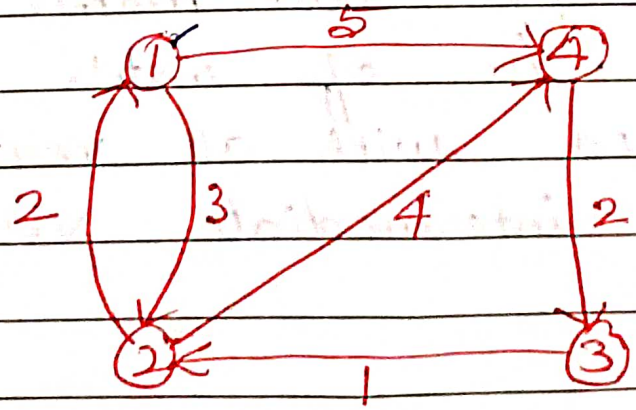
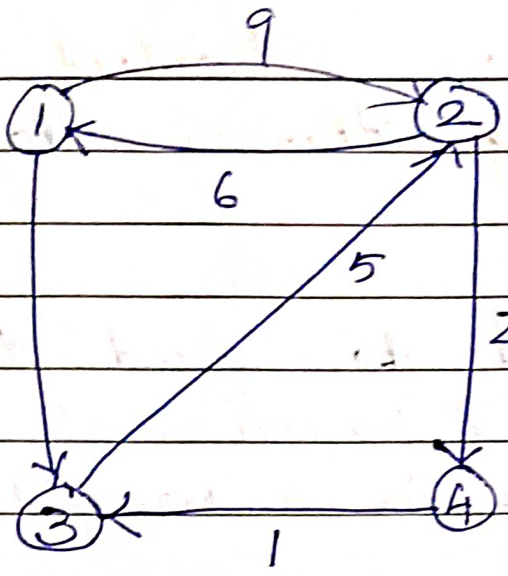
__/__/__

All-pairs shortest path algorithm - Floyd's Warshall algorithm

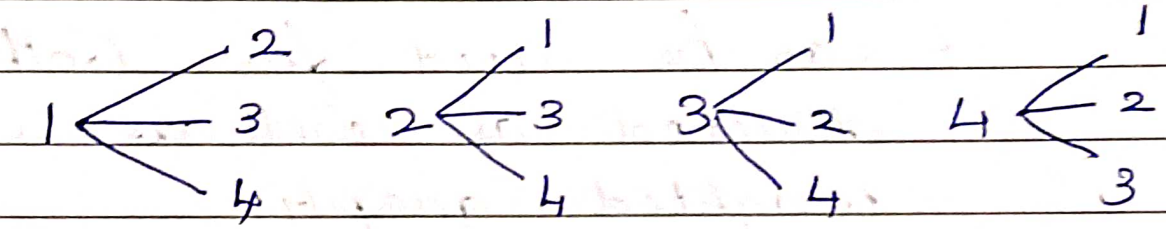
- used to find the shortest path between all pair of vertices in a weighted graph.
- can be used for both directed as well as undirected weighted graph
- it does not work for graph with negative cycles (when the sum of edges is negative).

Consider the following graph





All-pair →



The alg. generate a matrix which represent the minimum distance from any node to all other nodes in the graph.

Initially the o/p matrix is same as the give cost matrix of graph. Later, the o/p matrix is updated with all vertices k as the intermediate vertex.

__/__/__

Algorithm (Procedure)

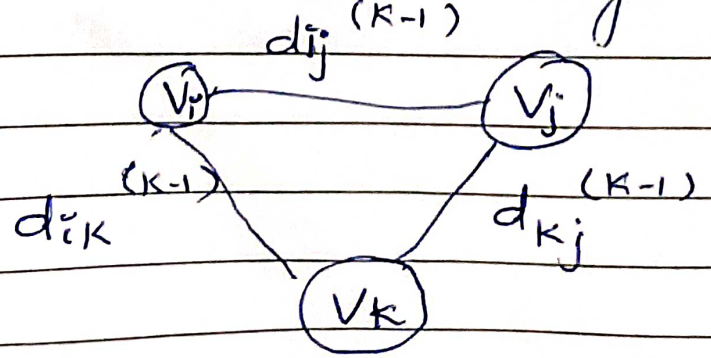
```
Begin
  for k=0 to n, do
    for i=0 to n, do
      for j=0 to n, do
        if cost[i,k] + cost[k,j]
        < cost[i,j], then
          cost[i,j] = cost[i,k] +
          cost[k,j]
        done for
      done for
    done for
  display the current cost
                matrix.
End.
```

__/__/__

- Floyd's algorithm computes the distance matrix of a weighted graph with 'n' vertices through a series of n-by-n matrices

$$D^{(0)} \cdots\cdots D^{(K-1)}, D^{(K)} \cdots D^{(n)}$$

- Compute all the elements of each matrix $D^{(K)}$ from its immediate predecessor $D^{(K-1)}$ in series.
- Let $d_{ij}^{(K)}$ be the element in the i^{th} row and j^{th} column of matrix $D^{(K)}$.
- d_{ij} is equal to the length of the shortest path among all paths from the i^{th} vertex V_i to all the j^{th} vertex V_j with their intermediate vertices numbered not higher than K_i



__/__/__

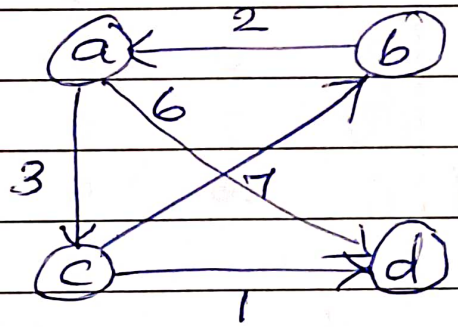
The formula for calculating $d_{ij}^{(k)}$

$$d_{ij}^{(k)} = \min \left\{ d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \right\}$$

for $k = 1$

$$d_{ij}^{(0)} = w_{ij}$$

Consider the following example.



$D^{(0)} =$

	a	b	c	d
a	0	∞	3	∞
b	2	0	∞	∞
c	∞	7	0	1
d	6	∞	∞	0

$D^{(1)}$ =

	a	b	c	d
a	0	∞	3	∞
b	2	0	(5)	∞
c	∞	7	0	1
d	6	∞	(9)	0

a – Known node
b – c through a
d – c through a

$D^{(2)}$ =

	a	b	c	d
a	0	∞	3	∞
b	2	0	5	∞
c	(9)	7	0	1
d	6	∞	9	0

a, b are known node
c – a through b

$D^{(3)}$ =

	a	b	c	d
a	0	(10)	3	(4)
b	2	0	5	(6)
c	9	7	0	1
d	6	(16)	9	0

a, b, c are known node

$D^{(4)}$ =

	a	b	c	d
a	0	10	3	4
b	2	0	5	6
c	(7)	7	0	1
d	6	16	9	0

a, b, c, d are known nodes