COMSATS UNIVERSITY ISLAMABAD

DEPARTMENT OF COMPUTER SCIENCE

Final Term Examination, Fall 2025

Class: BAI-III/BSE-III/BCS-III/BCT-III/BDS-III
Date: 06-01-2026
Subject: CSC211- Data Structures/Data Structures and Algorithms
Instructor:
Total Time Allowed: 180 minutes
Max Marks: 50
Name: [redacted]
Registration Number: [redacted]

NOTE: Do as directed don't go into extra detail. Only crisp and concise answers will earn you credit

- *Students are required to display their ID cards.*
- *Attempt all questions.* ***Return this question paper with your answer sheet.***
- *Use of Mobile Phones and Digital Diaries not allowed*

(CLO-1: Employ linear data structures to solve computing problems.)

Q # 01 **Marks = [2+4+4=10]**

You are managing the **Tech Fair Live Demonstration Schedule**. Each demo belongs to a category. There can be 'n' number of categories. Each category can have 'n' number of demo slots. Design the required C/C++ structures for storing a demo slot (DemoID, StartupName, StartTime).

Implement the following:

Task-1: Design Data Structure — Design appropriate data structure for the scenario and show their links by diagram.
Task-2: insertByCategoryAndDemoID() — Insert a new demo into the list in sorted order (first by category, then by DemoID).
Task-3: deleteByDemoID() — Remove a demo when a team cancels or is disqualified, keeping the schedule correctly ordered.

Q # 02 **Marks = [10]**

You are given a simple queue of integers with enqueue(int) operation and dequeue() operation. Your task is to write a C++ function **void reverse(int k)** that reverses the first k number of elements in the queue.

Constraint: You cannot modify the pre-defined enqueue and dequeue operations or define your own operations. You can use only one additional stack and no other data structure.

(CLO-2: Use non-linear data structures to solve computing problems.)

Q # 03 **Marks = [2.5+2.5+5=10]**

A ride-sharing company manages ride requests with fare amounts and driver ratings to efficiently allocate rides.

Task-1: Assume that you have chosen the **BST** to maintain the above rating. Draw the BST of the above rating and then delete the nodes with fares **300** and **500** from the **already-constructed BST of fares**. Draw the updated BST after both deletions.

Task-2: Drivers are dynamically re-ranked based on rider feedback with new ratings, which are:

[4.8, 4.1, 4.9, 3.6, 4.4, 4.7, 5.0, 4.2]

Insert them one by one into a structure that automatically balances the height of the BST itself whenever an imbalance occurs, draw the BST after each insertion, **apply balancing** wherever required, and show all intermediate diagrams including rotation steps.

Task-3: Assume that ride ratings are already stored using a BST or an AVL tree. Management now requires that rides be executed on a **minimum-fare-first** basis. Under this new requirement, continuing with a BST becomes inefficient,

as each dispatch involves repeatedly locating the minimum-fare ride. Moreover, there is a **strict memory constraint**: you are **not allowed to use additional memory** to reconstruct the entire data structure.
Your task is to write **C++ code** to construct a **priority queue** from the existing BST such that the **highest-priority element can be accessed in constant time**, and after dequeuing, the structure can be rearranged in **O(log n)** time (matching the worst-case complexity of AVL and Heap Trees). You may use **at most 8 bytes of extra memory per record**, so for n records, the total additional memory usage must not exceed **n × 8 bytes**.

(CLO-2: Use non-linear data structures to solve computing problems.)

Q # 04 **Marks = [5+5=10]**

Task-1: A telecom company plans to connect five cities—**Lahore, Islamabad, Faisalabad, and Peshawar**—using fiber-optic cables in such a way that the **total installation cost is minimized**. The cost of laying cable between **Lahore and Islamabad** is 4 million PKR, between **Lahore and Faisalabad** is 3 million PKR, between **Islamabad and Faisalabad** is 5 million PKR, between **Islamabad and Peshawar** is 6 million PKR, and between **Faisalabad and Peshawar** is 2 million PKR.

Sub Tasks:

1. Model the given information as a **weighted undirected graph**.
2. Apply appropriate algorithm for find the **minimum cost** for connection.

Task-2:

You are given a directed graph with **n nodes (0 to n−1)**. Each edge is either **red** or **blue**.
Two lists are provided:

- **redEdges[i] = [a, b]** → directed red edge from a to b
- **blueEdges[j] = [u, v]** → directed blue edge from u to v

Edges may include **self-loops** and **parallel edges**, and all edges have **weight = 1**.
Write **pseudocode** for a function that returns an array **answer[ ]** of size n, where:

- **answer[x]** = length of the **shortest path from node 0 to x**
- **Path must alternate edge colors** (red → blue → red → ... or vice versa)
- If node x is unreachable under *alternating* constraints, return **–1**.

(CLO-3 Apply fundamental sorting, searching, and hashing techniques on different data structures.)

Q # 05 **Marks = [4+3+3=10]**

Task-1: A university wants to store student records efficiently for fast advising assignments.
Reg_IDs are: **FA25-103, FA25-217, FA25-089, FA25-332, FA25-156, FA25-294, FA25-411**

(Use only the numeric part for hashing.)
Use the following hash functions:

$$h_1(x) = x \bmod 11$$

- Construct a hash table using Linear probing and chaining method, when the table size is 11..
- Insert all RegIDs, showing collisions and complete probe sequences.

Task-2: Write the insertion method for Linear probing and also write the structure for the required technique.
Task-3: Student records contain: **RegID, studentName, GPA, completedCredits.** Records arrive mostly in ascending ordered based on RegID, with some slightly out of place due to late corrections. Analyze the dataset and **choose the most suitable sorting technique** based on its characteristics. Implement the chosen sorting technique in C++, and display the sorted output along with a brief justification for your choice.

G D LUCK