Data Structure Homework 8

繳交期限：　　　2020/12/8 17:00 前

補交期限(7 折)：　2020/12/15 17:00 前

手寫題：

題目敘述中若有說明 show the balance factor，請像講義一樣在結果 AVL tree 上的每個 node 旁寫上 LH、EH 或 RH

2. Balance the AVL tree in Figure 8-18. Show the balance factors in the result.

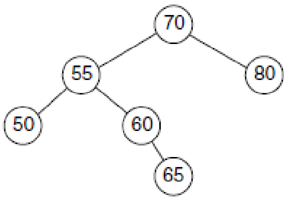


FIGURE 8-18 Figure for Exercise 2

6. Create an AVL tree using the following data entered as a sequential set. Show the balance factors in the resulting tree:

```
14 23 7 10 33 56 80 66 70
```

8. Create an AVL tree using the following data entered as a sequential set. Show the balance factors in the resulting tree:

```
80 70 66 56 33 23 14 10 7
```

10. Insert 44 and 50 into the tree created in Exercise 7.

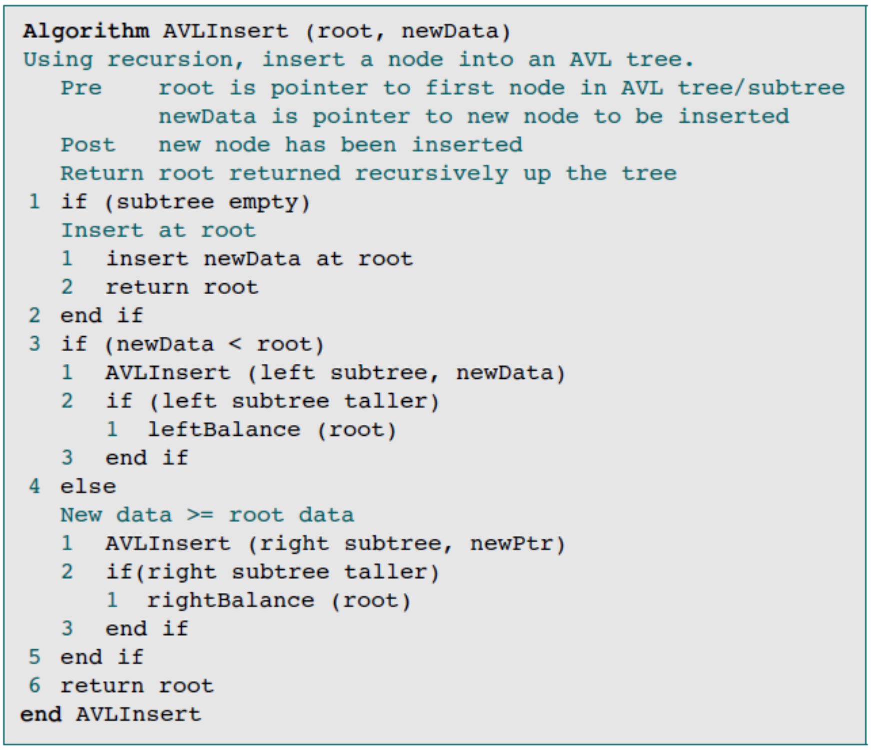
Exercise 7:

7. Create an AVL tree using the following data entered as a sequential set. Show the balance factors in the resulting tree:

```
7 10 14 23 33 56 66 70 80
```

14. Write an iterative version of Algorithm 8-1, “AVL Tree Insert.”

ALGORITHM 8-1 AVL Tree Insert



```
Algorithm AVLInsert (root, newData)
Using recursion, insert a node into an AVL tree.
   Pre    root is pointer to first node in AVL tree/subtree
          newData is pointer to new node to be inserted
   Post   new node has been inserted
   Return root returned recursively up the tree
1  if (subtree empty)
   Insert at root
   1  insert newData at root
   2  return root
2  end if
3  if (newData < root)
   1  AVLInsert (left subtree, newData)
   2  if (left subtree taller)
      1  leftBalance (root)
   3  end if
4  else
   New data >= root data
   1  AVLInsert (right subtree, newPtr)
   2  if(right subtree taller)
      1  rightBalance (root)
   3  end if
5  end if
6  return root
end AVLInsert
```

說明：

1. 請改寫 ALGORITHM 8-1 使其功能不變但不能呼叫 AVLinsert 自己。
2. 原演算法中有使用的 function 或寫法可直接使用不用定義(例如：rightBalance、left subtree taller、<operator of node pointer)。
3. 請在 50 行內完成。