Programming and Data Structures
Programming Project 3: Binary Search Trees

Project Objectives

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

1. Implement the generic data structure for Binary Search Tree
2. Instantiate the generic data structures in a test program
3. Use the BST data structure to store a country database
4. Invoke different methods on the BST data structure to determine specific characteristics of the constructed BST
5. Determine the performance of the operations on the BST data structure

Project Description

---

1. Create the generic class **`BST`** that has the following UML diagram.

`BST<E extends Comparable<E>`	Generic Binary Search Tree Class
`-root: TreeNode`	Reference to the first node in the BST
`-size: int`	Number of nodes in the BST
`+BST()`	Creates an empty BST by initializing **`root`** to **`null`** and **`size`** to **`0`**
`+BST(ArrayList<E> list)`	Creates an empty BST and insert the elements from **`list`** in the BST
`+size(): int`	Returns the value of **`size`**
`+isEmpty(): boolean`	Returns **`true`** is the BST is empty
`+clear(): void`	Reset **`root`** to **`null`** and **`size`** to **`0`**
`+search(E item): int`	Returns the number of iterations to determine if **`item`** is in the BST, or not
`+insert(E item): int`	Returns the number of iterations to insert **`item`** in the BST or find it in the BST
`+delete(E item): int`	Returns the number of iterations to delete **`item`** from the BST if **`item`** is found
`+leafNodes(): int`	Recursive method that returns the number of leaf nodes in the BST
`+height: int`	Recursive method that returns the height of the tree
`+isBalanced(): boolean`	Recursive method that returns **`true`** if the BST is balanced, or **`false`** otherwise

`+preorder(): void`	Recursive method that prints the values of the nodes in the BST in **preorder**
`+postorder(): void`	Recursive method that prints the values of the nodes in the BST in **postorder**
`+inorder(): void`	Recursive method that prints the values of the nodes in the BST in **inorder**

2. Determine the time complexity of all the methods in class **`BST`** and insert it as a comment before the methods' header.
3. Create a class **`Country`** to store information about a country. The class has three data members: code (two letters), name of the country, and the area in square kilometers. Add appropriate constructors, getters, and setters for the class. The class should implement the interface **`Comparable`** for type **`Country`** and therefore you must include a definition for the method **`compareTo()`** to compare two countries based on their names.

4. Create a test program named **`CountryDB`** to do the following:
    a. Declare, outside the main method, three static arrays of type **`int`** as follows:

    ```
    static int[][] performance_insert = new int[2][228];
    static int[][] performance_search = new int[2][10];
    static int[][] performance_delete = new int[2][10];
    ```

    b. In the main method, create an array list **`countryList`** of type **`Country`**. Fill in **`countryList`** with data loaded from the file "**`countries.txt`**". The provided text file contains the international country database maintained by the US Census Bureau.
    c. Instantiate the class **`BST`** for the type **`Country`** and name the class instance **`CountryBST`**.
    d. Insert the elements from **`countryList`** into **`countryBST`** using the method **`insert()`** and save the number of iterations returned by **`insert()`**, for each element, in the array **`performance_insert[0]`**.

e. Display the nodes of **countryBST** using the **inorder** traversal. Note that the list of countries is displayed with country names sorted alphabetically because the field **name** is used for the comparison in the method **compareTo()**.

f. Display the number of leaf nodes and the height of **countryBST**. Display a message indicating if **countryBST** is balanced or not.

g. Generate ten (10) random integers between 0 and **countryList.size()-1**. Search, in **countryBST**, for the ten random countries that you get at the ten random indices in **countryList**. Save the number of iterations returned by **search()**, for each country, in **performance_search[0]**. Delete the same ten random countries from **countryBST** and save the number of iterations returned by **delete()** in **performance_delete[0]**.

h. Shuffle the array list **countryList** and clear **countryBST**. Insert the elements from **countryList** into **countryBST** and save the number of iterations returned by the method **insert()** in the array **performance_insert[1]**.

i. Display the nodes of **countryBST** using the **inorder** traversal. Note that the list of countries is displayed with country names sorted alphabetically.

j. Display the number of leaf nodes and the height of **countryBST**. Display a message indicating if **countryBST** is balanced or not.

k. Repeat step **g** using the new **countryBST**. Save the number of iterations returned by **search()** and **delete()** in **performance_search[1]** and **performance_delete[1]** respectively.

l. Compare the results obtained for step **i** to the results for step **f**. Discuss the results. Add your discussion as a comment after the main method.

m. Display the content of the three arrays **performance_search**, **performance_delete**, and 10 randomly selected values from **performance_insert** as shown in the sample run of the program. Discuss the results obtained for each one of the BST operations: **insert()**, **search()**, and **delete()**. Add your discussion as a comment after the main method.

5. Submit the following files on coursesite: **`Country.java`**, **`BST.java`**, and **`CountryDB.java`**. Do not forget to include Javadoc comments in all your classes.

---------- **Sample Run** ----------

```
Sorted Country List
Code        Name                                    Area (sq.Km)
AF          Afghanistan                             652230
AL          Albania                                 27398
AG          Algeria                                 2381741
AQ          American Samoa                          198
AN          Andorra                                 468
AO          Angola                                  1246700
AV          Anguilla                                91
AC          Antigua and Barbuda                     443
AR          Argentina                               2736690
AM          Armenia                                 28203
AA          Aruba                                   180
AS          Australia                               7682300
AU          Austria                                 82445
AJ          Azerbaijan                              82629
BF          Bahamas, The                            10010
.
.
.
UK          United Kingdom                          241930
US          United States                           9148655
UY          Uruguay                                 175015
UZ          Uzbekistan                              425400
NH          Vanuatu                                 12189
VE          Venezuela                               882050
VM          Vietnam                                 310070
VI          Virgin Islands, British                 151
VQ          Virgin Islands, U.S.                    348
WF          Wallis and Futuna                       142
WE          West Bank                               5640
WI          Western Sahara                          266000
YM          Yemen                                   527968
ZA          Zambia                                  743398
ZI          Zimbabwe                                386847


Number of leaf Nodes: 1
Height of the tree:   228
Is the tree balanced? false
```

Shuffled Country List

Code	Name	Area (sq.Km)
AF	Afghanistan	652230
AL	Albania	27398
AG	Algeria	2381741
AQ	American Samoa	198
AN	Andorra	468
AO	Angola	1246700
AV	Anguilla	91
AC	Antigua and Barbuda	443
AR	Argentina	2736690
AM	Armenia	28203
AA	Aruba	180
AS	Australia	7682300
AU	Austria	82445
AJ	Azerbaijan	82629
BF	Bahamas, The	10010
.		
.		
.		
UK	United Kingdom	241930
US	United States	9148655
UY	Uruguay	175015
UZ	Uzbekistan	425400
NH	Vanuatu	12189
VE	Venezuela	882050
VM	Vietnam	310070
VI	Virgin Islands, British	151
VQ	Virgin Islands, U.S.	348
WF	Wallis and Futuna	142
WE	West Bank	5640
WI	Western Sahara	266000
YM	Yemen	527968
ZA	Zambia	743398
ZI	Zimbabwe	386847

Number of leaf Nodes: 78
Height of the tree: 14
Is the tree balanced? false

Sorted List			Shuffled List		
Insert	Search	Delete	Insert	Search	Delete
186	24	24	11	11	11
101	77	77	7	9	9
138	175	175	7	11	11
121	150	150	8	10	10
166	27	27	9	10	10
141	181	181	10	9	9
108	79	79	10	12	12
26	8	8	5	10	10
4	134	134	2	5	5
19	164	164	3	11	11