# Option #1: Binary Search Tree

This BST consists of two main components: the Node class representing individual nodes in the tree, and the Tree class representing the entire binary search tree.

## Node Class

The Node class represents a single node in the binary search tree. It contains attributes for storing data as well as references to its left and right children.

A screen shot of a computer code

Description automatically generated

## Attributes

data: Stores the value of the node.

left: Reference to the left child node.

right: Reference to the right child node.

## Tree Class

The Tree class represents the entire binary search tree. It accepts an array of data when initialized, which is used to construct a balanced binary tree.

A screen shot of a computer program

Description automatically generated

### Attributes

root: Represents the root node of the binary search tree.

### Methods

* build\_tree(arr): Constructs a balanced binary tree from the provided array of data. This method sorts the array, removes duplicates, and constructs the tree appropriately.
* insert(value): Inserts a new value into the binary search tree while maintaining its properties.
* delete(value): Deletes a value from the binary search tree while maintaining its properties.

### Usage

Initialize a Tree object with an array of data.

Perform operations such as insertion and deletion using the provided methods (insert and delete).

#### Example

**# Initialize the binary search tree**

arr = [1, 7, 4, 23, 8, 9, 4, 3, 5, 7, 9, 67, 6345, 324]

tree = Tree(arr)

**# Insert a new value**

tree.insert(6)

**# Delete a value**

tree.delete(5)

Root –

A screenshot of a computer

Description automatically generated

Base tree –

A black rectangle with white dots

Description automatically generated

Insert 6

A black rectangle with white dots

Description automatically generated

Delete 5

A black screen with white text

Description automatically generated

## Code Repo: [SchoolPython/CSC506\_DAA/Module6\_BST/BST.py at main · ArunSaxena200/SchoolPython (github.com)](https://github.com/ArunSaxena200/SchoolPython/blob/main/CSC506_DAA/Module6_BST/BST.py)

# Conclusion

The provided implementation of a binary search tree offers efficient operations for maintaining sorted data and facilitates quick insertion and deletion of values. You can leverage the functionality of the Node and Tree classes to manipulate and traverse binary search trees effectively.