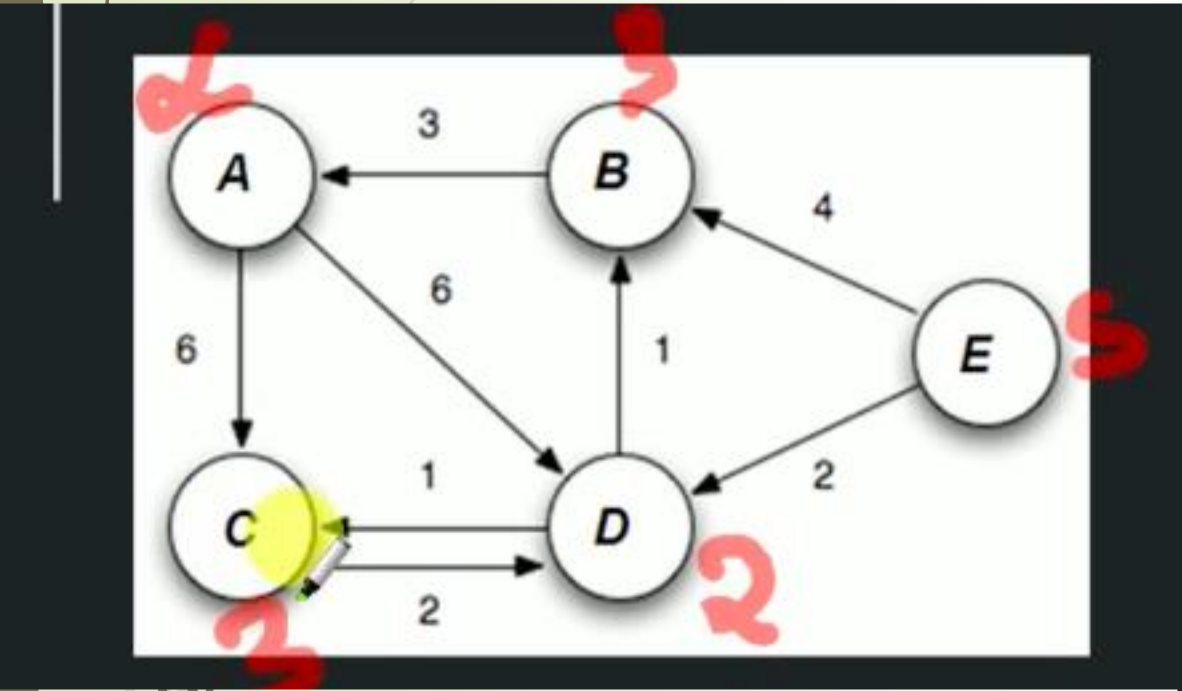

LA
SanJose
Source - NY
Seattle
Washington
NY
5
35
20
10

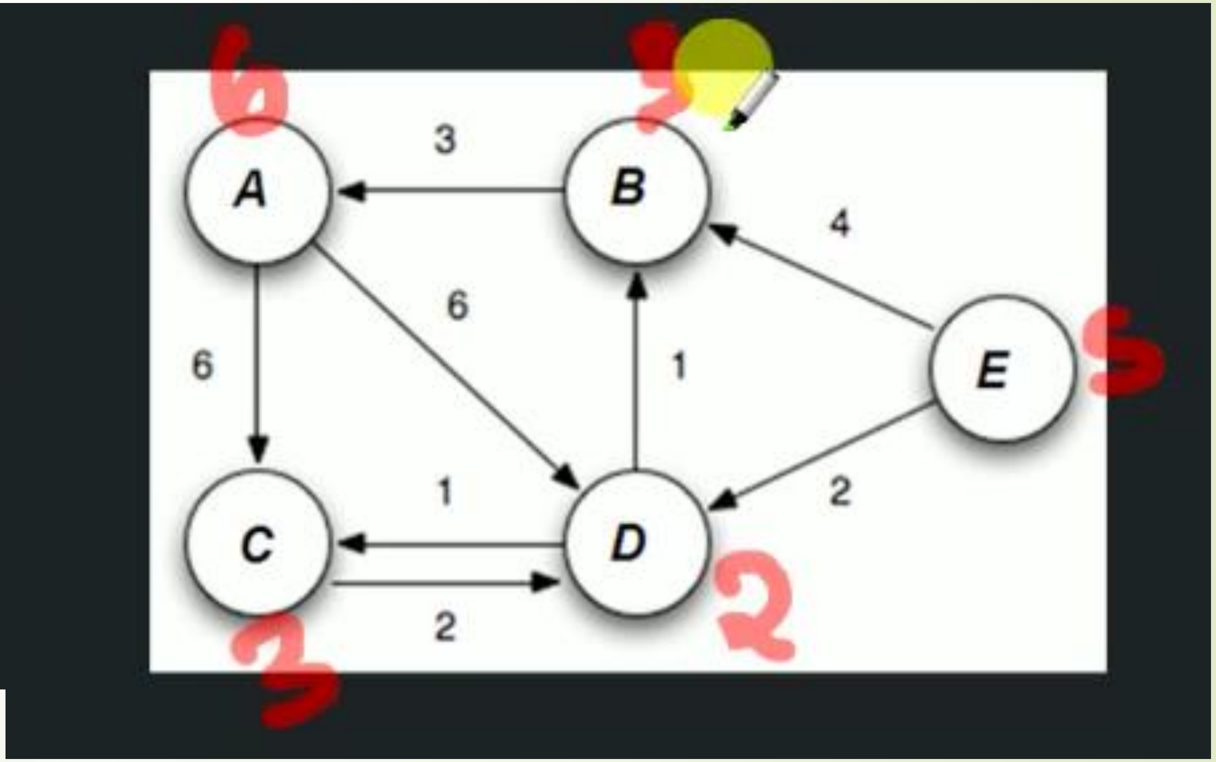
Dijkstra approach

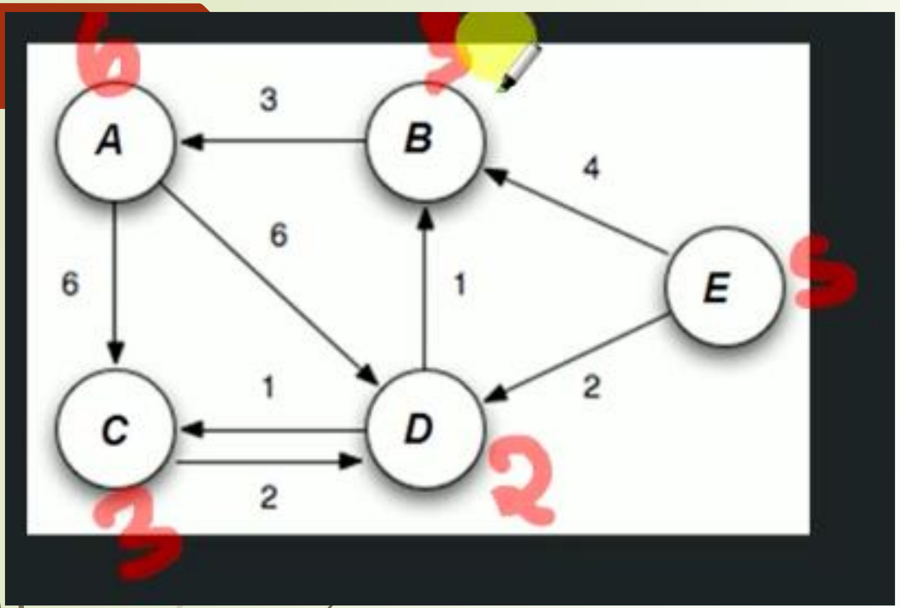







A
B
C
D
E
3
4
6
6
1
1
2
2


A
B
C
D
E
3
4
6
6
1
1
2
2



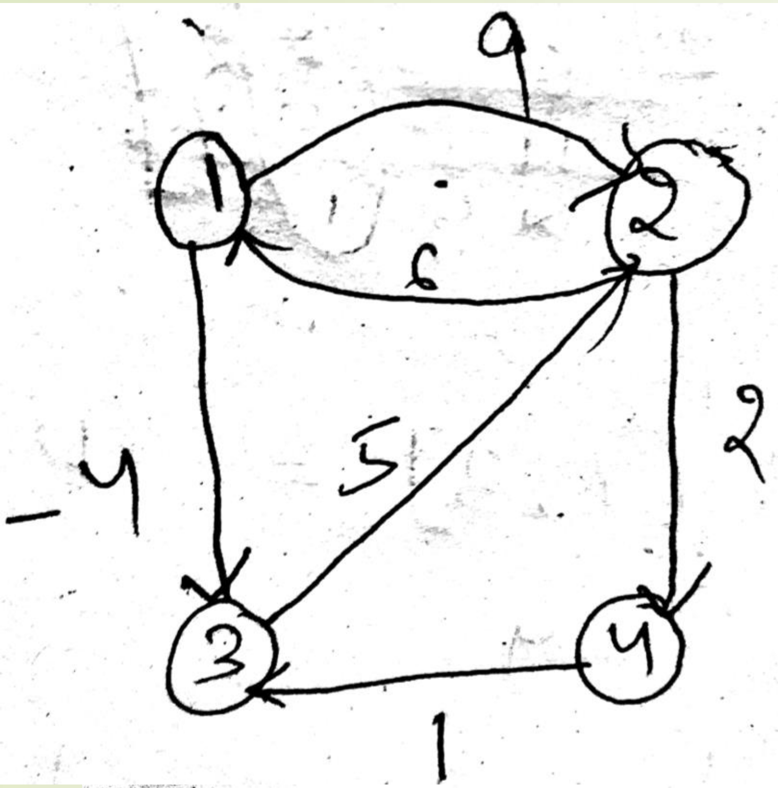
Source Vertex - 'E'	Path
A	E >D>B>A
B	E>D>B
C	E >D>C
D	E
E	-

Source Vertex - 'D'	Path
A	D>B>A
B	D>B
C	D>C
D	-
E	N/A

Source Vertex - 'C'	Path
A	C>D>B>A
B	C>D>B
C	-
D	C>D
E	N/A

Source Vertex - 'B'	Path
A	B>A
B	-
C	B>A>C
D	A>D
E	N/A

Source Vertex - 'A'	Path
A	-
B	A>D>B
C	A>D>C
D	A>D
E	N/A



$$D^0 = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 9 & -4 & \infty \\ 6 & 0 & \infty & 2 \\ \infty & 5 & 0 & \infty \\ \infty & \infty & 1 & 0 \end{bmatrix} \end{array}$$

$$D^0 = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 9 & -4 & \infty \\ 6 & 0 & \infty & 2 \\ \infty & 5 & 0 & \infty \\ \infty & \infty & 1 & 0 \end{bmatrix} \end{array}$$

$$D^1 = \begin{array}{cc} & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} & \begin{bmatrix} 0 & 9 & -4 & \infty \\ 6 & & & \\ \infty & & & \\ \infty & & & \end{bmatrix} \end{array}$$

$$D^1 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 9 & -4 & \infty \\ 2 & 6 & 0 & 2 & 2 \\ 3 & \infty & 5 & 0 & \infty \\ 4 & \infty & \infty & 1 & 0 \end{array}$$

$D^0[2,3]$ $\quad D^0[2,1] + D^0[1,3]$

∞ $\quad 6 + (-4) = 2$

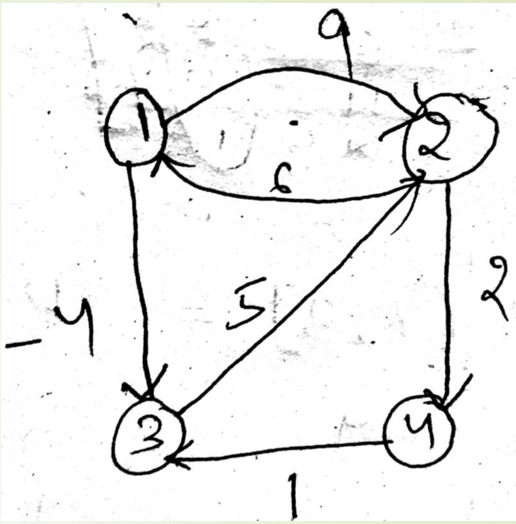
$D^0[2,4]$ $\quad D^0[2,1] + D^0[1,4]$

2 $\quad 6 + \infty = \infty$

$D^0[3,2]$ $\quad D^0[3,1] + D^0[1,2]$

5 $\quad \infty + 9 = \infty$

$$D^0 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 9 & -4 & \infty \\ 2 & 6 & 0 & \infty & 2 \\ 3 & \infty & 5 & 0 & \infty \\ 4 & \infty & \infty & 1 & 0 \end{array}$$



$$D^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 9 & & \\ 2 & 6 & 0 & 2 & 2 \\ 3 & & 5 & 0 & \\ 4 & & \infty & & 0 \end{array}$$

D_1 as base matrix

$$D^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 9 & -4 & \infty \\ 2 & 6 & 0 & 2 & 2 \\ 3 & 11 & 5 & 0 & \\ 4 & \infty & \infty & 1 & 0 \end{array}$$

D_1 as base matrix

$D^1[1,3] = D^1[1,2] + D^1[2,3]$

$-4 \quad 9 + 2$

$D^1[1,4] = D^1[1,2] + D^1[2,4]$

$\infty > 9 + 2 = 11$

$D^1[3,1] \quad D^1[3,2] + D^1[2,1]$

$\infty > 5 + 6 = 11$

$$D^2 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 9 & -4 & \infty \\ 2 & 6 & 0 & 2 & 2 \\ 3 & 11 & 5 & 0 & 7 \\ 4 & \infty & \infty & 1 & 0 \end{array}$$

$$D^3 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 1 & -4 & 3 \\ 2 & 6 & 0 & 2 & 2 \\ 3 & 11 & 5 & 0 & 7 \\ 4 & 12 & 6 & 1 & 0 \end{array}$$

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

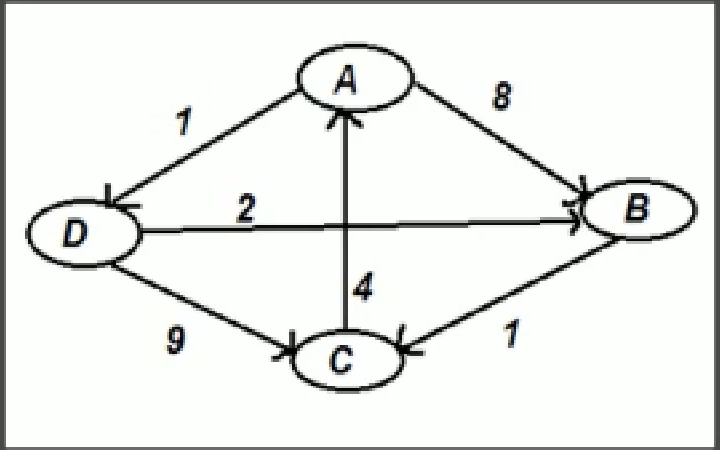
$i = 1 \quad k = 4$

$j = 2$

$$D^3[1,2] = D^3[1,4] + D^3[4,2]$$

$$1 < 3 + 6 = 9$$

$$D^4 = \begin{array}{cccc} 1 & 2 & 3 & 4 \\ \hline 0 & 1 & -4 & 3 \\ 6 & 0 & 2 & 2 \\ 11 & 5 & 0 & 7 \\ 12 & 6 & 1 & 0 \end{array}$$



For Each Edge

if D[u][v] > D[u]['via X'] + D['via X'][v]

Given				
	A	B	C	D
A	0	8	∞	1
B	∞	0	1	∞
C	4	∞	0	∞
D	∞	2	9	0

Iteration #1				
Via 'A'	A	B	C	D
A	0	8	∞	1
B	∞	0	1	∞
C	4	8+4 = 12	0	4+1 = 5
D	∞	2	9	0

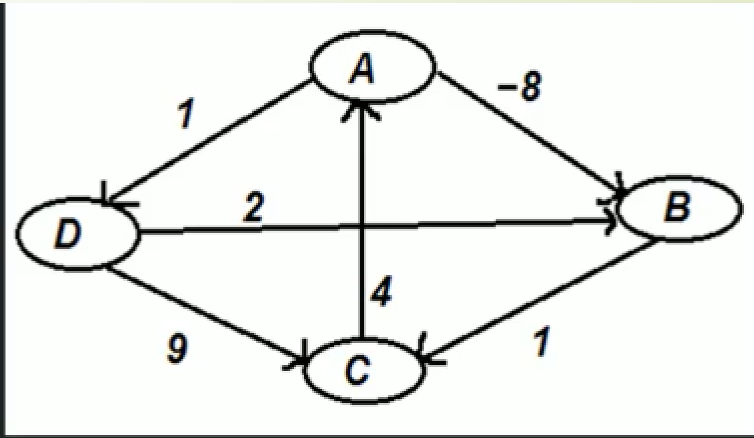
Iteration #2				
Via 'B'	A	B	C	D
A	0	8	8+1 = 9	1
B	∞	0	1	∞
C	4	12	0	5
D	∞	2	2+1 = 3	0

Iteration #3				
Via 'C'	A	B	C	D
A	0	8	9	1
B	4+1 = 5	0	1	5+1 = 6
C	4	12	0	5
D	4+3 = 7	2	3	0

Iteration #4				
Via 'D'	A	B	C	D
A	0	1+2 = 3	3+1 = 4	1
B	5	0	1	6
C	4	5+2 = 7	0	5
D	7	2	3	0

Final Result				
	A	B	C	D
A	0	3	4	1
B	5	0	1	6
C	4	7	0	5
D	7	2	3	0

Floyd Warshall Algorithm:




```
FloydWarshall (G)
  initialize a table of size VxV: D with ∞
  copy D from G
  for k = 0 to n-1 // run the loop as many time as number of vertices
     for i = 0 to n-1     //run the loop such that we visit each cell in 2D array in row wise fashion
        for j = 0 to n-1
           if D[i][j] > D[i][k] + D[k][j] then
              D[i][j] = D[i][k] + D[k][j]
  return D
```

Time & Space complexity of 'Floyd Warshall' Algorithm:

FloydWarshall (G)

initialize a table of size VxV: D with ∞ ------------------------------ $O(V^2)$

copy D from A -- $O(V^2)$

for k = 1 to n ------------------------------------ $O(V)$

for i = 1 to n --------------------------------- $O(V)$

for j = 1 to n ------------------------------ $O(V)$ ------------ $O(V^3)$

if D[i][j] > D[i][k] + D[k][j] then ------- $O(1)$

D[i][j] = D[i][k] + D[k][j] -------------- $O(1)$

return D -- $O(1)$

Time Complexity $= O(V^2) + O(V^2) + O(V^3) + O(1)$

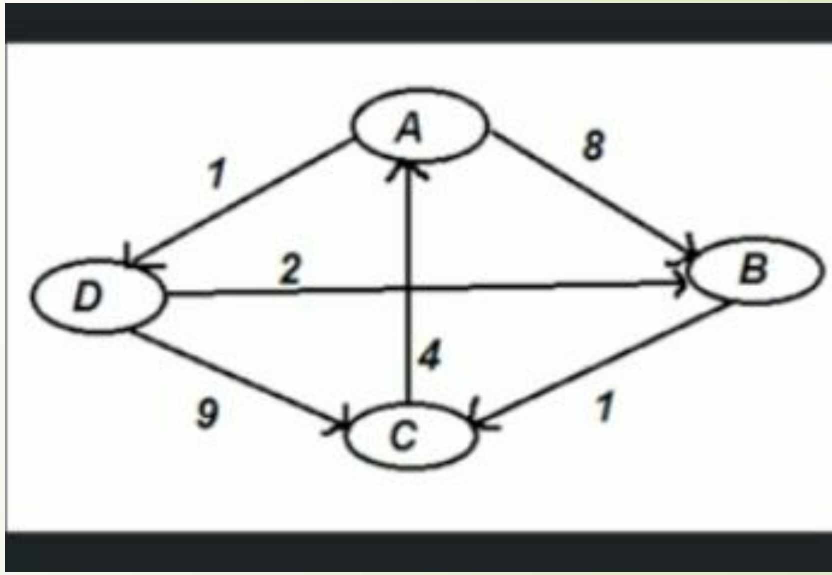
$= O(V^3)$

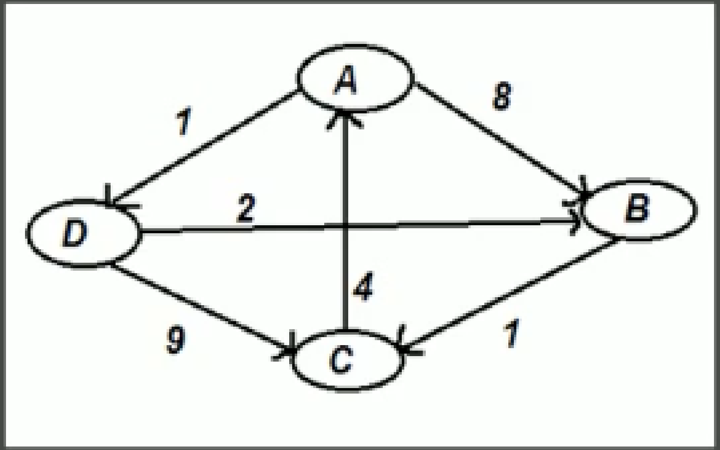
Space Complexity $= O(V^2)$

Why 'Floyd Warshall' Algorithm works ?

- ✓With any given 2 nodes(Source, destination) there can be only 3 probabilities to find distance between them:

- ✓ They are not reachable:
  - ✓ Not possible as it is given in the problem statement
- ✓ They are directly connected:
  - ✓ We keep this data initially in our 'Distance' matrix. We can have 2 more cases under it:
    - ✓ If this is best solution then we keep it for final answer (Ex-: A -> D)
    - ✓ If this can be improvised via some other vertex (Ex-: A -> B)
- ✓ They are reachable via some other node(s):
  - ✓ Nodes that are not reachable directly but are accessible via other nodes. (Ex: C > B)





For Each Edge

if D[u][v] > D[u]['via X'] + D['via X'][v]

Given	A	B	C	D
A	0	8	∞	1
B	∞	0	1	∞
C	4	∞	0	∞
D	∞	2	9	0

Iteration #1				
Via 'A'	A	B	C	D
A	0	8	∞	1
B	∞	0	1	∞
C	4	8+4 = 12	0	4+1 = 5
D	∞	2	9	0

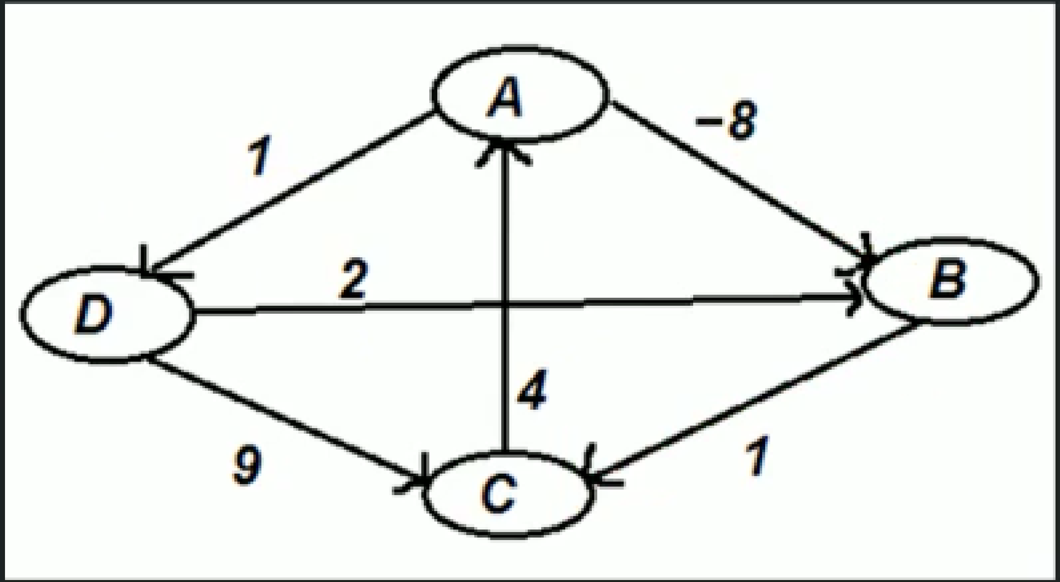
Iteration #2				
Via 'B'	A	B	C	D
A	0	8	8+1 = 9	1
B	∞	0	1	∞
C	4	12	0	5
D	∞	2	2+1 = 3	0

Iteration #3				
Via 'C'	A	B	C	D
A	0	8	9	1
B	4+1 = 5	0	1	5+1 = 6
C	4	12	0	5
D	4+3 = 7	2	3	0

Iteration #4				
Via 'D'	A	B	C	D
A	0	1+2 = 3	3+1 = 4	1
B	5	0	1	6
C	4	5+2 = 7	0	5
D	7	2	3	0

Final Result	A	B	C	D
A	0	3	4	1
B	5	0	1	6
C	4	7	0	5
D	7	2	3	0

Why ‘Negative Cycle’ does not work with ‘Floyd Warshall’ ?



✓We already know that, to go through a cycle, we need to go via ‘negative cycle participating vertex’ atleast twice.

✓We never run the loop twice ‘via same vertex’.


Thank you