I have completed this assignment individually, without support from anyone else. I hereby accept that only the below listed sources are approved to be used during this assignment:

(i) course textbook

(ii) all material that is made available to me by the professor

(iii) notes taken by me during lectures

I have not used, accessed or taken any unpermitted information from any other source. Hence, all effort belongs to me.

Emir Fatih Aygıldız

EFA

Pseudo Code

```
Class Node
 right left item height
 Node(value)
 right left item height

inserting (root data)
if root=null
  new node
if item > data
root.left = inserting (rec)
if item< data
root.right=inserting (rec)
return root

balancing root
  a= getheight
  if  -1< a <1
    return true
  return false

get Height root
 int left right
if root left exists
   left = getheight root left +1 (rec)
if root right exists
   right = getheight root right+1 (rec)
   return left-right

deleting root item
if root item = item
  root = null
if item< root item
 root left = deleting (rec)
if item > root item
 root right = deleting (rec)
return root

comparing ( first node second node )
if (first = second=null)
  return true
if first and second exists
return compare lefts     add compare rights
if only 1 is null
  return false
```

time complexity

inserting takes O(log n) time because binary

getHeight takes O(n) time because checks all nodes

balancing takes O(n) because of getHeight

deleting takes O(logn) times because it finds the node using binary search

comparing takes O(n) time because it checks and compares all nodes

space complexity

each node has 2 integers height and item and 2 pointers. Nodes have O(n) complexity with every insertion, combined space increase to $O(n^2)$. Comparing with others, the space complexity of the program is $O(n^2)$