Binary Search Tree (BST) Data Structure

ddas.tech/binary-search-tree-bst-data-structure/

January 2, 2023

Binary Search Tree is a important data structure where each vertex has up to 2 children and follows the below rules.

- All vertices in the left subtree of a vertex must hold a value smaller than its own.
- All vertices in the right subtree of a vertex must hold a value larger than its own.

A BST has ***search*** , ***insert*** and ***remove*** functionality that is implemented.

Properties of a Binary Search Tree

- Root Vertex does not have a parent
- There is only one root vertex
- Leaf Vertex does not have any children
- There can be more than one leaf vertex
- Internal Vertices – Vertices which are not leaf.
- Each Vertices has a parent, left vertex, right vertex and key, value/data

Binary Search Tree Algorithm

BST Search Algorithm
Search (Value)

```
if this == null
   return null
else if this.key == search value
   return this
else if this.key < search Value
   search right
else
   search left
```

BST Find Minimum

```
if this is null
   return null
if left != null
   go left
else
   return this.key
```

BST Find Maximum

```
if this is null
   return null
if right != null
   go right
else
   return this.key
```

Time Complexity

Time complexity is O(h) where h is the height of the Binary Search Tree.