National University of Computer & Emerging Sciences (NUCES) Islamabad,
Department of Computer Science

DATA STRUCTURES – FALL 2023

LAB 10

Learning Outcomes

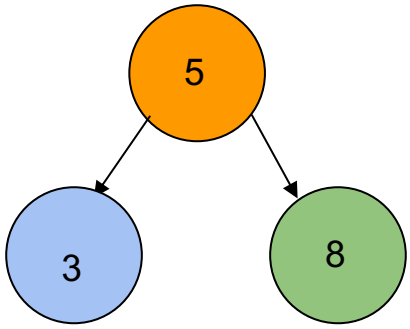
In this lab, you will implement the Binary Search Tree.

Tree ADT

Tree is a widely used nonlinear ADT. **Trees** structure data **hierarchically** based on relationships between the data elements. This means we can access the elements of the tree **non-sequentially**, it also allows trees to use memory more efficiently than linear types.

Structure of Tree

The data elements that make up a tree are called nodes and each node contains some data and pointers that connect the current node to the next nodes. Nodes can have none, one, or many children nodes.

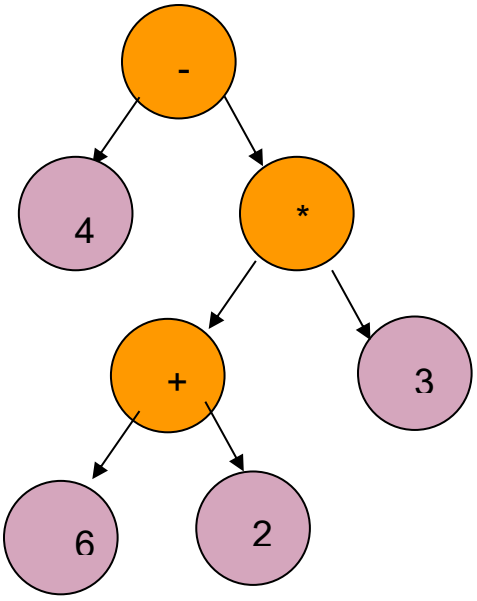


Terminologies in Tree ADT

- Root - the top most node in a tree.
- Parent - A node that is the predecessor of any node.
- Siblings - nodes with the same parent.
- Leaf - a node with no children.
- Internal node - a node with at least one child.

Binary Tree

A binary tree is a tree which has at most two children at each node.

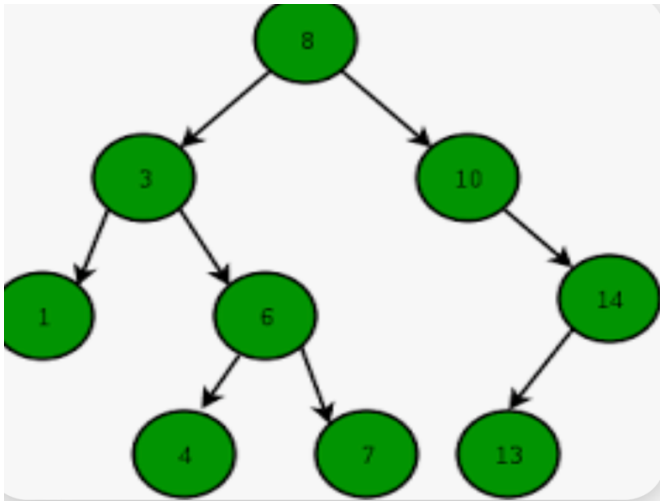


Binary Search Tree

It is a type of binary tree where each node has at most two child nodes, referred to as the left child and the right child. The key property that distinguishes a BST from a regular binary tree is that it maintains a specific ordering of its elements.

The key properties of a Binary Search Tree are:

- Ordering: The values (keys) in the left subtree of a node are smaller than the value at the node, and the values in the right subtree are greater. This ordering property makes it efficient for searching, insertion, and deletion of elements.
- Unique Keys: A BST typically contains unique keys, meaning that no two nodes in the tree have the same key value. This property simplifies search and retrieval operations.



Note: Your program should be menu-based program that allows the user to select and perform each task independently and provide the necessary input and output as specified in the tasks.

```
Menu:
Press 1 to find height of a BST
Press 2 to find Depth of a node in a  BST
Press 3 to count nodes in a range
Press 4 to find kth smallest number of a BST
```

TASK 1

Write a function that finds the height of a binary search tree. (30 minutes)

TASK 2

Write a function that finds the depth of a particular node in a binary search tree. (30 minutes)

TASK 3

Given a BST and a range [L, R], count the number of nodes in the BST that fall within the range (30 minutes).

TASK 4

Find the kth smallest element in a BST. You need to write a function that returns the kth smallest element efficiently. (30 minutes).