National University of Computer and Emerging Sciences

Data Structures (CL2001)

Date: November 11th, 2024

Course Instructor(s)

- Alishba Subhani

HACKATHON

Total Time [illegible] **minutes**
Total Ma[illegible]
Total Qu[illegible] **02**

Semester: SP-2024
Campus: Karachi
Dept.: AI

- Create a ZIP folder of all your solutions and copy it in the local storage with the title K23-xxxx_A.
- Submission are on local storage that can be accessed using win+r keys and entering **\\172.16.5.43**
- address in the dialog box.
- Enter your username as khifast\K23xxxx and its assigned password.
- Zip folder needs to be pasted in the "Exam Submission\teacherName\CourseName\Roll No.

Student Name	Roll No	Section	Student Signature

Q1. Provide an in-depth solution of the given question. Make sure to include every aspect of your suggested approach and write that in a simple word file. [7.5 Points, 2.5 Weightage]

Scenario: In the gaming tournament, player scores gradually decay over time if they are inactive, requiring continuous leaderboard adjustments. The system needs to:

- Efficiently update scores with a decay function that decreases the score of inactive players over time.
- Adjust leaderboard rankings dynamically as scores decay or new scores are added.
- Support efficient range queries for top players within certain score brackets, even with decaying scores.

Challenge: How would you handle efficient score decay, dynamic re-ranking, and range queries without compromising on performance?

Q2. Write the code for the following question and submit a .cpp file. [7.5 Points, 2.5 Weightage]

Implement an "Interval Manager" that tracks non-overlapping intervals. The data structure should support the following:

- **addInterval(start, end)**: Adds a new interval [start, end]. If the interval overlaps with an existing interval, it should be merged with the overlapping interval(s).
- **removeInterval(start, end)**: Removes the interval [start, end].
- **findOverlapping(value)**: Finds all intervals that overlap with the given value.