National University of Computer and Emerging Sciences

Laboratory Manual

for

Data Structures Lab

Course Instructor	Mr. Razi Uddin
Lab Instructor(s)	Mr. Arsil Khan Ms. Hira Ilyas
Section	CS-4J1&2
Date	May 24, 2021
Semester	Spring 2021

Department of Computer Science

FAST-NU, Lahore, Pakistan

Objectives:

In this lab, students will practice:

1. Binary Search Trees and its problems

Note: Use the class StudentBST implemented in the last lab for the following tasks.

```
class StudentBST;
class StudentNode {
      friend class StudentBST;
  private:
      int rollNo;               // Student's roll number (must be unique)
      string name;              // Student's name
      double cgpa;              // Student's CGPA
      StudentNode *left;        // Pointer to the left subtree of a node
      StudentNode *right;       // Pointer to the right subtree of a node
};
class StudentBST {
  private:
      StudentNode *root;        // Pointer to the root node of the BST
  public:
      StudentBST();             // Default constructor
};
```

Question-1 **(5 points)**
(Estimate time: 30 min)

Implement a method **bool remove(int rollno)** which is a member function of StudentBST. This function takes roll no of a student as input if the student is present in the tree then removes it and returns true else false. You also have to balance the tree after the removal.

Question-2 **(5 points)**
(Estimate time: 20 min)

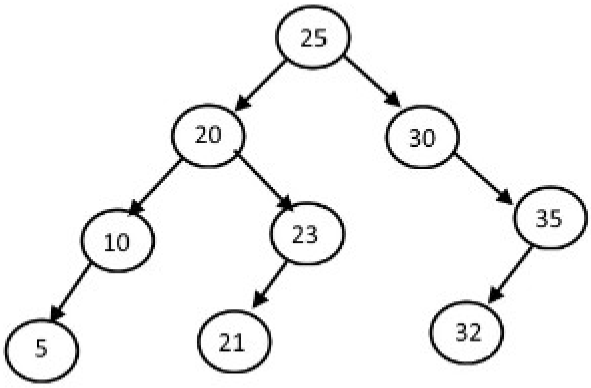
Implement a recursive function **int maxDepth(StudentBST root)** that finds maximum depth of a tree. Note that this function is not a part of StudentBST class.

Question-3 **(5 points)**
(Estimate time: 30 min)

Implement a recursive function **void leafSum(Node *root, int& sum)** to find total leaf nodes of StudentBST.

Question-4 **(10 points)**
(Estimate time: 1 hour)

Find the Lowest Common Ancestor of two given nodes in a StudentBST. In a given binary search tree, The lowest common ancestor of two nodes n1 and n2 will be a node X such that node X will be the lowest node who has n1 and n2 as its descendants.



Lowest Ancestor Ancestor (5, 21) = 20
Lowest Ancestor Ancestor (10, 30) = 25
Lowest Ancestor Ancestor (5, 32) = 25
Lowest Ancestor Ancestor (10, 23) = 20

Implement a recursive function **Node* LCA(Node* root, Node* n1, Node* n2)**