

Programming and Data Structures
Active Learning Activity 8: Binary Trees (BST and Heap)

Activity Objectives

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

1. Implement the generic data structures for binary search tree and heap
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 operations (contains, add, and remove) on the two data structures

Activity

1. Create the class **BST** as seen in class.
2. Create the class **Heap** as seen in class.
3. Modify the methods **contains**, **add**, and **remove** to return the number of iterations performed by these operations in the classes **BST** and **Heap**.

public int contains(E item)

public int add(E item)

public int remove(E item) for the BST

public int remove() for the Heap

4. Create a test program named **AnimalDB** to do the following:
 - a. Instantiate the **BST** and **Heap** classes for the type **String** and name the two instances **animalBST** and **animalHeap** respectively.
 - b. Read the file "**animals.txt**" and add each line from the file to **animalBST**, **animalHeap**, and to an array list **animalAL**. Display the number of iterations of the methods **add** each time 24 animals have been read from the file. Determine the average number of iterations of the methods **add** after reading the complete file for **animalBST** and **animalHeap**.
 - c. Select 20 random animal names from **animalAL** and search for these names in **animalBST** and **animalHeap**. Display the number of iterations of the method

- contains** in **animalBST** and **animalHeap** for each animal. Display the average number of iterations of the two **contains** methods.
- d. Select 20 random animal names from **animalAL** and remove these names from **animalBST** and call the method **remove** on **animalHeap** 20 times. Display the number of iterations of the methods **remove** for **animalBST** and **animalHeap**. Display the average number of iterations of the two **remove** methods.
- e. Compare and discuss the results obtained with the big-O time complexity of the methods **contains**, **add**, and **remove**.
5. Submit the following files on Github. Javadoc comments are not required.
- BST.java**, **Heap.java**, and **AnimalDB.java**.

===== Sample output of the program =====

Testing add()	BST	Heap
Water buffalo breeds	0	0
Goose	7	2
Wildfowl	1	5
Vampire bat	6	4
Cattle	6	1
Domestic pigeon	11	1
Shrimp	9	2
Termite	7	3
Zebra finch	5	7
Chimpanzee	6	1
Wolf	4	6
Tiger shark	6	3
Platypus	9	1
Sea slug	13	3
Bald eagle	12	1
Landfowl	12	2
Fowl	12	3
English pointer	13	2
Quokka	9	4
Blackbird	14	1
Tortoise	8	5
Average	9	2

Testing contains()	BST	Heap
Water buffalo breeds	13	468
Parrot	9	45
Chipmunk	9	395
Kangaroo mouse	11	374
Fancy rat varieties	12	387
Deer	10	134
Grizzly bear	5	195
Mollusk	10	304
Duck breeds	8	274
Gamefowl	13	211
Tarsier	14	342
Hippopotamus	10	174
Cow	12	367
Ape	12	340
Grouse	13	335
Urinal	8	82
Manta ray	8	274
Hornet	9	79
Eel	11	190
Marmot	8	178
Average	10	257

Testing remove()	BST	Heap
Water buffalo breeds	5	8
Parrot	12	9
Chipmunk	5	9
Kangaroo mouse	13	9
Fancy rat varieties	10	9
Deer	11	9
Grizzly bear	11	8
Mollusk	9	9
Duck breeds	11	9
Gamefowl	9	9
Tarsier	11	8
Hippopotamus	7	9
Cow	12	8
Ape	12	8
Grouse	12	9
Urinal	11	9
Manta ray	10	9
Hornet	14	9
Eel	8	9
Marmot	10	9
Average	10	8