Assignment: 04

Department of Computer Science

Iqra University Islamabad

DSA

Maqsood Ahmed

ID: 3818

Problem 1: [CLO1]

You are managing an online bookstore where books are organized based on their ISBN numbers
(unique for each book). The bookstore uses a Binary Search Tree (BST) to efficiently store and
retrieve book records. The ISBN numbers of the books are as follows:
18000, 15000, 30000, 20000, 10000, 25000, 35000.

Task:

1. Construct a Binary Search Tree (BST) using the given ISBN numbers.
2. The bookstore has implemented a feature to remove books that are out of stock. Write an algorithm and demonstrate the step-by-step procedure to delete the book with ISBN 20000 from the BST.
3. After deletion, ensure the BST property is maintained.

Solution:

Step 1: Constructing the BST

Given ISBN numbers: 18000, 15000, 30000, 20000, 10000, 25000, 35000.

BST Structure:

```
        18000
       /     \
    15000    30000
    /        /    \
10000    20000   35000
            \
            25000
```

Step 2: Deleting ISBN 20000

Algorithm to delete a node in BST:

Case 1: If the node has no children, simply remove it.

Case 2: If the node has one child, replace it with its child.

Case 3: If the node has two children, find the in-order successor (smallest in the right subtree), replace the node with the successor, and delete the successor.

Steps to delete 20000:

1. Locate 20000 (left of 30000).
2. 20000 has one child (25000).
3. Replace 20000 with 25000.

Updated BST:

```
        18000
       /      \
   15000      30000
   /          /   \
10000     25000 35000
```

Problem 2: [CLO1]

An e-commerce platform uses a BST to store the discount percentages for various products. Each product is assigned a unique discount percentage.

Task:

1. Build a BST using the following discount percentages: 20, 10, 30, 5, 15, 25, 35.
2. Write an algorithm to find the product with the **highest discount.**
3. Write another algorithm to find the **second-highest discount** and explain the steps involved.

<u>Solution:</u>

Step 1: Constructing BST

Given percentages: 20, 10, 30, 5, 15, 25, 35.

BST Structure:

```
      20
     /  \
   10    30
   / \   /  \
  5  15 25   35
```

Step 2: Finding the Highest Discount

1. Traverse to the rightmost node.
2. The highest discount is **35**.

Step 3: Finding the Second-Highest Discount

1. If the highest discount node (35) has a left subtree, find the max in that subtree.
2. Otherwise, its parent (30) is the second-highest.
3. Here, **30** is the second-highest discount.

Problem 3: [CLO1]

A stock tracking system uses a BST to store daily closing prices of a stock. The system supports insertion, search, and finding the range of prices.

Task:

1. Build a BST using the following stock prices: 150, 120, 200, 100, 130, 180, 220.
2. Write an algorithm to find the price closest to 170 in the BST.
3. Demonstrate the process to delete the stock price 150 and show the updated BST.

<u>Solution:</u>

Step 1: Constructing BST

Given prices: 150, 120, 200, 100, 130, 180, 220.

BST Structure:

```
      150
     /   \
  120    200
  /  \   /   \
100 130 180  220
```

Step 2: Finding Closest to 170

1. Start at 150 (closer than 200).
2. Move to 200 (further away).
3. Move to 180 (closer than 150 and 200).
4. **Closest price is 180.**

Step 3: Deleting 150

Locate 150 (root).
150 has two children (120 and 200).
Find the in-order successor (180).
Replace 150 with 180.
Delete 180 from its original position.

Updated BST:

```
      180
     /   \
  120    200
  /  \      \
100 130     220
```

Problem 4: [CLO1]

A social media platform models user interactions as a weighted directed graph. Each node represents a user, and each edge represents a mention or tag, with the weight indicating the number of mentions.

Task:

1. Create a graph for the following interactions:
   User 1 → User 2 (mentions: 5), User 1 → User 3 (mentions: 3), User 2 → User 4 (mentions: 8), User 3 → User 4 (mentions: 2), User 4 → User 5 (mentions: 1).
2. Represent the graph in an adjacency matrix.
3. Find the shortest path from User 1 to User 5 based on the weights.
4. Determine the in-degree and out-degree for each user.
5. Convert the graph into a rooted tree with User 1 as the root.

Solution:

Graph Representation:

```
        User1
```

```
        /    \
  User2(5) User3(3)
   |          \
  User4(8)     User4(2)
   |
  User5(1)
```

Adjacency Matrix:

```
  1 2 3 4 5
1 0 5 3 0 0
2 0 0 0 8 0
3 0 0 0 2 0
4 0 0 0 0 1
5 0 0 0 0 0
```

Shortest Path from 1 to 5

Using Dijkstra's Algorithm: 1 → 3 (3) → 4 (2) → 5 (1) → **Total: 6**

In-degree and Out-degree

User	In-degree	Out-degree
1	0	2
2	1	1
3	1	1
4	2	1
5	1	0

Rooted Tree (User 1 as Root)

```
       User1
       /  \
   User2   User3
       \   /
       User4
        \
       User5
```

Problem 5: [CLO1]

An airline models its flight routes as a weighted directed graph. Each node represents an airport, and each edge represents a flight route, with the weight indicating the flight duration (in hours).

Task:

1. Create a graph for the following flight routes:
   Airport X → Y (duration: 2), X → Z (duration: 3), Y → W (duration: 4), Z → W (duration: 1), W → V (duration: 2).
2. Represent the graph using an adjacency matrix.
3. Find the shortest path from X to V.
4. Determine the in-degree and out-degree for each airport.
5. Convert the graph into a rooted tree with X as the root.

Solution:

Graph Representation:

```
      X
     /  \
 Y(2)  Z(3)
   \     /
    W(4)
      |
    V(2)
```

Adjacency Matrix:

```
  X Y Z W V
X 0 2 3 0 0
Y 0 0 0 4 0
Z 0 0 0 1 0
W 0 0 0 0 2
V 0 0 0 0 0
```

Shortest Path from X to V

Path: X → Z (3) → W (1) → V (2) → **Total: 6**

Rooted Tree (X as Root)

```
    X
   / \
  Y   Z
   \   \
    W   W
     \
      V
```

Problem 6: [CLO1]

A company models its supply chain as a weighted directed graph. Each node represents a supplier, and each edge represents a supply route, with the weight indicating the transportation cost.Task:

1. Draw a graph for the following supply chain:
   Supplier S1 → S2 (cost: 50), S1 → S3 (cost: 30), S2 → S4 (cost: 40), S3 → S4 (cost: 20), S4 → S5 (cost: 10).
2. Represent the graph in an adjacency matrix.
3. Calculate the shortest path from S1 to S5 based on transportation costs.
4. Find the in-degree and out-degree of each supplier.
5. Convert the graph into a rooted tree with S1 as the root.

Solution:

Graph Representation:

```
       S1
      /  \
  S2(50) S3(30)
    \      /
     S4(40)
       |
     S5(10)
```

Adjacency Matrix:

	S1	S2	S3	S4	S5
S1	0	50	30	0	0
S2	0	0	0	40	0
S3	0	0	0	20	0
S4	0	0	0	0	10
S5	0	0	0	0	0

Shortest Path from S1 to S5

Path: S1 → S3 (30) → S4 (20) → S5 (10) → **Total: 60**

Rooted Tree (S1 as Root)

```
    S1
   /  \
  S2   S3
   \   /
    S4
     |
    S5
```

The End