* **Code Reflection:** A brief explanation of the code and its purpose, and a brief discussion of your experience in developing it, including any issues that you encountered while completing the exercise and what approaches you took to solve them.
  + This assignment was probably the easiest for me to do and understand. The only issue I had with it was the absence of the removeNode function. I searched for it forever before deciding it wasn’t there, then I had to figure out how to put it together.
* **Pseudocode or Flowchart:** A pseudocode or flowchart description of the code that is clear and understandable and captures accurate logic to translate to the programming language

START main()

PRINT menu()

IF case “1”:

CALL loadBids()

IF case “2”:

CALL InOrder():

CALL inOrder(root):

IF node IS NOT NULL:

RECURSIVELY CALL inOrder(left node)

PRINT bidID, title, amount, fund

RECURSIVELY CALL inOrder(right node)

BREAK

IF case “3”:

PRINT "Input the bid ID you would like to find: "

INPUT bidKey

START clock()

bid = CALL Search(bidKey):

INIT “current” node AS root

WHILE there is stuff for “current” to do:

IF current bidId IS THE SAME as “bidId”:

RETURN current bid

IF current bidId IS LESS THAN “bidId”:

“current” = current left node

ELSE:

“current” = current right node

RETURN bid;

END clock

IF bid IS NOT empty:

CALL displayBid()

ELSE:

PRINT “bid ID not found”

PRINT time in seconds

BREAK

IF case “4”:

PRINT “Input the bid ID you would like to remove: ”

CALL Remove(bidkey):

CALL removeNode(root, “bidId”):

IF node is NULL:

RETURN node

IF “bidId” IS LESS THAN node bidId:

Left node = CALL removeNode(left node, “bidId”)

ELSE IF “bidId” IS GREATER THAN node bidId:

Right node = CALL removeNode(right node, “bidId)

ELSE:

IF left AND right node are NULL:

DELETE node

Node = NULL

ELSE IF left node IS NOT NULL AND right node IS NULL:

INIT NODE “temp” = node

node = left node

delete “temp”

ELSE IF right node IS NOT NULL AND left node IS NULL:

INIT NODE “temp” = node

node = right node

delete “temp”

ELSE:

INIT NODE “temp” = right node

WHILE temp left IS NOT NULL:

Temp = temp left

Bid node = temp bid

Right node = CALL removeNode(right node, temp bidId)

DELETE temp

RETURN node

PRINT “Goodbye.”