**PSG COLLEGE OF TECHNOLOGY**

**DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES**

**M.Sc (SS) – DESIGN AND ANALYSIS OF ALGORITHMS LAB**

**PROBLEM SHEET- I**

1. Write a program to create a binary search tree such that, the first element is the root. The tree cannot contain duplicate values. Perform the following operations on the created tree:
2. Delete the node with minimum value in the tree
3. Finding an element
4. Finding the maximum element
5. Left child of the given node
6. Right child of the given node
7. Traverse the tree in inorder, preorder and postorder
8. Build a BST with the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Add them in that order and find the number of searches would it take to find the number 10.
9. Implement a BST which returns a count of the number of leaf nodes in the tree. A leaf node is any node with empty left and right subtrees.
10. Implement a BST and traverse the tree in level-order.
11. Given an array A of N integers, classify it as being Good Bad or Average. It is called Good, if it contains exactly X distinct integers, Bad if it contains less than X distinct integers and Average if it contains more than X distinct integers.
12. Once Monk was watching a fight between an array and a tree, of being better. Tree got frustrated and converted that array into a Binary Search Tree by inserting the elements as nodes in BST, processing elements in the given order in the array. Now Monk wants to know the height of the created Binary Search Tree. Help Monk for the same.

**Note:**

1) In Binary Search Tree, the left sub-tree contains only nodes with values less than or equal to the parent node; the right sub-tree contains only nodes with values greater than the parent node.

2) Binary Search Tree with one node, has height equal to 1.

1. Given a complete binary tree with *N* nodes and each node have an distinct integer ai attached with it, find the minimum number of swaps you can make to convert the binary tree into binary search tree. In one swap, you can select any two nodes and swap their values. You will be given the array representation of the binary tree. Root of the tree will be at a1 . Left child of root will be at a2 and right child of root will be at a3. Left child of node at array position *k* will be at a2\*k and right child of node at array position *k* will be at a2\*k+1.

8. Monk is standing at the door of his classroom. There are currently **N** students in the class, **i**'th student got **Ai** candies. There are still **M** more students to come. At every instant, a student enters the class and wishes to be seated with a student who has **exactly** the same number of candies. For each student, Monk shouts YES if such a student is found, NO otherwise.

**PSG COLLEGE OF TECHNOLOGY**

**DEPARTMENT OF APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCES**

**M.Sc (SS) – DESIGN AND ANALYSIS OF ALGORITHMS LAB**

**PROBLEM SHEET- II**

1. Insert a given sequence of numbers in an initially empty AVL tree. Display the resulting binary tree after each step. If you have to rotate the tree after inserting an element, display the tree after each rotation.
2. Create a AVL search tree and perform the following operations on it.
3. Delete the node with minimum value in the tree
4. Delete a node with only left child
5. Delete the root node
6. Finding an element
7. Traverse the tree in inorder
8. Find the height of the tree
9. Return a list of all keys in a tree between a and b
10. Count the number of leaves
11. You are given two binary search trees (not necessarily balanced). Design an algorithm that merges the two given trees into a balanced binary search tree in linear time.
12. Given two binary search trees, design a generic algorithm that uses left and right rotations to convert one binary search tree into another binary search tree.