Project 1

Please create a Binary search tree class to implement a Tree interface and use a TreeNode class below:

```
private static class TreeNode {
        Object element;
        TreeNode left;
        TreeNode right;

        public TreeNode(Object o) {
          element = o;
        }

}
```

```
public interface Tree<E>{

  /** Return true if the element is in the tree*/
  public boolean search(E e);

   /** Insert element e into the binary search tree.
   Return true if the element is inserted successfully*/
  public boolean insert(E e);

  /** Delete the specified element from the tree.
  * Return true if the element is deleted successfully*/
  public boolean delete(E e);

  /*Inorder traversal from the root*/
  public void inorder();

  /*postorder traversal from the root*/
  public void postorder();

  /*preorder traversal from the root*/
  public void preorder();

  /* Get the number of nodes in the tree*/
  public int getSize();

  /**return true if the tree is empty*/
  public boolean isEmpty();
```

}

Question 1: (*Implement postorder traversal without using recursion*) Implement the postorder method in BST using a stack instead of recursion.

public void postorderNoRecursion()

Question 2: (*Find the non-leaves*) Add a method in the BST class to return the number of the non-leaves as follows:

/** Returns the number of non-leaf nodes */

public int getNumberofNonLeaves()

If your team has three people, you will need to implement question3.

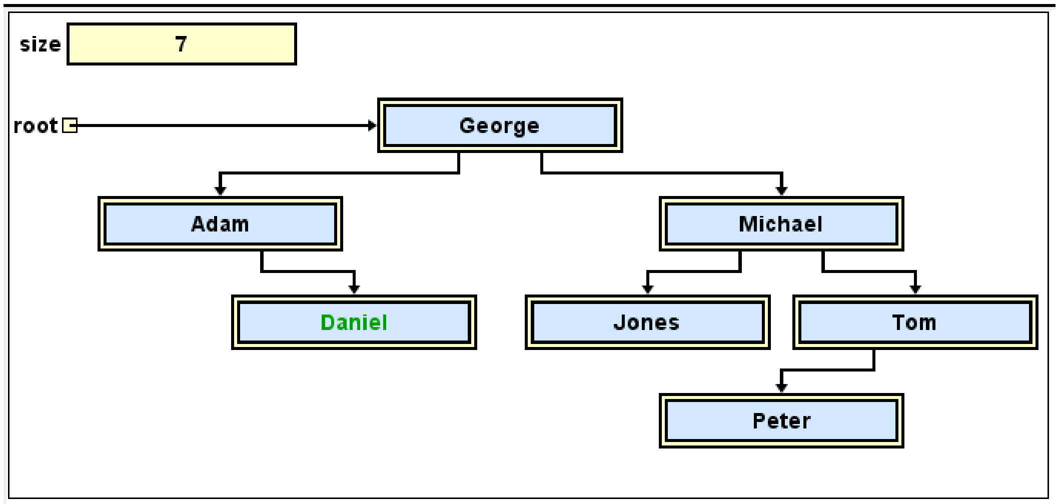
Question3: (*Implement inorder traversal without using recursion*) Implement the inorder method in BST using a stack instead of recursion.

public void inorderNoRecursion()

Each team submits only one copy. *The Binary Search Tree class should be named as a format of Group#.java. An example of group 1, the name of the class is Group01 and file name should be Group1.java.Do not write your main function in this binary search tree class. Create another class as a driver class that has main function for testing. The Binary Search tree class should contain only the definitions of all methods. Ex:* You could test your code by inserting the names in this order:

George
Michael
Tom
Adam
Jone
Peter
Daniel

In the figure below, this tree has 7 nodes. There are four of them are non-leaf nodes. They are George, Adam, Michael, and Tom. You do not need to implement visualization of the tree. This is just used for illustration.



Academic integrity is important for this class. You could check other supplementary data as references. However, you cannot copy their code or it would be considered as plagiarism.

Indicate which team member does what work.

Example:

Group1: Amy, Alex

Amy: question 1

Alex: question 2