Readme

How the Code Works

Inserting Nodes:

- Nodes are inserted into the BST by comparing keys. If the key is smaller than the current node's key, the new node is inserted into the left subtree. Otherwise, it's inserted into the right subtree.

Searching for a Node:

- When searching for a node, the tree is traversed from the root, comparing the search key to the keys of the nodes. The algorithm moves left or right based on the comparison, until it finds the matching key or reaches a None node.

Inorder Traversal:

- Inorder traversal ensures that the keys are printed in sorted order. The tree is traversed left-to-right, recursively visiting the left subtree, the root, and then the right subtree.

Deleting a Node:

- Deleting a node involves three cases:
    - Node has no children: simply remove the node.
    - Node has one child: replace the node with its child.
    - Node has two children: replace the node with its in-order successor (the smallest key in the right subtree).