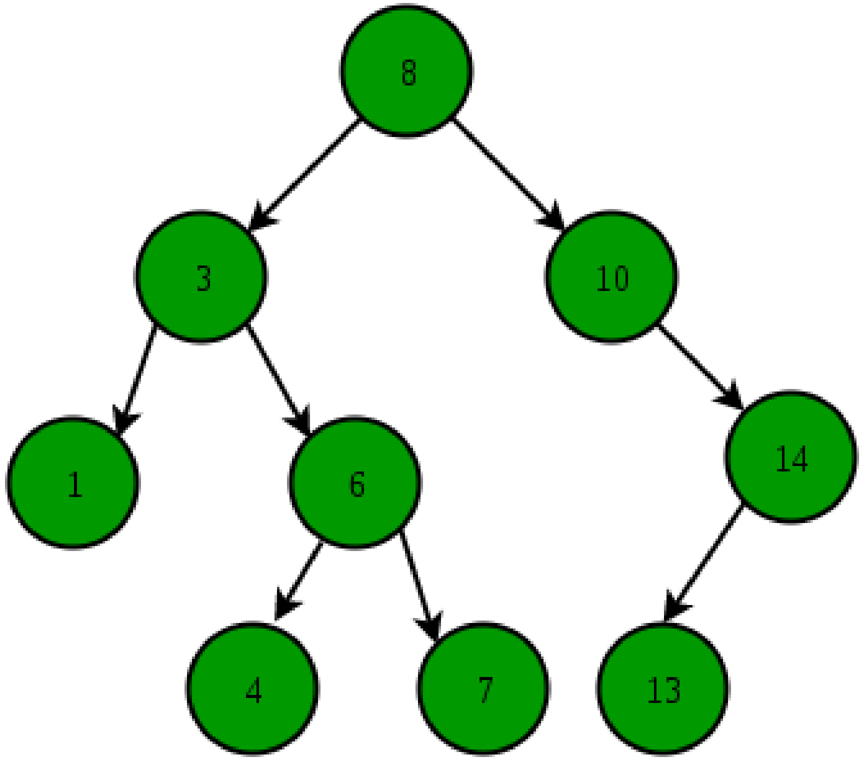
BINARY SEARCH TREE LIBRARY

TREE DATA STRUCTURE

Binary Search tree is a tree data structure in which each node have atmost two childs

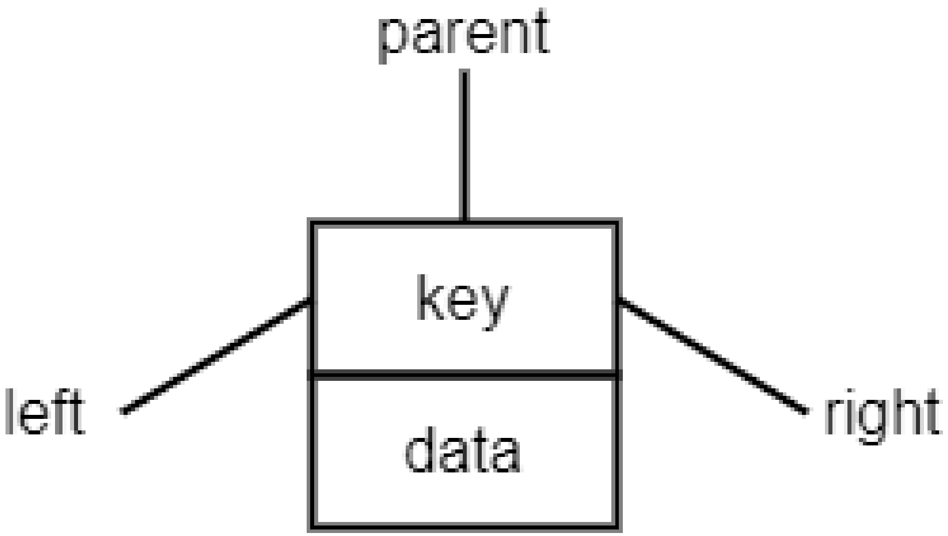
- The left subtree of each node contains nodes having key value lesser than the node's key value
- The right subtree of each node contains nodes having key values greater than the node's key value
- The left and right subtree for each node both form a binary search tree within themselves



Construction

Created a structure of node which has three members with it

- Left -> Which attaches the left subtree of the node
- Right -> Which attaches the right subtree of the node
- Key -> Which stores the key value of the node



BASIC OPERATIONS

- insert (x) -> It Inserts the node x in the Binary Search Tree .
- delete (x) -> It Deletes the node x from the Binary Search Tree .
- search (x) -> It searches the node x in the Binary Search Tree . If it is present then it returns 1 else returns 0 .
- findceil (x) -> It finds the Ceil value of node x that is the smallest value greater than or equal to x . If no such value is present then it returns –1
- findfloor(x) -> It finds the floor value of node x that is the largest value smaller than or equal to x . If no such value is present then it returns –1.

All these Operations have Time Complexity of O(H) where H is the Height of the tree and O(N) in worst case where N is the number of nodes in the tree

Applications and Advantages of Binary Search Tree

- These are used in various Searching Algorithms like searching in maps or sets in various languages
- It is considered as an efficient data structure than Arrays and Linked List . As the searching in Binary Search Tree takes average time of O(log N) time where N is the number of nodes in Binary Search Tree
- These are used to maintain the sorted stream of data for example in some online shopping websites , the data is sorted according to Price with the help of Binary Search Tree