Data Structures and Algorithms Lab

07. Trees

Subject Code: 17ECSP201 **Lab No:** 07 **Semester:** III

Date: Oct 2017 **Batch:** MSM

Question: Computer Representation of a Binary Search Tree

Objective: Usage of list representation to implement a BST and its operations

----

Implement and add the following functions to the BST code:

1. Print the out-degree of the root node

2. Count the number of edges present in the tree

3. Print the total out-degree of all the leaf nodes

4. Find and delete all the duplicate nodes from the tree. Keep only the first reachable copy out of all copies that are present in the tree.

5. Count the number of nodes having value lesser than the given value K

6. Print the in-order predecessor of the given item

7. Find the maximum valued item from the tree

8. Make a duplicate copy of the existing binary search tree. The function is passed with new root initialized to NULL and existing root of the tree. Wisely decide the return type of the function.

9. Print the address of all the leaf nodes

10. Find and print the number of comparisons made to find the maximum element in the tree

11. Count the number of nodes present at level 1 of the tree

12. Implement the insert_into_bst function using recursion

13. Count and print the number of leaf nodes present in the tree

14. Find the memory occupied by the tree in terms of bytes

15. Implement the recursive Tree search algorithm given below:

```
TREE-SEARCH (x, k)
If x = NULL or k = key[x]
    then return x
If k < key[x]
   then return TREE-SEARCH(left[x], k)
   else return TREE-SEARCH(right[x], k)
```

16. Find and delete all the duplicate nodes from the tree.

17. Count the number of nodes having value greater than the given value K

18. Find the minimum valued item from the tree

19. Print the address of the root node

20. Find and print the number of comparisons made to search a given item from the tree

21. Count the number of nodes present at level 1 of the tree

22. Find the memory occupied by the tree in terms of bytes

23. Find the number of edges between root node and the largest element in the tree

**** Happy Coding ****