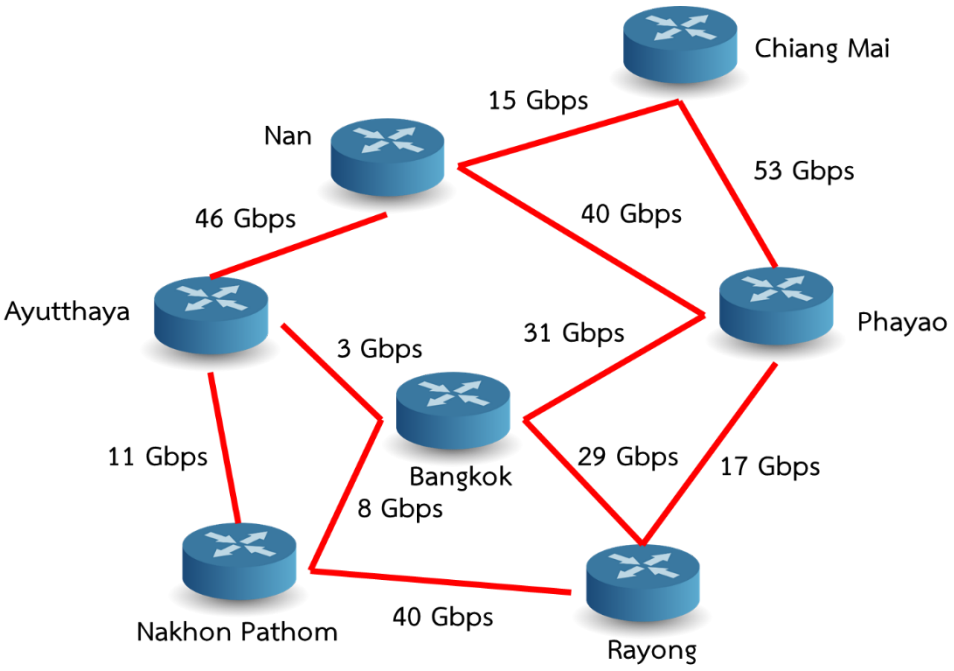
Assignment # 9: Graph Traversal and Shortest Path

1. Draw a graph corresponding to the data in the matrix below.

    a. The numbers indicate the weight of the edge between (*u*, *v*).

	BKK	**PHS**	**CNX**	**KKC**	**NST**
BKK	0	1	3	3	2
PHS	1	0	4	4	3
CNX	3	4	0	6	5
KKC	3	4	6	0	7
NST	2	3	5	7	0

2. Given a computer network below where a number represents the transfer **speed**, provide your answer to the following questions.



2.1 List all the nodes (v_1, v_2, …, v_n) in the network as a set of V.

V =

2.2 How many edges are there in the network?

2.3 List the adjacent nodes of Phayao.

2.4 Provides 3 cycle paths in the computer network.

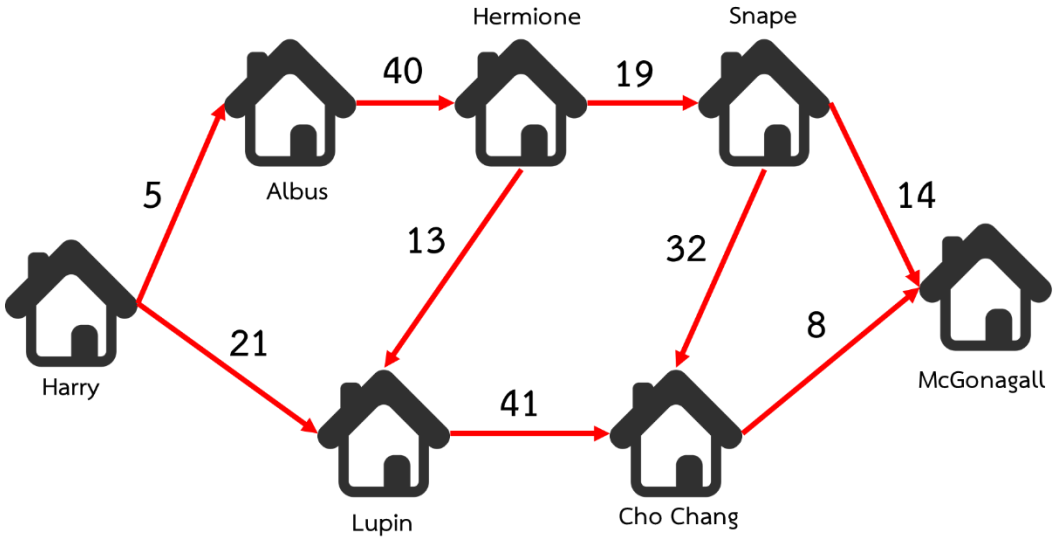
2.5 Create an adjacency matrix of the computer network where each cell in the matrix is the weight of the edge between (*u*, *v*).

	Ayutthaya	Bangkok	Chiang Mai	Nan	Nakhon Pathom	Phayao	Rayong
Ayutthaya							
Bangkok							
Chiang Mai							
Nan							
Nakhon Pathom							
Phayao							
Rayong							

2.6 Provide the most efficient traversal path using depth-first search, starting from Ayutthaya. The rule is to access a node with the highest speed first.

2.7 Provide the most efficient traversal path using breadth-first search, starting from Rayong. The rule is to access a node with the highest speed first.

3. Given the graph below where a number represents the traveling **distance**, provide your answer to the following questions.



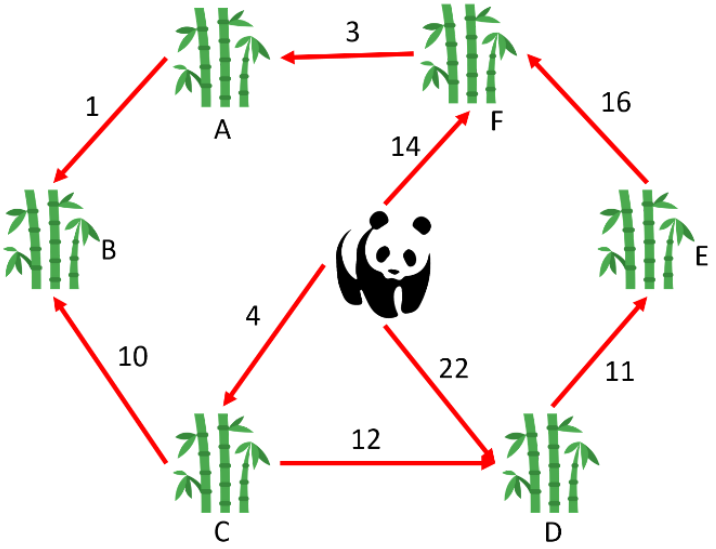
3.1 Write the shortest path (step-by-step) from node *Harry* and calculate the total weight with the distance between node *Harry* to the other nodes.

3.2 [Optional] Write the MST from node *Harry* (step-by-step) to other nodes using Prim's algorithm.

3.3 [Optional] Write the MST (step-by-step) to other nodes using Kruskal's algorithm.

3.4 [Optional] Why are the result MSTs from the Prim and Kruskal algorithm different from each other?

4. Given the graph below where the number represents the traveling **distance**, provide your answer to the following questions.



4.1 Write the shortest path (step-by-step) from node *Panda* and calculate the total weight with the distance between node *Panda* to the other nodes.

4.2 Provide the most efficient traversal path using depth-first search, starting from a panda's location. The rule is to go to the closest bamboo first.

4.3 Provide the most efficient traversal path using breadth-first search, starting from a panda's location. The rule is to go to the closest bamboo first.

4.4 [Optional] Write the MST from node *Panda* (step-by-step) to other nodes using Prim's algorithm.