Pseudocode Module 6 (Binary Search Tree)

START

INPUT root and set equal to nullptr

INPUT Left\_Node and Right\_Node

COMPUTE inOrder to pass in root

INPUT Bid functions

IF root equal to nullptr

SET root equal to new node for a bid

ELSE add the node root and bid

COMPUTE Insert function to add Bid within binary search tree

COMPUTE Remove function to delete parameters root and Bid

COMPUTE search function, parameter Bid

SET curr Node equal to root

WHILE curr Node is not equal to nullptr

IF curr Bid is found, return curr bid

IF the bid is less than curr node

Go down left branch

ELSE

Go down right branch

END

COMPUTE addNode function with parameter Node pointer node and Bid

IF node is greater than root

SET child to left branch

IF no left node exists

SET node left

ELSE

IF no right node

SET node right

ELSE go to the left down tree

END

COMPUTE inOrder

IF node is not equal to nullptr

SET node to left

DISPLAY node bid details

ELSE

SET node to right for loop to traverse entire tree in order

COMPUTE removeNode

IF node equal to nullptr

SET node

IF Bid given is less than node

DELETE lefty node

ELSEIF Bid is greater than 0

DELETE right node

ELSE

IF left node is nullptr and right node is nullptr

DELETE node

Node equals nullptr

ELSE IF left node not nullptr and right node is not nullptr

SET temp

SET node left

DELETE temp

ELSE IF right node not nullptr and left node is not nullptr

SET temp

SET node right

DELETE temp

ELSE node temp equals right

WHILE temp left is not nullptr

SET temp left

SET Bid to temp

DELETE node right

RETURN NODE

END