Indian Institute of Technology, Guwahati

Department of Computer Science and Engineering

Data Structure Lab: (CS210)

Assignment: 3

Date: 25th August, 2016. Total Marks: 20 (lab assignments) + 30 (Offline assignments)

Last date of Offline Assignment Submission: 31st August, 2016.

Lab Assignments:

1. Write a program to perform the following experiment: Generate 100 random numbers. As each number is generated, insert it into an initially empty binary search tree. When all 100 numbers have been inserted, print the level of the leaf node with largest level and the level of leaf node with smallest level. Repeat this process 50 times, Print out a table with a count of how many of the 50 runs resulted in a difference between the maximum and minimum leaf level of 0, 1, 2, 3, and so on**.** **(20)**

Marks: Insertion of a node in a Binary search tree: 5, Find the level difference between smallest and largest level of leaf nodes: 10, Associate results generation: 5.

Offline Assignments:

2. Write a program that counts the number of leaves of a binary tree. Use the binary search trees generated in Problem 1 to test your program. **(10)**

3. Let T be a tree with n nodes. The lowest common ancestor (LCA) between two nodes v and w as the lowest node in T that has both v and w as descendants (where we allow a node to be a descendant of itself). Given two nodes v and w, describe an efficient algorithm for finding the LCA of v and w. What is the running time of your method? Describe the complexity in comments in program. Use trees in problem 1 to test your algorithm. **(10)**

4. Let T be a tree with n nodes. For any node v in T, let d_v denote the depth of v in T. The distance between two nodes v and w in T is $d_v + d_w – 2d_u$, where u is the LCA u of v and w (as defined in problem 3). The diameter of T is the maximum distance between two nodes in T. Describe an efficient algorithm for finding the diameter of T. What is the running time of your method? Describe the complexity in comments in program. Again use the tree in problem 1 to test your algorithm**.** **(10)**