CSE 3131: ALGORITHM DESIGN 1

ASSIGNMENT 3:

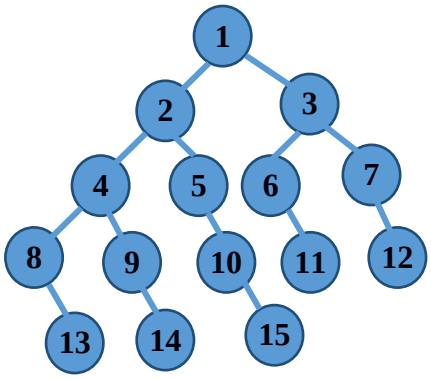
Submission due date: 14/12/2022

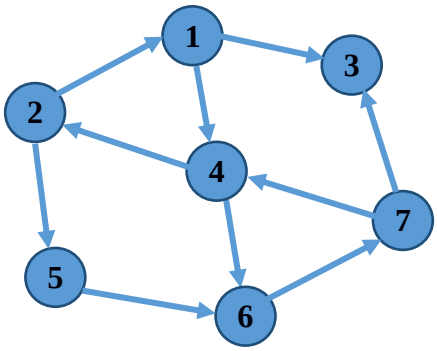
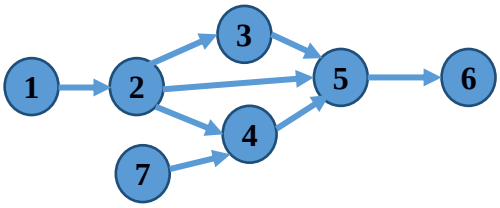
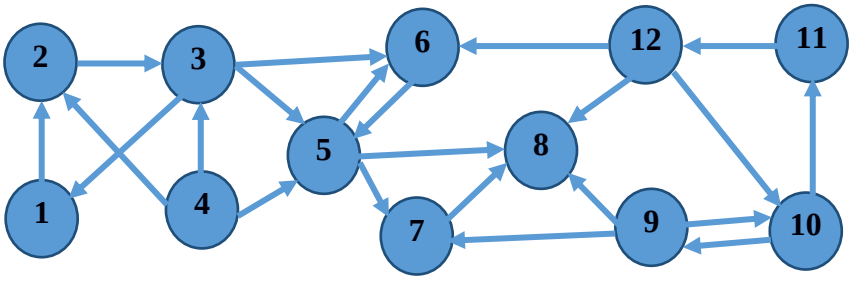
- *Assignment scores/markings depend on neatness and clarity.*
- *Write your answers with enough detail about your approach and concepts used, so that the grader will be able to understand it easily. You should ALWAYS prove the correctness of your algorithms either directly or by referring to a proof in the book.*
- *The marking would be out of 100.*
- *You are allowed to use only those concepts which are covered in the lecture class till date.*
- *Plagiarized assignments will be given a zero mark.*

---

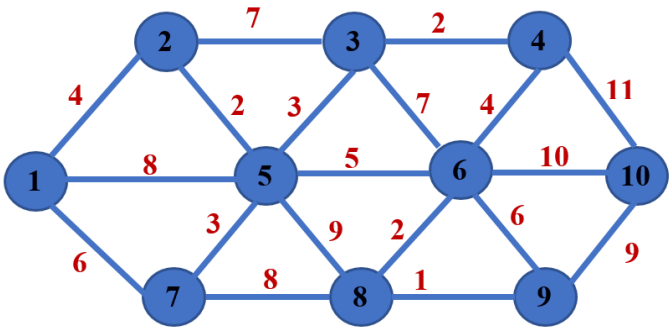
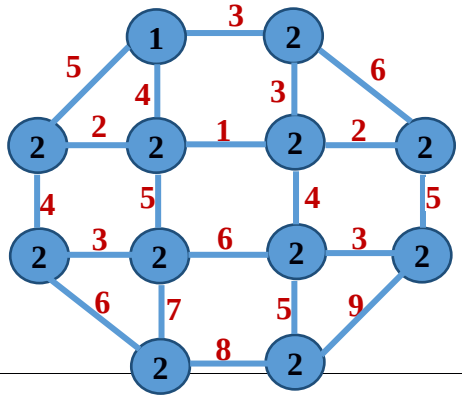
*CO2: to understand various types and aspects of **Graph algorithms**.*

Graph algorithms:

Sl.No.	***Question***	***PO***	***Level***
1.	Given a graph *G* in adjacency matrix representation. Write a pseudocode to find the adjacency list representation. Discuss its time complexity.	PO1, PO2	L2, L3
2.	Given a graph *G* in adjacency list representation, Write a pseudocode to find the adjacency matrix representation. Discuss its time complexity.	PO1, PO2	L2, L3
3.	Given a graph *G* in which each vertex is marked with a number as its index. Assume that if there is ever a choice amongst multiple nodes, both the BFS and DFS algorithms will choose the node with lesser index first. Apply BFS and DFS on *G*, starting from the *node 1*. Show the steps in detail.  i. Write the sequence in which the nodes will be visited in the complete traversal of *G* using BFS and DFS respectively. ii. What is the maximum size of the Queue in BFS and that of the Stack in DFS in full traversal of *G*? iii. Which of the two algorithms is preferred if the node to be searched is the *node 6*? iv. Which of the two algorithms is preferred if the node to be searched is the *node 14*?	PO1, PO2	L2, L3, L4

4.	Given a graph G in which each vertex is marked with a number as its index. Assume that if there is ever a choice amongst multiple nodes, the traversal algorithms will choose the node with lesser index first.  i. Find the Depth first tree. ii. Identify the edges of the graph as tree edge, back edge, forward edge and cross edge. iii. Find the parenthesis structure of the complete traversal. For example, If a node v is traversed from a node u in the DFS, we have a parenthesis structure like $(u(v\ \ v)u)$. iv. Write the nodes of the graph in topologically sorted order.	PO1, PO2	L2, L3, L4
5.	Prove that the minimum weight edge in a graph G with no duplicate edge weights must be present in every Minimum Spanning Tree of G.	PO1, PO2	L2, L3
6.	Prove that a directed graph G is a DAG if and only if the DFS traversal of G has no back edge. Perform the DFS traversal on the graph given below to identify the types of edges and check if it is a DAG or not.  	PO1, PO2	L2, L3, L4
7.	Let the nodes of a graph G are present in a set A in topologically sorted order. Prove that the DFS traversal of the transpose graph G^T, where in the main loop of DFS, we select the nodes in the topologically sorted order as present in A, will produce a depth first forest where each tree represents a strongly connected component of G. Find the connected components of the given graph using DFS.  	PO1, PO2	L2, L3, L4

8.	The adjacency matrix of an undirected graph G is given below. Find the connected components of the graph using UNION-FIND data structure. Show each step of the operation using the forest representation for the disjoint sets. Use union by rank heuristic for UNION() operation and path compression for FIND-SET() operation. Draw the forest for the UNION-FIND in each step and show the operations on the trees. **1** 0 1 1 0 0 0 0 0 0 **2** 1 0 1 0 0 0 0 0 0 **3** 1 1 0 0 0 0 0 0 0 **4** 0 0 0 0 1 0 0 0 0 **5** 0 0 0 1 0 0 0 0 0 **6** 0 0 0 0 0 0 0 1 0 **7** 0 0 0 0 0 0 0 1 0 **8** 0 0 0 0 0 1 1 0 0 **9** 0 0 0 0 0 0 0 0 0	PO1, PO2	L2, L3, L4
9.	The transpose of a directed graph $G = (V, E)$ is the graph $G^T = (V, E^T)$, where $E^T = \{(v, u) \in V \times V : (u, v) \in E\}$. Thus, G^T is G with all its edges reversed. Describe an efficient algorithm for computing G^T from G, using: i. adjacency-list representation of G ii. adjacency-matrix representations of G Analyze the running times of your algorithms with explanation.	PO1, PO2, PO3	L2, L3
10.	The incidence matrix of a directed graph $G = (V, E)$ is a $\|V\| \times \|E\|$ matrix $B = (b_{ij})$ such that $b_{ij} = \begin{cases} -1 & \text{If edge } j \text{ leaves vertex } i \\ 1 & \text{If edge } j \text{ enters vertex } i \\ 0 & \text{Otherwise} \end{cases}$ Describe what the entries of the matrix product $B \times B^T$ represent, where B^T is the transpose of B.	PO1	L2, L3
11.	Let $G = (V, E)$ be an undirected graph and let T be the shortest path spanning tree rooted at vertex v. Suppose all the edge weights in G are increased by a constant number k. Is T still the shortest-path spanning tree from v? Explain with an example.	PO1, PO2	L2, L3
12.	We have a connected graph $G = (V, E)$ and a specific vertex$u \in V$. Suppose we compute a depth-first tree rooted at u and obtain a tree T that includes all nodes of G. We then compute a breadth-first search tree rooted at u, and obtain the same tree T. Prove that G = T. (In other words , if T is both a DFS tree and a BFS tree of G rooted at u, then G cannot contain any age that does not belong to T.)	PO1, PO2	L2, L3

13.	Suppose that some edge costs in a directed graph *G* is negative. Make all edge costs positive by adding a fixed positive bias to each cost. Then run Dijkstra's algorithm on this updated graph. Give an example to demonstrate that this algorithm may fail to give the shortest paths in the original algorithm.	PO1, PO2	L2, L3
14.	Modify the Dijkstra's algorithm to compute the number of edges present in the shortest paths along with the shortest distances from the source to each node in the graph *G*. Then find the nodes having the shortest distance among all nodes at k-edge distance from the source for each $k \leq |E|$. Apply this modified algorithm on the following graph and find the maximum value of *k* (i.e., the length of the shortest path from the source to the farthest node in terms of the number of edges). 	PO1, PO2, PO3	L2, L3, L4
15.	Verify the correctness of the Krukal's algorithm to find the MST of a given graph by proving the following statement. "Repeatedly selecting the minimum weight edge available which is not inducing a cycle in the already selected subset of edges will always lead to an MST". Find the MST of the following graph explaining the operations carried out in each step. 	PO1, PO2	L2, L4

Submission and Grading:

Submit the hard copy of your assignment by the due date, i.e. 14.12.2022.

Part of your assignment grade comes from its "external correctness." This is based on correct output on various sample inputs.

The rest of your assignment's score comes from "internal correctness." Internal correctness includes:
1. Use of methods to minimize the number of steps.
2. Appropriate use of rules, axioms, and suitable diagrams to enhance readability of your responses.

Send a zip folder (name of the zip folder must be your registration number_AD1) containing the code and output file/screen-shot of each program implementation mentioned to the official email id of your AD1 class teacher. On the top of each program, you must mention your full name, registration number, title of the program and date.