ASSIGNMENT 4

SECI1013 – Discrete Structure

Semester 2024/2025-1

Instructions:

This is a group assignment. Each group should consist of no more than 3 members.

Please write all your answers by hand using a pen. Ensure that your answers are well-structured and that your handwriting is neat and easy to read. Submissions in printed form will not be accepted.

Please submit your assignment by **4/2/2025, 7:00 PM**, in room **N28-346-05**.

Question 1 **[10 marks]**

You are given the following directed weighted graph:

From	To	Weight
A	B	7
A	C	9
A	F	14
B	C	10
B	D	15
C	D	11
C	F	2
D	E	6
E	F	9

a. Using Dijkstra's algorithm, calculate the shortest paths and their costs from A to all other nodes (B, C, D, E, F). Show all steps of the algorithm, including initialization, updates during iterations, and the final shortest paths.

b. Suppose the edge weight between C → F is reduced to 1, and the edge weight between D → E is increased to 20. Recalculate the shortest paths from A to all other nodes and compare the results with the original graph.

c. If the graph represents a network of roads and E becomes temporarily unavailable due to maintenance, determine how the shortest path tree changes for all destinations from A.

d. The graph is updated to include a new node G, connected as follows:

D → G (3)

E → G (2)

F → G (7)

Determine the shortest path from A to G and describe how this addition affects the overall shortest path tree.

Question 2 **[10 marks]**

You are given a weighted, directed graph representing a road network, where nodes represent cities and edge weights represent the time (in minutes) to travel between two cities. Use Dijkstra's algorithm to find the shortest travel time from City A to all other cities in the network.

Graph Representation:

Number of cities (nodes): 6

Edges (with weights):

- (A → B, 4)
- (A → C, 2)
- (B → C, 5)
- (B → D, 10)
- (C → E, 3)
- (D → F, 11)
- (E → D, 4)
- (E → F, 8)

a. Show the step-by-step process of Dijkstra's algorithm from City A to all other cities. Include the priority queue and distance table at every step.

b. Draw the shortest path tree derived from Dijkstra's algorithm.

c. Identify the longest shortest path (the maximum distance among all the shortest paths from City A) and explain its significance in the context of network reliability.

d. Add a new city G and connect it to cities E and F with weights (E → G, 2) and (G → F, 1). Reapply Dijkstra's algorithm from City A and compare the results to the original graph. Which paths changed, and why?

Question 3 **[10 marks]**

a. Consider a rooted tree with the following structure:

```
      1
     / \
    2   3
   /|   | \
  4 5   6  7
 /     /
8     9
```

i. Define the height of a rooted tree.

ii. Calculate the height and depth of each node in the given tree.

iii. What is the maximum path length from the root to any leaf node in this tree?

b. Given a **full binary tree** where each node has exactly two children, the number of internal nodes is 15. Answer the following:

i. How many total nodes (internal + leaf nodes) are in the tree?

ii. What is the maximum depth (height) of the tree?

iii. What is the **maximum number of leaves** that this tree can have, and what is the relationship between the number of internal nodes and leaves in a full binary tree?

c. Consider the following tree structure, and perform the following tasks:

```
       1
     /   \
    2     3
   / \   /  \
  4   5 6    7
     /     /   \
    8     9   10
```

i. Pre-order traversal: Write the sequence of nodes visited.

ii. Post-order traversal: Write the sequence of nodes visited.

iii. In-order traversal: Write the sequence of nodes visited.

iv. Level-order traversal: Write the sequence of nodes visited.

d. Given the following weighted graph G with vertices $V = \{A, B, C, D, E\}$ and edges E along with their weights:

$A - B: 4$

$A - C: 1$

$B - C: 2$

$B - D: 5$

$C - D: 8$

$C - E: 10$

$D - E: 3$

Using Kruskal's Algorithm, find the Minimum Spanning Tree (MST) of the graph. List the edges included in the MST and the total weight of the MST.

Question 4 **[10 marks]**

You are designing a DFA that recognizes strings over the alphabet {a, b} where the string contains exactly three consecutive 'a's. The DFA must accept strings that contain the substring "aaa" and reject those that do not.

a. Design the states and transitions of a DFA that accepts strings with exactly three consecutive 'a's. Describe the states, transitions, and specify the start and accepting states.

b. Draw the state diagram for the DFA that recognizes strings with the substring "aaa". Then, demonstrate how the DFA processes the string "baaaabb".

c. What happens if the DFA receives an input string that contains more than three consecutive 'a's (e.g., "aaaaa")? How should the DFA handle such cases? Explain the transitions involved.

d. Write the formal transition function for the DFA and explain how the DFA transitions when processing a string like "baaaab".

e. If you were to minimize the DFA designed in Question 1, what changes would you make to reduce the number of states, if any? Explain the process of minimization.

Question 5 **[10 marks]**

You are tasked with designing a Finite State Machine (FSM) that simulates a simple turnstile system for a subway station. The turnstile accepts a token (represented as **'T'**) or a coin (represented as **'C'**) to allow access. The turnstile can either be **locked** or **unlocked**. Initially, the turnstile is **locked** and only unlocks when a **coin** or **token** is inserted. Once it is unlocked, it remains open for a short time, allowing the person to pass through, but then automatically locks again.

The FSM should have two inputs:

- **'C'**: Represents the insertion of a coin.
- **'T'**: Represents the insertion of a token.

The FSM should have the following states:

- **Locked**: The turnstile is locked, and the user must insert a coin or token to pass.
- **Unlocked**: The turnstile is unlocked, and the user is allowed to pass.

a. Design the states and transitions for the FSM based on the scenario. Describe the states, the input events, and how the FSM transitions between states.

b. Draw the state diagram for the turnstile FSM. Then, explain the process step-by-step when a coin is inserted into a locked turnstile, followed by a token insertion after it is unlocked.

c. What should the FSM do if it receives an invalid input, such as a 'C' when the turnstile is already unlocked or a 'T' when the turnstile is already locked? Explain how these cases should be handled.

d. Write the formal transition function for the FSM and explain how it behaves when it processes a sequence of inputs like "CTT".

e. If you were to minimize the FSM designed in Question 1, would any states or transitions be redundant? Explain the process of minimization and whether it's possible to reduce the number of states.