Programming and Data Structures
Active Learning Activity 9: Linked Lists and Binary Search Trees

Activity Objectives

At the end of this activity, students should be able to:

1. Implement generic data structures for Linked List and Binary Search Tree

2. Instantiate the two generic data structures in a test program

3. Store an animal database in the two instantiated data structures

4. Compare the performance of the search operations for each data structure

Activity

---

3. Implement the generic classes **LinkedList** and **BST** as seen in class. Add the method **searchIterations()** to both classes to search for an item and return the number of iterations performed by the method to find the item or not. The header of the method is shown below:

    ```
    public int searchIterations(E item)
    ```

4. Analyze the time complexity of the method **searchIterations()** in both classes and include it in the code as a comment.

5. In the main method, instantiate **LinkedList** and **BST** for the type **String** and name the instances **animalLL** and **animalBST** respectively.

6. Read the file **animals.txt** and add each animal name to an array list named **list**.

7. Iterate through the elements of **list** and add each element to the two data structures **animalLL** and **animalBST** using the methods **add()** and **insert()** respectively.

8. Generate ten (10) random integers with value from **0** to **list.size()** and search for the random animal names in both **animalLL** and **animaBST** using the method **searchIterations**. Display the results returned by the method for each animal and for each data structure. Display the average number of iterations for each data structure.

9. Use the method **Collections.sort()** to sort the elements of **list**. Clear the data structures **animalLL** and **animalBST** using the method **clear()**. Repeat steps 7 and 8 using the sorted **list**.

10. Compare the results found when an unsorted list is used versus a sorted list. Explain the difference between the two if any.

1. Submit the following files on coursesite:
    LinkedList.java, **BST.java**, and **LL_BST.java** on courseSite.

Here is a sample run of the program. Please note that the numbers may vary because the animal names are generated randomly.

```
-------------------------------------------------- Sample RUN --------------------------------------------------
Comparing search operations on Linked List and BST using a shuffled list

Animal                             Iterations (LL)           Iterations (BST)
Black widow spider                 355                       14
Koi                                446                       14
Amphibian                          320                       11
Yak                                214                       7
Rabbit                             157                       10
Sailfish                           490                       13
Elephant                           158                       12
Hookworm                           162                       10
Domestic pigeon                    127                       13
Hammerhead shark                   410                       16
------------------------------------------------------------------------------
Average # iterations               283                       12


Comparing search operations on Linked List and BST using a sorted list

Animal                             Iterations (LL)           Iterations (BST)
Domestic goat                      131                       123
list                               498                       473
Stoat                              428                       405
Pig                                337                       317
Sawfish                            388                       366
Steelhead trout                    426                       403
Ostrich                            319                       299
Blue bird                          52                        51
Meerkat                            292                       272
Star-nosed mole                    424                       401
------------------------------------------------------------------------------
Average # iterations               329                       311
```