

Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavrepalanchowk



Lab report on **Lab 3**

IMPLEMENTATION AND TESTING OF BINARY SEARCH TREE

Sub Code: COMP 314

Submitted by:

Saskar Khadka, CE

Roll No: 27

Submitted to:

Dr. Rajani Chulyadyo

Department of Computer Science and Engineering

Submission Date: 21 April, 2023

LAB 3:

Implementation and testing of Binary Search Tree

Binary Search Tree:

A Binary Search Tree is a special type of Binary Tree that allows for an extremely fast searching of data. It is also known as an ordered or a sorted binary tree. A Binary Tree is said to be a Binary Search tree if and only if **(i)** all the nodes of the left subtree are less than the root node, **(ii)** all the nodes of the right subtree are more than the root node **(iii)** both subtrees of each node are also a BST i.e they hold the above two properties. It allows searching of a key in $O(\log(n))$ time.

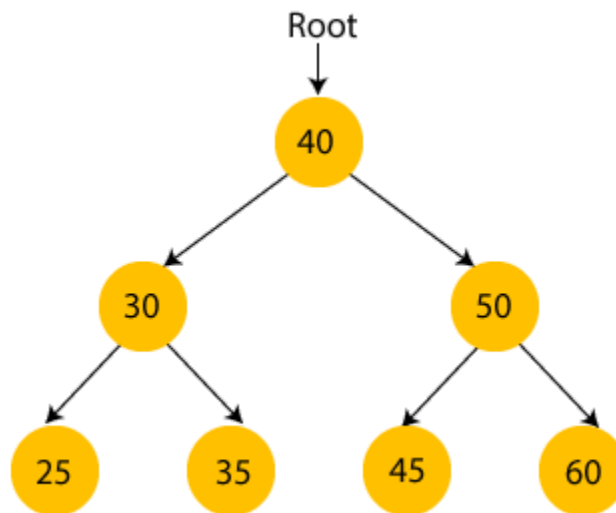


Fig 1: Binary Search Tree

1. Implementation of Binary Search Tree:

[Source Code](#)

1.1 Output:

```
PS C:\Users\Saskarkhadka\OneDrive\Desktop\Algorithms-Labs\Lab 3\src> py main.py
BST empty: True
10 added to BST
40 added to BST
BST empty: False
2 added to BST
1 added to BST
33 added to BST
9 added to BST
Name of student with roll no. 10: Ram
Name of student with roll no. 100: Doesn't exist
23 added to BST
Largest: 40
Smallest: 1
89 added to BST
Largest: 89
Preorder walk: [10, 2, 1, 9, 40, 33, 23, 89]
Inorder walk: [1, 2, 9, 10, 23, 33, 40, 89]
Postorder walk: [1, 9, 2, 23, 33, 89, 40, 10]
40 removed from BST
1 removed from BST
Preorder walk: [10, 2, 9, 33, 23, 89]
Inorder walk: [2, 9, 10, 23, 33, 89]
Postorder walk: [9, 2, 23, 89, 33, 10]
```

1.2 Test Output:

```
.
-----
Ran 11 tests in 0.015s

OK
```

Conclusion:

Hence, a Binary Search Tree was successfully implemented in python and tested according to the given and some added test cases.