CS 2401 Assignment #10

Due Date: Sunday, December 03, 2017 11:59PM (See the syllabus for late policy)

Objective: The goal of this assignment is to practice implementing binary search tree.

Assignment:

I was feeling very generous this weekend due to the upcoming thanksgiving spirit. Out of tremendous kindness of my heart, I already wrote all the three files for you that are required for this assignment – BTNode.java, BST.java, and Runner.java.

BTNode.java can be used as the node of any binary tree. It is provided below.

```
class BTNode{
  Object data;
  BTNode left;
  BTNode right;

  BTNode(){}
  BTNode(Object obj){
    data = obj;
  }
}
```

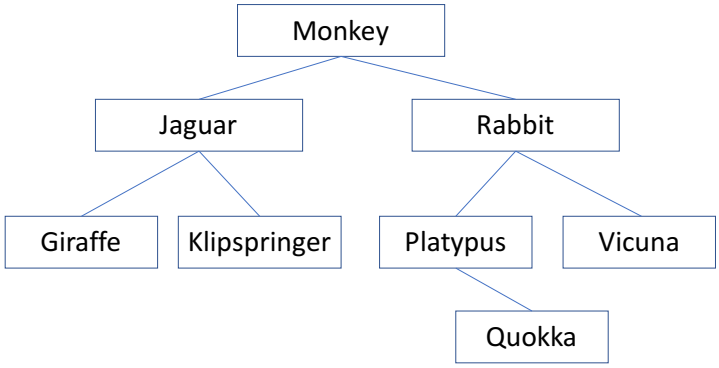
Runner.java is the class that tests if the binary search tree provided in BST.java works well. The code for Runner.java is here.

```
class Runner{
  public static void main(String[] args){
    BST bst = new BST();
    bst.insert("Monkey");
    bst.insert("Jaguar");
    bst.insert("Rabbit");
    bst.insert("Platypus");
    bst.insert("Giraffe");
    bst.insert("Klipspringer");
    bst.insert("Vicuna");
    bst.insert("Quokka");

    System.out.println("----------------------------");
    System.out.println("Printing BST:");
    bst.printBT();
    System.out.println("----------------------------");

    System.out.print("Total number of nodes: ");
    System.out.println(bst.size());
    System.out.println("----------------------------");

    System.out.println("Printing BST in ascending order:");
    bst.printAscending();
    System.out.println("----------------------------");

    System.out.println("Printing BST in descending order:");
    bst.printDescending();
    System.out.println("----------------------------");

    System.out.print("The longest string is: ");
    System.out.println(bst.getLongestString());

  }
}
```

Everything was going fine. My program was running well until my dog randomly pressed the keyboard and deleted some of the codes from the BST.java file, while I was away. By the way, BST.java is the file that contained all the methods to support calls made from Runner.java. Although I could write the deleted parts again, I thought seeking your help in reviving the code will be a great practice. Here is the leftover of the BST.java code that I have now after my dog ate it partially. (It sounds like the famous "my dog ate my homework" excuse.)

```
class BST{
  BTNode root;
  int count;

  BST(){}

  BST(String str){
    root = new BTNode(str);
  }

  /**
   * @return Number of elements in the binary search tree.
   */
  public int size(){

  }

  /**
   * Insert the string in the parameter into the Binary Search Tree.
   * @param str
   * @return true if insertion is successful.
   */
  public boolean insert(String str){

  }

  /**
   * Print the binary search tree in the format shown in the output in next page.
   */
  public void printBT(){

  }

  /**
   * Print the elements of the binary search tree in ascending order
   * (lexicographic order).
   */
  public void printAscending(){

  }

  /**
   * Print the elements of the binary search tree in descending order.
   */
  public void printDescending(){

  }

  /**
  * Return the longest string of the binary search tree.
  * @return the longest string
  */
  public String getLongestString(){

  }
}
```

In the Runner class I basically constructed the following binary search tree.



Things were so good before my dog destroyed some parts of BST.java. I even have the compiled class files. I obtained the following output by executing the compiled classes.

```
----------------------------
Printing BST:
-Monkey
 -Jaguar
  -Giraffe
   -
   -
  -Klipspringer
   -
   -
 -Rabbit
  -Platypus
   -
   -Quokka
    -
    -
  -Vicuna
   -
   -
----------------------------
Total number of nodes: 8
----------------------------
Printing BST in ascending order:
Giraffe
Jaguar
Klipspringer
Monkey
Platypus
Quokka
Rabbit
Vicuna
----------------------------
Printing BST in descending order:
Vicuna
Rabbit
Quokka
Platypus
Monkey
Klipspringer
Jaguar
Giraffe
----------------------------
The longest string is: Klipspringer
```

Please reconstruct BST.java in such a way that the output of Runner.java does not change. I appreciate your help in reviving the code.

Your TA will ask you to change the Runner.java file slightly during the demo to construct a slightly different binary search tree to verify if your BST is working.

Deliverables: You will need to submit three Java files (BTNode.java, BST.java, and Runner.java) using Blackboard. Your TA will instruct you with further details.