National University of Computer and Emerging Sciences

Data Structures (CL2001)

Date: December 4th 2024

Course Instructor(s)

Sir Shafique Ms. Zainab Asif Jawed, Sir Misbah

Paper B

Lab Final Exam

Total Time: 120 minutes
Total Marks: 100
Total Questions: 03

Semester: FA-2024
Campus: Karachi
Dept: Computer science

- You must comment your student ID on top of each file in the .cpp and word file.
- Name the .cpp file for each question according to Roll_No e.g. k23-xxxx_Q1.cpp, k23- xxxx _Q2.cpp etc.
- Create a ZIP folder of all your solutions and copy it in the local storage with the title K23-xxxx_B.
- 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.

Student Name	Roll No	Section	Student Signature

CLO # 1: Use & explain concepts related to basic and advanced data structures and describe their usage in terms of common algorithmic operations. **Marks: [15 wtg, 20 marks]**

Imagine we have two computers connected with each other with 50 MB data cable, where one computer needs to send data packets containing certain characters (string). Each character in the data stream represents a resource or a type of operation (such as a task). Each character's frequency is directly tied to how often it appears in the data stream, impacting network performance.
The system has the ability to remove some characters from the data stream to utilize the data cable efficiently so that maximum data can be passed via data cable, but there's a limit to how many characters (denoted as k) can be removed from stream (string). The "**load**" of a string is defined as the sum of squares of the count of each distinct character. For example, consider the string "saideep", here frequencies of characters are s=1, a=1, i=1, e=2, d=1, p=1 and load of the string is 1^2 + 1^2 + 1^2 + 1^2 + 1^2 + 2^2 = 9. The goal is to maximize the the "load" of the stream on the system by removing k characters from stream(string). Implement this using a heap.

Example:

Input: A data packet stream abccc with a limit to remove 1 character (K = 1).
Output: After removing one a or b the remaining characters are bccc, or acc and the resulting "cost" or "value" is 1^2 + 2^3 = 9. If c will be removed then load will be 6 so this is not effiently utilizing the cable

Input : str = aaab, K = 2
Output : 4 because a and b are removed

CLO # 2: Compare different data structures in terms of their relative efficiency and design effective solutions and algorithms that make use of them. **Marks: [20 wtg, 40 marks]**

You are working on a warehouse management system where each item in the warehouse is represented by a unique integer ID and a description (a string). You need to implement a Binary Search Tree (BST) that holds these items, where each node contains an integer (ID) and a string (description).Your task includes the following operations: Insert an item (node) with an ID and description into the BST, Print the k items within a given ID range (e.g., from 10 to 20). Find the closest item to a given ID. Swap the descriptions of nodes whose descriptions contain specific characters provided by the user. For example, swap the descriptions of nodes containing the character 'a' with those containing the character 'b'. Delete the m items whose IDs are multiples of a n number. And print updated tree.

For this implementation:

- k is the number of nodes to print within the given range.
- m is the number of items to delete.
- n is the integer used to find multiples for deletion.
- a and b are the characters used to swap descriptions.

CLO # 3: Use & explain concepts related to basic and advanced data structures and describe their usage in terms of common algorithmic operations. **Marks: [15 wtg, 40 marks]**

You are tasked with developing a system for managing a collection of airline flight codes. Each flight code consists of two uppercase letters (e.g., AA, BA, CZ) representing the airline and the flight number. The system is designed to store a maximum of 20 flight codes in a booking system, with each code mapped to a specific position on the system's storage grid, which contains 20 available positions. When a flight code is entered, the system must ensure that no two codes with the same name occupy the same position. If a conflict occurs (i.e., two codes map to the same position), a new location must be created for the conflicting codes. When a new code is added, the system checks the corresponding position. If the position is already occupied, it will create a new space at that position to store the code.

To retrieve a flight code, the system must organize the codes at each position by their names using an efficient sorting method. The system should ensure that the codes within each position are sorted using a divide-and-conquer method, making retrieval fast and efficient. Additionally, the system must support deleting flight codes, ensuring that if a position becomes empty after deletion, it remains empty or is reused for new entries. The system should also provide functionality to display all flight codes, sorted alphabetically by their names across all positions. Edge cases such as managing up to 20 codes and handling empty positions must also be addressed.

Choose the appropriate type of hashing and sorting algorithm for the above scenario.