National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Data Structures	Course Code:	CS2001
Degree Program:	BS (CS, SE, DS, Robotics)	Semester:	Fall 2023
Exam Duration:	60 Minutes	Total Marks:	20
Paper Date:	11-Nov-2023	Weight	15
Section:	ALL	Page(s):	6
Exam Type:	Midterm-II		

Student: Name: _____ Roll No._____ Section: _____

Instruction/Notes: Attempt all questions. Answer in the space provided. You can ask for rough sheets but will not be attached with this exam. **Answers written on rough sheet will not be marked**. Do not use pencil or red ink to answer the questions**.** In case of confusion or ambiguity make a reasonable assumption.

Question 1: [CLO: 1] **(Marks: 5+3+2)**

a) Considering the following tree as an AVL tree. Show the resulting AVL tree for each of the following cases. Show each step including unbalanced node, and any rotations or transformations applied. If double rotation is applied then also show the intermediate steps.

a. Insert 30

b. Delete 52

b) For each of the scenarios given below, suggest the most appropriate data structure chosen from the list and **give appropriate reason of your choice**.

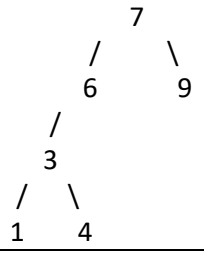
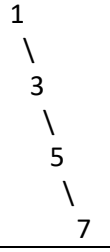
(Arrays, linked-list (single, double, circular), Queue, Stack, tree(BST, AVL))

To implement a dictionary of words in a language.	
Implementing a music playlist with a shuffle and repeat feature.	
Tracking a history of web pages visited by a user in a web browser.	

c) Suppose we want to write a function that requires LIFO order (stack) and memorize the minimum data element of the stack in O(1). If the minimum value is popped from the stack then new minimum should be compued in O(1) time. This can be done using a temporary stack. Explain how a tempoary stack can be utilized to accomplish this task.

Question 2: [CLO: 3] **(Marks: 10)**

Write a **recursive** C++ function ConstructBST() for an integer based binary search tree that takes an array ***preorder*** and its size ***n*** as parameter. The array preorder stores the data of a BST traversed in pre-order. Your task is to construct and return the root of that BST. You may need to pass some other parameters to complete this task. Compute the time complexity of your function. Note that less credit will be given to less efficient solutions.

Sample Input	Sample Output
7,6,3,1,4,9	
1,3,5,7	

What extra parameters you want to pass to this function and what is their significance? Explicitly write the names and types of those parameters and their purpose.

Four extra parameters
Preindex (passed by reference) to pass the index of the array whose data needs to be inserted in the tree
Data – The data element that needs to be inserted and later can be used to specify the range of elements that can be inserted in a particular(sub)tree
Min – The lower limit of the range
Max – The upper limit of the range

Give complete function Definition

Function Prototype

```
node* constructTree(int pre[], int size, int& preIndex, int data, int min, int max)
```

Base case

```
if (preIndex >= size)
            return NULL;
```

Recursive case

```
      node* root = NULL;

      // If current element of pre[] is in range, then
      // only it is part of current subtree
      if (pre[preIndex] > min && pre[preIndex] < max) {
            root = newNode(data);
            preIndex = preIndex + 1;

            if (preIndex < size) {
                 root->left = constructTree(pre, size, preIndex, pre[preIndex], min, data);
            }
            if (preIndex < size) {
                root->right = constructTree(pre, size, preIndex, pre[preIndex], data, max);
            }
      }

      return root;
```

Time Complexity

O(n)

Rough Sheet

Rough Sheet