# Code Reflection and Pseudocode

This code implements a Binary Search Tree (BST) to manage bids for property auctions. The primary purpose of this implementation is to efficiently insert, search, and remove bids while maintaining the properties of a binary search tree.  
  
To solve various problems encountered during development, techniques such as error handling, modular programming, and clear documentation were employed. For instance, the use of modular methods allowed for easier debugging and testing of individual functionalities, such as loading bids from CSV files and performing operations on the BST. Challenges included ensuring the correctness of tree operations (insertion, deletion, searching) and handling potential errors when processing input data. These challenges were overcome by implementing thorough testing and incorporating validation checks for input data.  
  
Overall, the code meets the specified requirements and behaves as intended, with attention to modularity and readability, making it suitable for further enhancements and maintenance.

Pseudocode for Binary Search Tree Operations:  
1. Initialize the Binary Search Tree (BST).  
2. Function insert(bid):  
 a. If the tree is empty, create a new node with bid and set it as root.  
 b. If the tree is not empty:  
 i. Compare bid with current node.  
 ii. If bid < current node, insert in the left subtree.  
 iii. If bid >= current node, insert in the right subtree.  
  
3. Function search(bid):  
 a. Start at the root.  
 b. If current node is NULL, bid is not found.  
 c. If bid == current node, return found.  
 d. If bid < current node, search in the left subtree.  
 e. If bid >= current node, search in the right subtree.  
  
4. Function remove(bid):  
 a. If the tree is empty, return.  
 b. Find the node to be deleted using search().  
 c. If the node has no children, simply remove it.  
 d. If the node has one child, replace it with its child.  
 e. If the node has two children:  
 i. Find the in-order predecessor or successor.  
 ii. Replace the value of the node with the predecessor/successor.  
 iii. Remove the predecessor/successor node.  
  
5. Function displayAll():  
 a. Traverse the tree in-order to display all bids in sorted order.  
 b. For each node, print the bid value.  
  
6. Function main():  
 a. Load bids from CSV file into the BST.  
 b. Provide a menu for user interactions (add, search, remove, display).  
 c. Call appropriate functions based on user input.