National University of Computer and Emerging Sciences
Lahore Campus

0058

Data Structures and Algorithms (CS2002)

Sessional-II Exam

Total Time (Hrs): 1
Total Marks: 25
Total Questions: 2

Date: November 4th 2024
Course Instructor(s)
Ms. Shazia Haque, Mr. Ahmad Hamza

Roll No _____ Section _____ Student Signature _____

Do not write below this line

Answer all parts of a question together on the answer sheet.

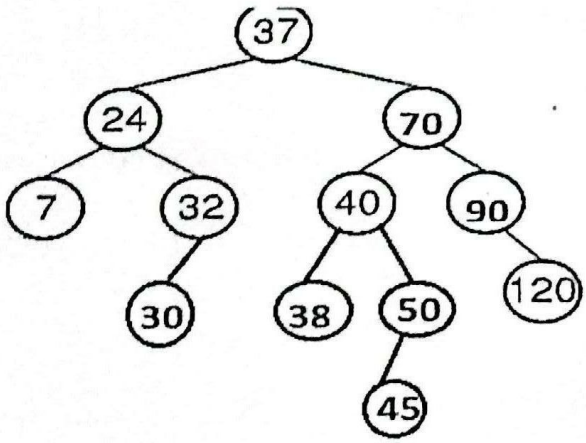
***PLO 3, C5, CLO #3:* Design tree and graph data structures and their associated algorithms**

Q1: [10 marks]

a. Given the following traversals **assemble** a binary tree.

Pre-Order Traversal	ABDCEGHF
Post-Order Traversal	DBGHEFCA

b. Given the following Binary Search Tree



Construct the resulting Binary Search Tree after deleting the node with data value 70 showing all the steps for full credit.

c. **Construct** the Min Heap that the following array represents.

Input Array: [0, 3, 5, 9, 6, 8, 20, 10, 12, 18, 9]
Then Insert a new node with data value 4, showing all the steps for full credit.

***PLO 3, CS, CLO #3:* Design tree and graph data structures and their associated algorithms**

Q2: [15 marks]

a) [7 marks] Suppose we have the following interface of a Binary Tree. **Construct** the C++ code for the PreOrderIterative() member function given below. Node class is the same as discussed in your theory class. The time complexity of the function should not be worse than O(n).

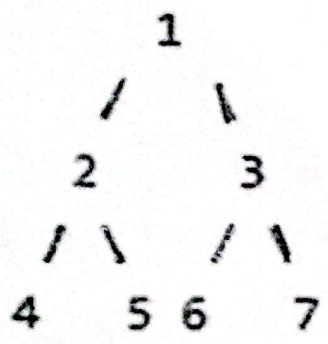
```
template<class DT>
class BinaryTree
{
public:
        BinaryTree();
        BNode<DT> * getRoot();
        void PreOrderIterative();//iterative

private:
        BNode<DT> * root;

};
```

b) **[8 marks] Develop** the following non-member function that **constructs** the mirror image of a given binary tree. The function should return the pointer to a Binary tree which is a mirror image of the BinaryTree passed in as a parameter. Like the interface given in part (a) above you may assume you have access to the root node via the getRoot() function.

.In the mirror image, the left and right subtrees of every node are interchanged. The time complexity should not be worse than O(n).

```
BinaryTree* MirrorImage(BinaryTree* input);
```

Input Tree:



Output Tree (Mirror Image):