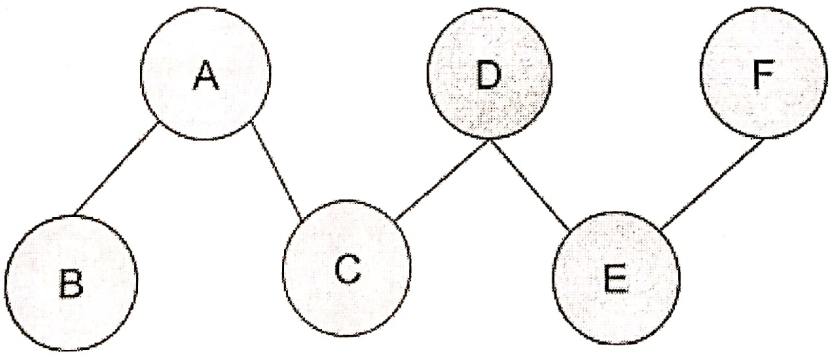
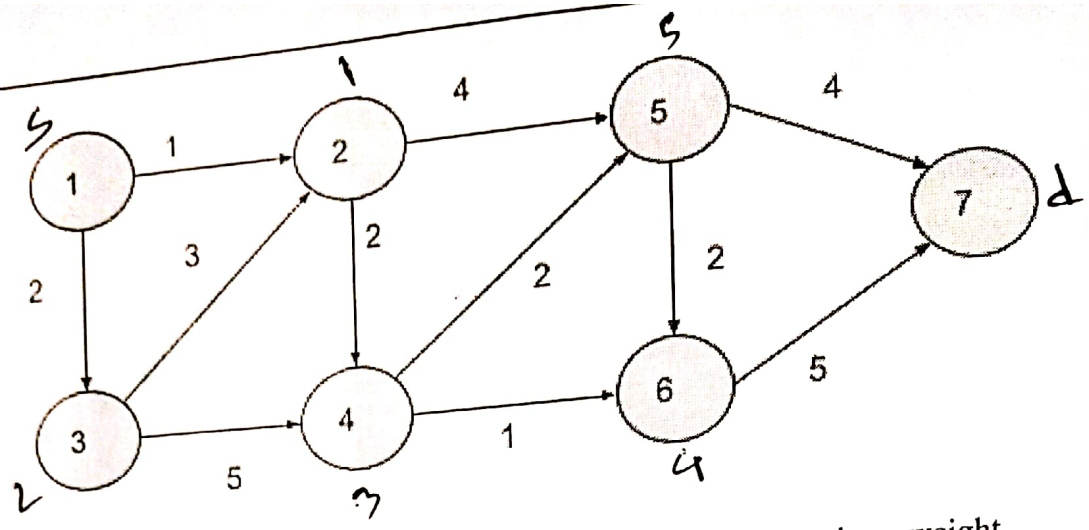
EAST WEST UNIVERSITY
Department of Computer Science and Engineering
B.Sc. in Computer Science and Engineering Program
Final Examination, Spring 2022 Semester

Course	**CSE 246 Algorithms, Section 04**
Instructor	**Redwan Ahmed Rizvee**
Full Marks	**30 (Will be converted to 20)**
Time	**1 hour 40 minutes**
Date	**May 19, 2022**

1		CO3
	Look at the following undirected graph,  **Design** a formal algorithm to find the articulation points or nodes from this graph. Also **apply** your designed algorithm over the given graph to find such points. An articulation point, is such a point that divides the given graph into two components if that node along with all the adjacent edges connected to it are removed from the graph.	4

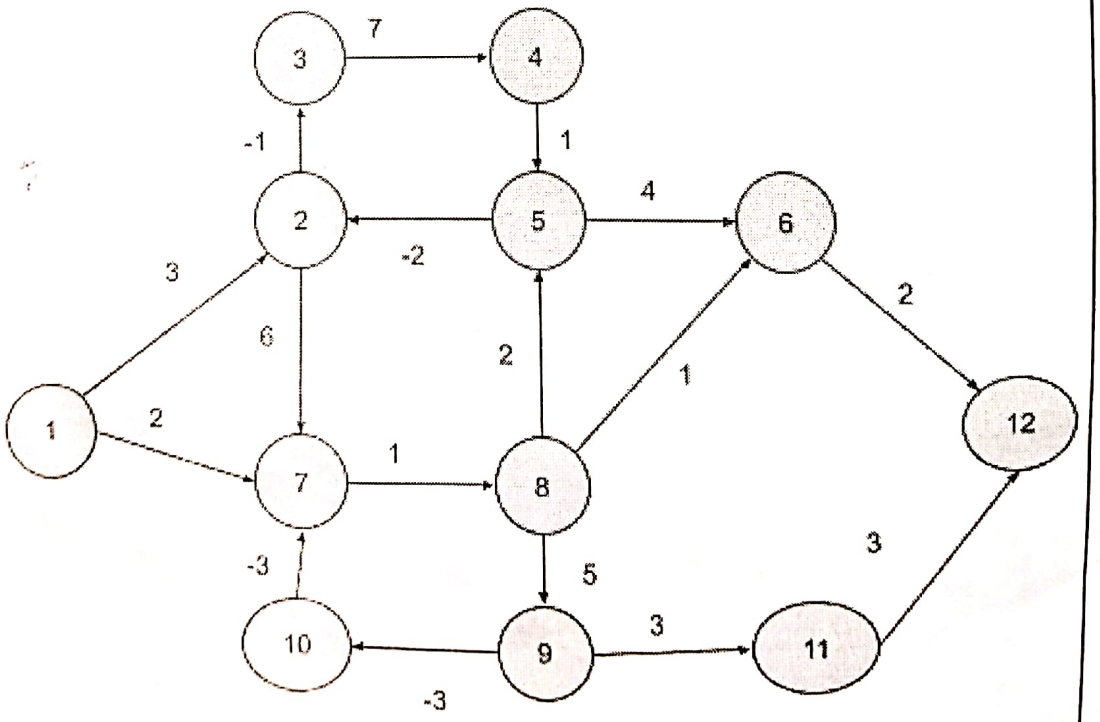
2		CO3



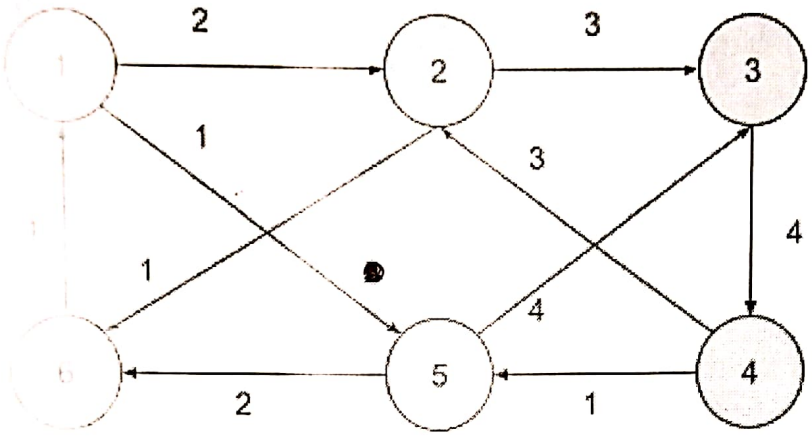
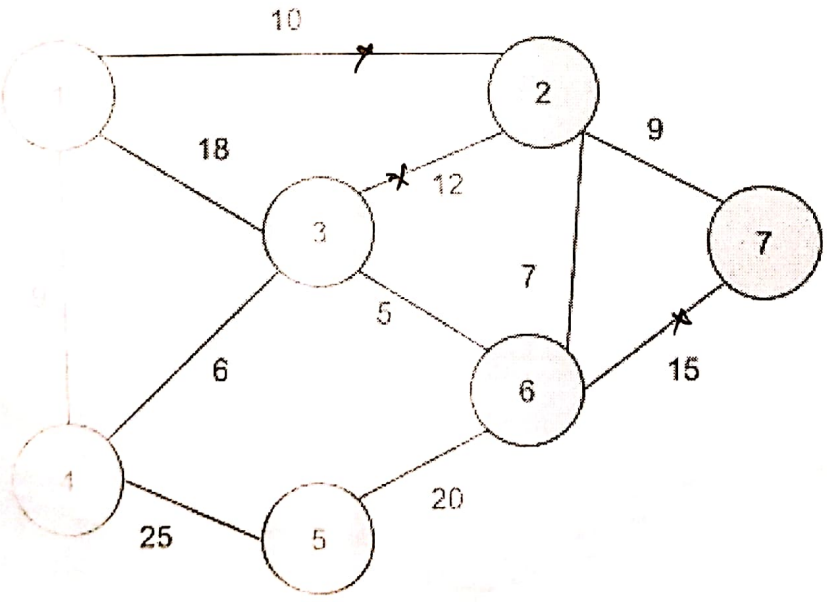
Look at the above directed graph. In the graph, for each edge there is an weight associated which denotes the distance to be convered to reach v from u (u -> v) using that edge. For the given problem, you will consider the node with id 1 and 7 as source and destinate node respectively.

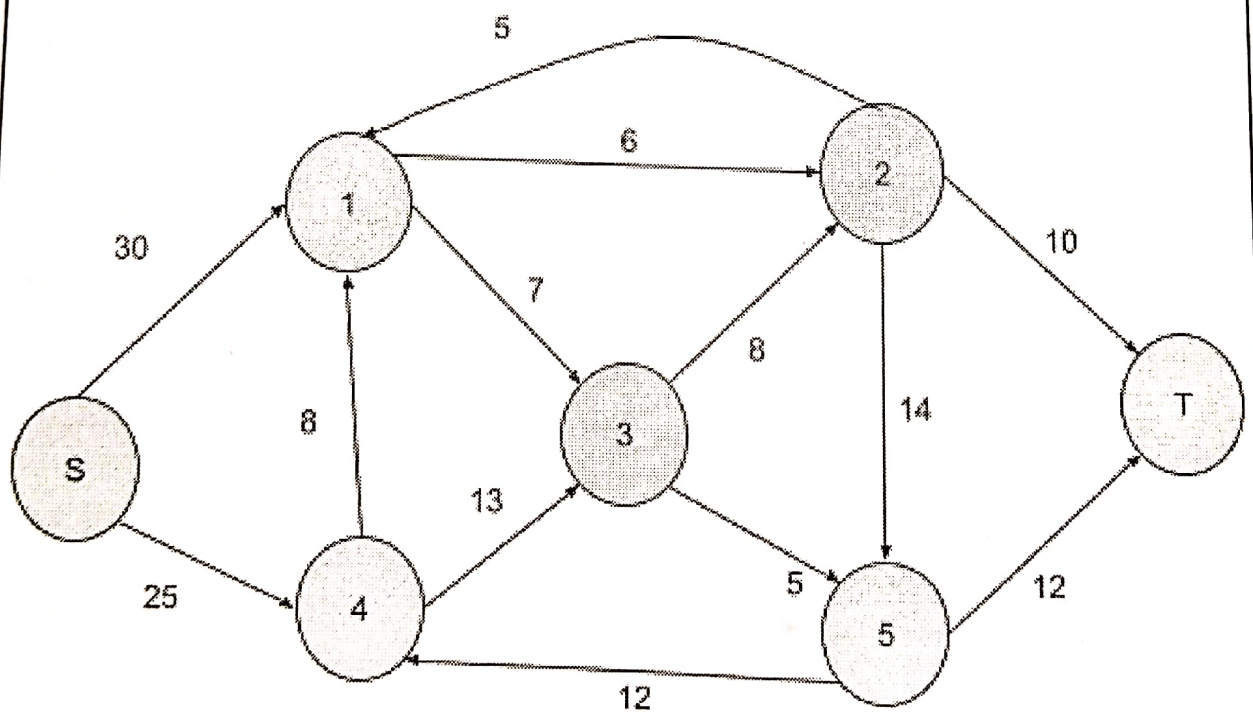
You need to **design** a shortest path algorithm to reach node 7 from node 1 using minimum weight/distance. **Apply** your designed algorithm to solve the problem for the given scenario. Also, **identify** if there lies multiple shortest paths to reach the destination from the source or not. If **exists**, report them.

3		CO3
	Look at the following directed weighted graph.	5



The graph contains negative weighted edges. For this problem, assume the source and destination nodes as 1 and 12 respectively. **Design** a formal algorithm to calculate the minimum distance to reach the destination node from the source node.

	Also apply the designed algorithm over the given graph.	
4		CO3
	 Look at the above given directed weighted graph. **Design** an efficient algorithm to calculate All pair shortest paths' distances. **Apply** the designed algorithm over the given graph and compute the results.	4
5		CO3
	Look at the following undirected graph,  **Design** an algorithm to calculate the Minimum Spanning Tree (MST) from an undirected weighted graph. **Apply** the algorithm over the given graph and show the resultant MST.	4

6			CO4
	Design an algorithm to detect if there exists negative cycle(s) from a given weighted directed graph where the weight of an edge denotes the distance between adjacent nodes. Also write an algorithm to **identify** the nodes that are affected due to the presence of negative cycles.		2
	Design an algorithm to order the edges of a directed weighted graph to run Bellman-Ford Algorithm so that the edges closest to the source are ordered first, then the edges closest to them and so on.		2
7			CO3
	 Look at the above given directed weighted graph where the edge weight denotes the capacity of the maximum flow that can be passed through that edge. **Design** an algorithm to calculate the maximum flow possible from a given network or graph. Also **apply** the algorithm over the given graph to calculate the resultant maximum flow from this network.		4

EAST WEST UNIVERSITY
Department of Computer Science and Engineering
B.Sc. in Computer Science and Engineering Program
Mid Term II Examination, Spring 2022 Semester

Course	**CSE 246 Algorithms, Section 04**
Instructor	**Redwan Ahmed Rizvee**
Full Marks	**30 (Will be converted to 20)**
Time	**1 hour 20 minutes**
Date	**April 07, 2022**

1		CO4
	Here you are given the already calculated dp table **DP** of the 0/1 knapsack problem. You are also given the Knapsack weight **W** and the corresponding items' information **Inf** , e.g, weight and profit. From the given information, you need to **Identify** and **justify** which items will give the maximum benefit *not violating the knapsack weight constraint.*	4

DP Table (Row: Weight, Column: Items)

i/w	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	21	21	21	21	21	21	21
2	0	21	21	31	31	31	31	31
3	0	21	21	36	36	46	46	46
4	0	21	31	36	46	46	56	56
5	0	21	31	36	46	46	56	61
6	0	21	31	36	46	46	56	61

W = 7, ***Inf*** Table

Item	Weight	Value
1	1	21
2	2	10
3	2	15
4	1	10
5	3	15
6	2	10

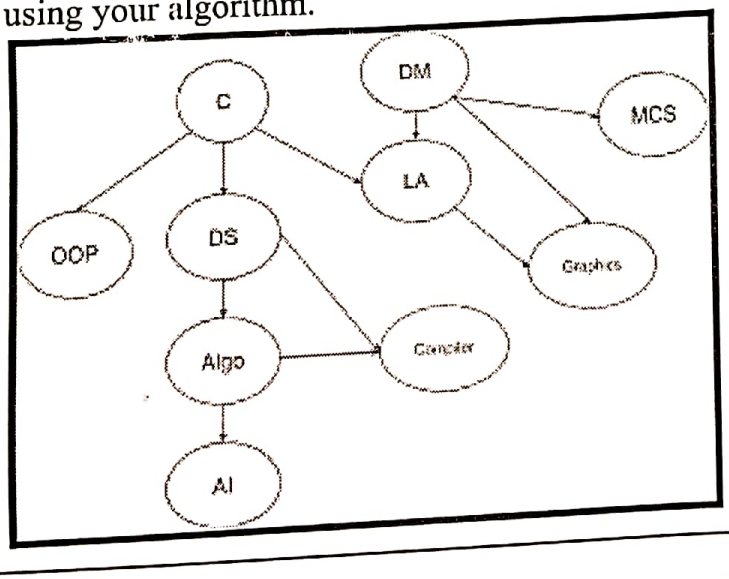
2			CO3
	You will be given two integers **a** and **b**. **Design** an algorithm to calculate the Greatest Common Divisor (GCD) of the given numbers.		2

3			CO3
	You will be given an integer number **N**. **Design** an algorithm to find if the given number is prime or not. No need to write the complete code, rather explain the main idea.		2

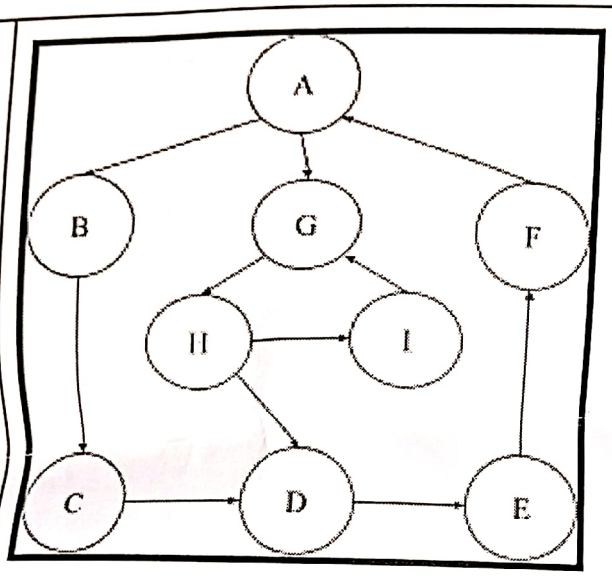
4			CO4
	Here, you are given a text **T** = 'abbbaabbcfabacaf' and a pattern P = 'bba'. You are going to simulate the string matching algorithm Robin-Karp here. For hashing, here the base **B** will be 2 (B=2) and modular value **M** will be 7 (M=7). In the algorithm modular value is used to reduce the case of numeric overflow. For ease, how a hashing is conducted is exemplified below for a pattern **P** of length **m**. $Hash(P) = (B^{(m-1)} P[0] + B^{(m-2)} P[1] + B^{(m-3)} P[2] + ... + P[m-1])\%M$ In this problem, you need to simulate if there lies any position(s) in **T** that might cause problems while finding the occurrences of **P** in **T**. **Determine** the possibility and **justify** the reasoning behind it. You also need to **develop** strategies to solve this challenge.		4+1

5			CO4
	In this problem, you will be dealing with EWU's classrooms and stairs.		4

	You will be given a graph **G** where nodes represent either a classroom or a stair. Edges represent the connectivity between nodes. If there is an edge between room **A** to room **B**, it means there is a direct path to move from **A** to **B** or **B** to **A** and they are adjacent. Nodes may also represent stairs. So an edge between stairs C to room D means, you can directly move to stairs from this room and vice versa, stairs to directly this room. Now, given such a graph **G** along with the edges and information about which nodes are rooms and which node (always a single node) is stairs and your current position. You need to **design** a shortest path algorithm to reach stairs from your current position/node. In this problem, it is regarded that all the edges are of the same distance. So, if there is an edge between A to B and C to D, then Distance(A,B) = Distance(C,D) = Distance(B,A) = Distance(D,C), etc.	

6		CO4
	In this problem, you are given a directed acyclic graph (DAG). Here each node represents a subject and the directed edges represent prerequisites of taking a course. So an edge from A to B (A->B) means that, to take course B, you need to take course A first. So, given such a graph consisting of many partial orders, you need to **design** an algorithm to find the total ordering of taking the subjects, so that, before taking a subject all of its prerequisites are taken. **Simulate** the given graph using your algorithm. 	4

7		CO3
	In this problem, you are given the following directed acyclic graph (DAG). You need to **design** an algorithm to find the Strongly Connected Components (SCC) from this graph. You need to **simulate** this input using your algorithm.	4



8		CO3
	Given an undirected graph, **design** an algorithm to detect if there is any cycle in the graph. Also **add** a logical block in your design to print all the found cycles.	3+2