**Lab 9**

**Binary Search Tree**

**Introduction:**

A binary search tree, also called an ordered or sorted binary tree, is a rooted binary tree whose internal nodes each store a key greater than all the keys in the node's left subtree and less than those in its right subtree.

A binary search tree (shown in figure) satisfies the following conditions:

* It is a binary tree
* Every element in it has a unique key
* The keys in a nonempty left subtree (right subtree) are smaller (larger) than the key in the root of subtree
* The left and right subtrees are also binary search trees.



**OBJECTIVE:**

* The objective of this experiment is to implement a Binary Search Tree to be used for search applications.
* Binary search trees are a fundamental data structure used to construct more abstract data structures such as [sets](https://en.wikipedia.org/wiki/Set_(computer_science)), [multisets](https://en.wikipedia.org/wiki/Set_(computer_science)), and associative arrays.
* When inserting or searching for an element in a binary search tree, the key of each visited node has to be compared with the key of the element to be inserted or found.

**APPLICATION:**

* Used to efficiently store data in sorted form in order to access and search stored elements quickly.
* They can be used to represent arithmetic expressions.
* BST used in Unix kernels for managing a set of virtual memory areas.

**ISSUE:**

Faced issue in delete function.

**CONCLUSION:**

Binary search trees are a very powerful (but not perfect) data structure to have in your programming tool belt. If done right, handling large amounts of sorted data becomes easier and quicker.