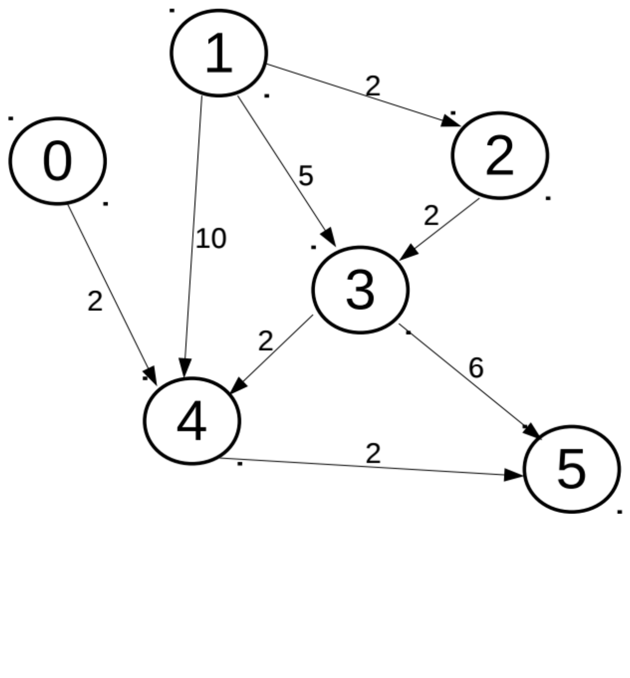
Workshop 10 – Week 11 – Worksheet 11



Question 11.1

1. Draw the matrix representation of the graph above.

	To										
From		0	1	2	3	4	5	6	7	8	9
	0	∞	∞	∞	∞	2	∞	∞	∞	∞	∞
	1	∞	∞	2	∞	10	∞	∞	∞	∞	∞
	2	∞	∞	∞	2	∞	∞	∞	∞	∞	∞
	3	∞	∞	∞	∞	2	6	∞	∞	∞	∞
	4	∞	∞	∞	∞	∞	2	∞	∞	∞	∞
	5	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	∞	5	9	∞
	7	∞	∞	∞	∞	∞	∞	∞	∞	6	∞
	8	∞	∞	∞	∞	∞	∞	∞	∞	∞	3
	9	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞

2. Trace the operation of the Floyd-Warshall algorithm on this graph, to get the shortest paths between all pairs of vertices.

i = 0	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	∞	2	∞	∞	∞	∞	∞
	1	∞	0	2	∞	10	∞	∞	∞	∞	∞
	2	∞	∞	0	2	∞	∞	∞	∞	∞	∞
	3	∞	∞	∞	0	2	6	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	2	∞	∞	∞	∞

	5	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	5	9	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	6	∞
	8	∞	∞	∞	∞	∞	∞	∞	∞	0	3
	9	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
i = 1	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	∞	2	∞	∞	∞	∞	∞
	1	∞	0	2	∞	10	∞	∞	∞	∞	∞
	2	∞	∞	0	2	∞	∞	∞	∞	∞	∞
	3	∞	∞	∞	0	2	6	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	2	∞	∞	∞	∞
	5	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	5	9	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	6	∞
	8	∞	∞	∞	∞	∞	∞	∞	∞	0	3
	9	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
i = 2	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	∞	2	∞	∞	∞	∞	∞
	1	∞	0	2	4	10	∞	∞	∞	∞	∞
	2	∞	∞	0	2	∞	∞	∞	∞	∞	∞
	3	∞	∞	∞	0	2	6	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	2	∞	∞	∞	∞
	5	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	5	9	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	6	∞
	8	∞	∞	∞	∞	∞	∞	∞	∞	0	3
	9	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
i = 3	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	∞	2	∞	∞	∞	∞	∞
	1	∞	0	2	4	6	10	∞	∞	∞	∞
	2	∞	∞	0	2	4	8	∞	∞	∞	∞
	3	∞	∞	∞	0	2	6	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	2	∞	∞	∞	∞
	5	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	5	9	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	6	∞
	8	∞	∞	∞	∞	∞	∞	∞	∞	0	3
	9	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
i = 4	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	∞	2	4	∞	∞	∞	∞
	1	∞	0	2	4	6	8	∞	∞	∞	∞
	2	∞	∞	0	2	4	6	∞	∞	∞	∞
	3	∞	∞	∞	0	2	4	∞	∞	∞	∞

	4	∞	∞	∞	∞	0	2	∞	∞	∞	∞
	5	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	5	9	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	6	∞
	8	∞	∞	∞	∞	∞	∞	∞	∞	0	3
	9	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
i = 5	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	∞	2	4	∞	∞	∞	∞
	1	∞	0	2	4	6	8	∞	∞	∞	∞
	2	∞	∞	0	2	4	6	∞	∞	∞	∞
	3	∞	∞	∞	0	2	4	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	2	∞	∞	∞	∞
	5	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	5	9	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	6	∞
	8	∞	∞	∞	∞	∞	∞	∞	∞	0	3
	9	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
i = 6	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	∞	2	4	∞	∞	∞	∞
	1	∞	0	2	4	6	8	∞	∞	∞	∞
	2	∞	∞	0	2	4	6	∞	∞	∞	∞
	3	∞	∞	∞	0	2	4	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	2	∞	∞	∞	∞
	5	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	5	9	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	6	∞
	8	∞	∞	∞	∞	∞	∞	∞	∞	0	3
	9	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
i = 7	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	∞	2	4	∞	∞	∞	∞
	1	∞	0	2	4	6	8	∞	∞	∞	∞
	2	∞	∞	0	2	4	6	∞	∞	∞	∞
	3	∞	∞	∞	0	2	4	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	2	∞	∞	∞	∞
	5	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	5	9	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	6	∞
	8	∞	∞	∞	∞	∞	∞	∞	∞	0	3
	9	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
i = 8	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	∞	2	4	∞	∞	∞	∞
	1	∞	0	2	4	6	8	∞	∞	∞	∞
	2	∞	∞	0	2	4	6	∞	∞	∞	∞

	3	∞	∞	∞	0	2	4	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	2	∞	∞	∞	∞
	5	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	5	9	12
	7	∞	∞	∞	∞	∞	∞	∞	0	6	9
	8	∞	∞	∞	∞	∞	∞	∞	∞	0	3
	9	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
i = 9	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	∞	2	4	∞	∞	∞	∞
	1	∞	0	2	4	6	8	∞	∞	∞	∞
	2	∞	∞	0	2	4	6	∞	∞	∞	∞
	3	∞	∞	∞	0	2	4	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	2	∞	∞	∞	∞
	5	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	5	9	12
	7	∞	∞	∞	∞	∞	∞	∞	0	6	9
	8	∞	∞	∞	∞	∞	∞	∞	∞	0	3
	9	∞	∞	∞	∞	∞	∞	∞	∞	∞	0

And the path matrix ends up being

	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0				4	4				
	1		1	2	2	2	2				
	2			2	3	3	3				
	3				3	4	4				
	4					4	5				
	5						5				
	6							6	7	8	8
	7								7	8	8
	8									8	9
	9										9

3. Does the order in which the vertices are examined affect the operation of the algorithm? Change the numbering of the vertices (but not the weights) as indicated below and retrace the algorithm. You should end up with the same result. Have any of the intermediate steps changed?

Old vertex number 3, rename to 5

Old vertex number 4, rename to 3

Old vertex number 5, rename to 4

Old vertex number 6, rename to 7

Old vertex number 7, rename to 9

Old vertex number 9, rename to 6

	To										
From		0	1	2	5	3	4	7	9	8	6
	0	0	∞	∞	∞	2	∞	∞	∞	∞	∞
	1	∞	0	2	∞	10	∞	∞	∞	∞	∞
	2	∞	∞	0	2	∞	∞	∞	∞	∞	∞
	5	∞	∞	∞	0	2	6	∞	∞	∞	∞
	3	∞	∞	∞	∞	0	2	∞	∞	∞	∞
	4	∞	∞	∞	∞	∞	0	∞	∞	∞	∞
	7	∞	∞	∞	∞	∞	∞	0	5	9	∞
	9	∞	∞	∞	∞	∞	∞	∞	0	6	∞
	8	∞	∞	∞	∞	∞	∞	∞	∞	0	3
	6	∞	∞	∞	∞	∞	∞	∞	∞	∞	0

…

	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	2	∞	∞	∞	∞	∞	∞
	1	∞	0	2	10	∞	∞	∞	∞	∞	∞
	2	∞	∞	0	∞	∞	2	∞	∞	∞	∞
	3	∞	∞	∞	0	2	∞	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	∞	∞	∞	∞	∞
	5	∞	∞	∞	2	6	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	9	5
	8	∞	∞	∞	∞	∞	∞	3	∞	0	∞
	9	∞	∞	∞	∞	∞	∞	∞	∞	6	0
i=0	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	2	∞	∞	∞	∞	∞	∞
	1	∞	0	2	10	∞	∞	∞	∞	∞	∞
	2	∞	∞	0	∞	∞	2	∞	∞	∞	∞
	3	∞	∞	∞	0	2	∞	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	∞	∞	∞	∞	∞
	5	∞	∞	∞	2	6	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	9	5
	8	∞	∞	∞	∞	∞	∞	3	∞	0	∞
	9	∞	∞	∞	∞	∞	∞	∞	∞	6	0
i=1	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	2	∞	∞	∞	∞	∞	∞
	1	∞	0	2	10	∞	∞	∞	∞	∞	∞
	2	∞	∞	0	∞	∞	2	∞	∞	∞	∞
	3	∞	∞	∞	0	2	∞	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	∞	∞	∞	∞	∞
	5	∞	∞	∞	2	6	0	∞	∞	∞	∞

	6	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	9	5
	8	∞	∞	∞	∞	∞	∞	3	∞	0	∞
	9	∞	∞	∞	∞	∞	∞	∞	∞	6	0
i=2	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	2	∞	∞	∞	∞	∞	∞
	1	∞	0	2	10	∞	4	∞	∞	∞	∞
	2	∞	∞	0	∞	∞	2	∞	∞	∞	∞
	3	∞	∞	∞	0	2	∞	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	∞	∞	∞	∞	∞
	5	∞	∞	∞	2	6	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	9	5
	8	∞	∞	∞	∞	∞	∞	3	∞	0	∞
	9	∞	∞	∞	∞	∞	∞	∞	∞	6	0
i=3	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	2	4	∞	∞	∞	∞	∞
	1	∞	0	2	10	12	4	∞	∞	∞	∞
	2	∞	∞	0	∞	∞	2	∞	∞	∞	∞
	3	∞	∞	∞	0	2	∞	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	∞	∞	∞	∞	∞
	5	∞	∞	∞	2	4	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	9	5
	8	∞	∞	∞	∞	∞	∞	3	∞	0	∞
	9	∞	∞	∞	∞	∞	∞	∞	∞	6	0
i=4	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	2	4	∞	∞	∞	∞	∞
	1	∞	0	2	10	12	4	∞	∞	∞	∞
	2	∞	∞	0	∞	∞	2	∞	∞	∞	∞
	3	∞	∞	∞	0	2	∞	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	∞	∞	∞	∞	∞
	5	∞	∞	∞	2	4	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	9	5
	8	∞	∞	∞	∞	∞	∞	3	∞	0	∞
	9	∞	∞	∞	∞	∞	∞	∞	∞	6	0
i=5	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	2	4	∞	∞	∞	∞	∞
	1	∞	0	2	6	8	4	∞	∞	∞	∞
	2	∞	∞	0	4	6	2	∞	∞	∞	∞

	3	∞	∞	∞	0	2	∞	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	∞	∞	∞	∞	∞
	5	∞	∞	∞	2	4	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	9	5
	8	∞	∞	∞	∞	∞	∞	3	∞	0	∞
	9	∞	∞	∞	∞	∞	∞	∞	∞	6	0
i=6	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	2	4	∞	∞	∞	∞	∞
	1	∞	0	2	6	8	4	∞	∞	∞	∞
	2	∞	∞	0	4	6	2	∞	∞	∞	∞
	3	∞	∞	∞	0	2	∞	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	∞	∞	∞	∞	∞
	5	∞	∞	∞	2	4	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	9	5
	8	∞	∞	∞	∞	∞	∞	3	∞	0	∞
	9	∞	∞	∞	∞	∞	∞	∞	∞	6	0
i=7	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	2	4	∞	∞	∞	∞	∞
	1	∞	0	2	6	8	4	∞	∞	∞	∞
	2	∞	∞	0	4	6	2	∞	∞	∞	∞
	3	∞	∞	∞	0	2	∞	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	∞	∞	∞	∞	∞
	5	∞	∞	∞	2	4	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
	7	∞	∞	∞	∞	∞	∞	∞	0	9	5
	8	∞	∞	∞	∞	∞	∞	3	∞	0	∞
	9	∞	∞	∞	∞	∞	∞	∞	∞	6	0
i=8	To										
From		0	1	2	3	4	5	6	7	8	9
	0	0	∞	∞	2	4	∞	∞	∞	∞	∞
	1	∞	0	2	6	8	4	∞	∞	∞	∞
	2	∞	∞	0	4	6	2	∞	∞	∞	∞
	3	∞	∞	∞	0	2	∞	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	∞	∞	∞	∞	∞
	5	∞	∞	∞	2	4	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
	7	∞	∞	∞	∞	∞	∞	12	0	9	5
	8	∞	∞	∞	∞	∞	∞	3	∞	0	∞
	9	∞	∞	∞	∞	∞	∞	9	∞	6	0
i=9	To										
From		0	1	2	3	4	5	6	7	8	9

	0	0	∞	∞	2	4	∞	∞	∞	∞	∞
	1	∞	0	2	6	8	4	∞	∞	∞	∞
	2	∞	∞	0	4	6	2	∞	∞	∞	∞
	3	∞	∞	∞	0	2	∞	∞	∞	∞	∞
	4	∞	∞	∞	∞	0	∞	∞	∞	∞	∞
	5	∞	∞	∞	2	4	0	∞	∞	∞	∞
	6	∞	∞	∞	∞	∞	∞	0	∞	∞	∞
	7	∞	∞	∞	∞	∞	∞	12	0	9	5
	8	∞	∞	∞	∞	∞	∞	3	∞	0	∞
	9	∞	∞	∞	∞	∞	∞	9	∞	6	0

The steps have changed but the result remains the same.

4. Does the Floyd-Warshall algorithm work on graphs where there are negative weights? Justify your answer.

No. When negative weights are present the Floyd-Warshall Algorithm could give incorrect results depending on how the nodes are ordered when negative cycles are present. If infinity is represented as a numeric value, negative values could also cause a fall below the value needed to signify no path and end up implying paths exist where they do not.

5. Given a sparse graph represented as an adjacency list, how would you approach the all pairs shortest paths problem? Does this differ from the approach you would take for a dense graph represented as a matrix? Compare the computational complexity of the two approaches.

Dijkstra’s Algorithm using the Adjacency List: $\Theta(|V|^2 \log|V| + |E||V|\log|V|)$.

Floyd-Warshall Algorithm converting the Adjacency List to an Adjacency Matrix: $\Theta(|V|^2 + |V|^3)$.

In a sparse graph, $|E| \in O(|V|)$, so in that case Dijkstra’s Algorithm would have complexity $\Theta(|V|^2 \log|V| + |V|^2 \log|V|) = \Theta(|V|^2 \log|V|)$ so would perform asymptotically better than the Floyd-Warshall Algorithm approach. In a dense graph, $|E| \in \Theta(|V|^2)$, so in that case Dijkstra’s Algorithm would have complexity $\Theta(|V|^2 \log|V| + |V|^3 \log|V|) = \Theta(|V|^3 \log|V|)$ so would perform asymptotically worse than the Floyd-Warshall Algorithm approach.