

LAB ASSIGNMENT 1

Implement ADT Binary Search Tree using a linked list of nodes of structure having MIS, Name, left pointer, right pointer, and parent pointers pointing to left sub-trees, right sub tree and parent of the node respectively. The BST is organized by the MIS number.

Write following functions:

initBST() // to initialize the tree.
insertNode() // **recursive function** to add a new node to the tree.
removeNode() // to remove a node from a tree.
search() // to **recursively search** for a node with the given key.
postorder() // write a **non-recursive** function for post order traversal.
Display_Level() // to display all nodes of ith level.
destroyTree() // to delete all nodes of a tree

Write a menu driven program to invoke all above functions.

(10 Marks)

Tentative date of submission: 10th April 2022

DATA STRUCTURES AND ALGORITHMS– II (SEM – IV)
DEPARTMENT OF COMPUTER ENGINEERING AND IT
COLLEGE OF ENGINEERING PUNE

LAB ASSIGNMENT 2

A. Implement ADT Binary Search Tree using an Array of integer values.

Include following functions:

1. init_bst()
2. insert_bst() //non-recursive insert
3. recursive_traverse() // inorder, preorder, postorder
4. isComplete() // Check whether the BST is a complete tree

(5 Marks)

B. Accept a post-order traversal of a tree from user. Construct a BST using Linked representation of nodes with the node structure having key, left pointer and right pointer.

(5 Marks)

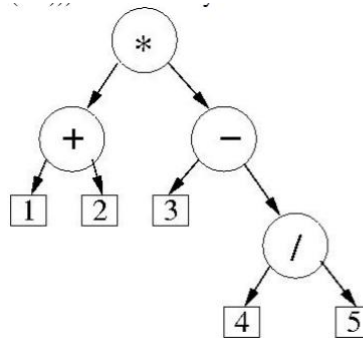
Date of submission: 17th April 2022

LAB ASSIGNMENT 3

One of the applications of Binary Trees is Expression Tree.

An expression tree is a binary tree that models an expression.

For example, the expression $((1+2)*(3-(4/5)))$ is modelled by a tree that looks like this:



Where all internal nodes represent operators and leaf nodes represent operands.

Create an ADT ExpressionTree of operators : +, -, * and / and integer operands.

It has following functions:

1. **InitTree()**: This function creates a new expression tree and populates it using an infix expression read from user. It should appropriately handle errors in the input.
2. **Traverse()**: This function performs in-order traversal of the tree.
3. **Compute()**: This function evaluates each subtree and displays the in-order traversal of intermediate tree created in the process. It returns the final result of the evaluation of the expression.
4. Perform complexity analysis of all functions in a separate text file.

(10 Marks)

Date of submission: 24th April 2022

LAB ASSIGNMENT 4

Implement ADT AVL Tree using a linked list of nodes of structure having Balance_Factor, Month Name, left pointer, right pointer, and parent pointers pointing to left sub-trees, right sub tree and parent of the node respectively. Perform series of insertions of months - December, January, April, March, July, August, October, February, November, May, June.

Write following functions:

initAVL() // to initialize the tree.

insertNode() // recursive/ non-recursive function to add a new node to the AVL tree.

removeNode() // to remove a node from a tree.

traverse() // write any of the traversal method to display month name, parent and balance_factor of each node.

destroyTree() // to delete all nodes of the tree

Write a menu driven program to invoke all above functions.

10 Marks

Date of submission: 1st May 2022

DATA STRUCTURES AND ALGORITHMS– II (SEM – IV)
DEPARTMENT OF COMPUTER ENGINEERING AND IT
COLLEGE OF ENGINEERING PUNE
Heap Sort

LAB ASSIGNMENT 5

Write a program to accept file name containing random number of integers as a command line argument. Sort and display these integers using ADT Heap.

10 Marks

Date of submission: 8th May 2022

DSA - II Assignments 2022 (Ashwini Matange)