GIT Department of Computer Engineering
CSE 222/505 - Spring 2022
Homework #5 Report

171044073
Berkan AKIN

1. System Requirements

In the 1st and 2nd questions, we are asked to make the calculations and show them together with the calculation steps.
In question 3, we are asked to create a heap using node-list. functions with the desired Heap structure and combining 2 Heaps.
In questions 3 and 4, it should be able to work with any class that inherits from Comparable. For the 3rd question, the BinaryTree class should be implemented.
For the 4th question, the SearchTree interface must be implemented. Functions must be overridden in Array Based BST.

2) Problem Solution Approach

Regarding my system's requirements and problems, i created a container class to keep and modift the data easily. Then I was able to set up a hierarchy and find a solution, by correctly determining the class relationships and the ease provided by my container class.

PROBLEM SOLUTION APPROACH My Problem solution steps are;

1. – Specify the problem requirements
2. – Analyze the problem
3. – Design an algorithm and Program
4. – Implement the algorithm
5. – Test and verify the program
6. – Maintain and update the program

3) Test Case

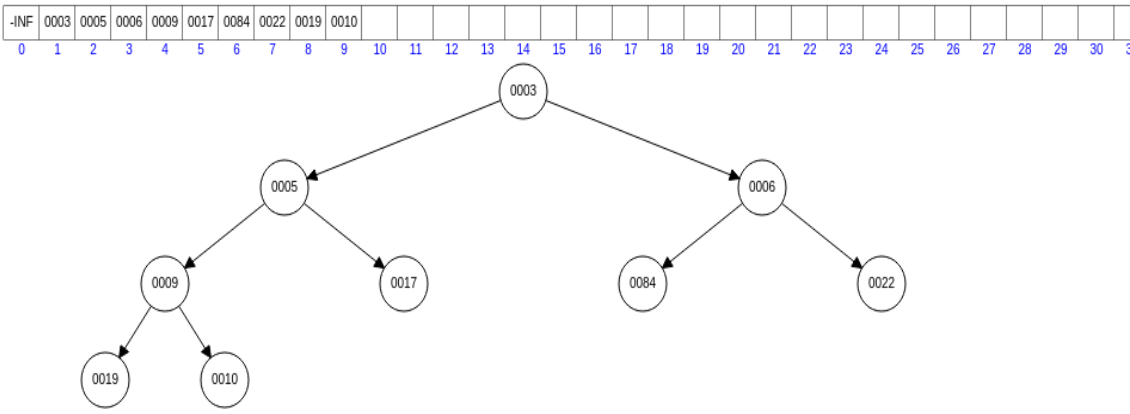
Tests of Question 3

Create BinaryHeap Class Object

```
public static void main(String[] args) {
    BinaryHeap minHeap = new BinaryHeap(15);
    BinaryHeap minHeap2 = new BinaryHeap(15);
```

Minheap add element and embodied heap representation

```
minHeap.insert(17);
minHeap.insert(10);
minHeap.insert(84);
minHeap.insert(19);
minHeap.insert(6);
minHeap.insert(3);
minHeap.insert(22);
minHeap.insert(9);
minHeap.insert(5);
minHeap.minHeap();
```

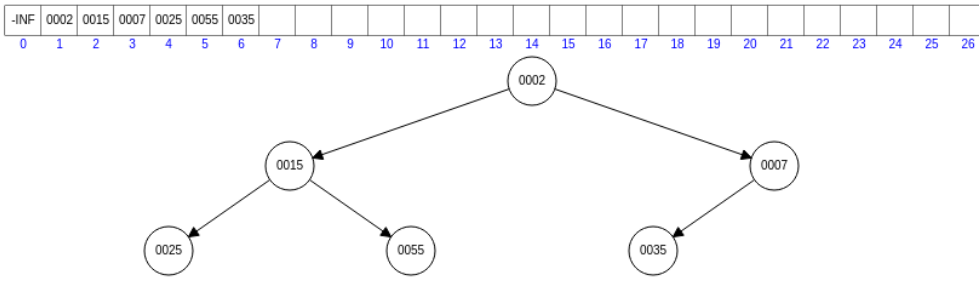


BinaryHeap print Test and remove test

```
 minHeap.inOrder(minHeap.getRoot());
 System.out.println("\nTree Representation");
 System.out.println(minHeap.toString());
System.out.println("\nRemove min element");
 minHeap.removet();
 minHeap.minHeap();
minHeap.inOrder(minHeap.getRoot());
System.out.println("\ndata:" +minHeap.getRoot().data);
System.out.println("data right:" +minHeap.getRoot().right.data);
System.out.println("data left:" +minHeap.getRoot().left.data);
```

Binnary Heap add element (minheap2) and embodied heap representation

```
minHeap2.insert(15);
minHeap2.insert(25);
minHeap2.insert(35);
minHeap2.insert(2);
minHeap2.insert(55);
minHeap2.insert(7);
minHeap2.minHeap();
```



Merge Test add minHeap2 in minHeap and print test

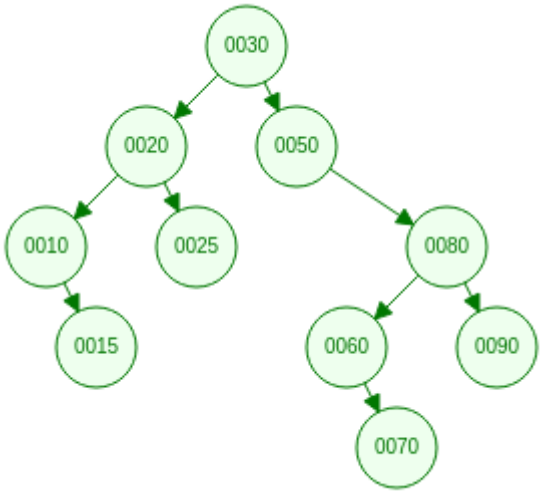
```
minHeap.heapMerge(minHeap, minHeap2);
minHeap.minHeap();
System.out.println("After Merger");
minHeap.inOrder(minHeap.getRoot());

System.out.println("\ndata root:" +minHeap.getRoot().data);
System.out.println("data right:" +minHeap.getRoot().right.data);
System.out.println("data left:" +minHeap.getRoot().left.data);
```

Tests of Question 4

Array Based Binary search tree add method test and embodied heap representation

```
arrayTree.add(30);
arrayTree.add(20);
arrayTree.add(50);
arrayTree.add(10);
arrayTree.add(15);
arrayTree.add(80);
arrayTree.add(25);
arrayTree.add(70);
arrayTree.add(90);
arrayTree.add(60);
```



Contains Method Test

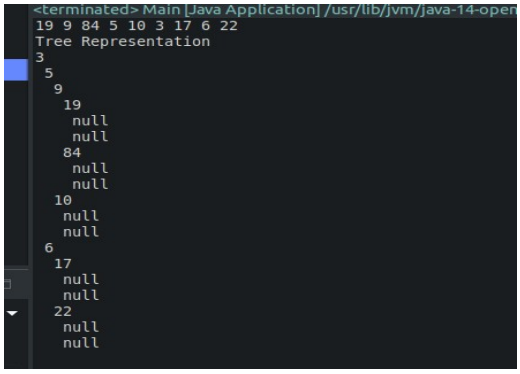
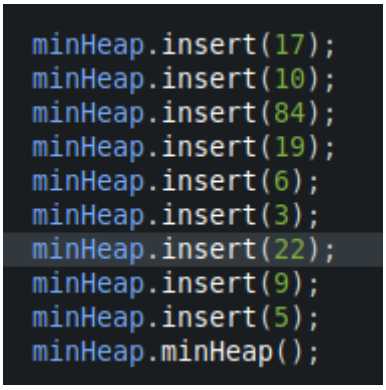
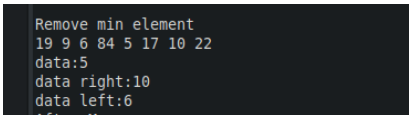
```
System.out.println("Contains 50: " + arrayTree.contains(50));
System.out.println("Contains 100: " + arrayTree.contains(100));
```

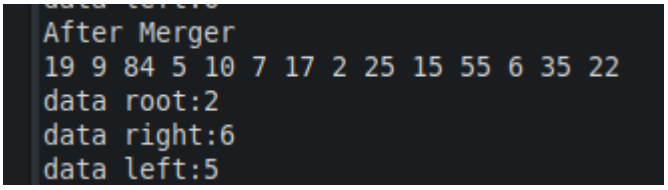
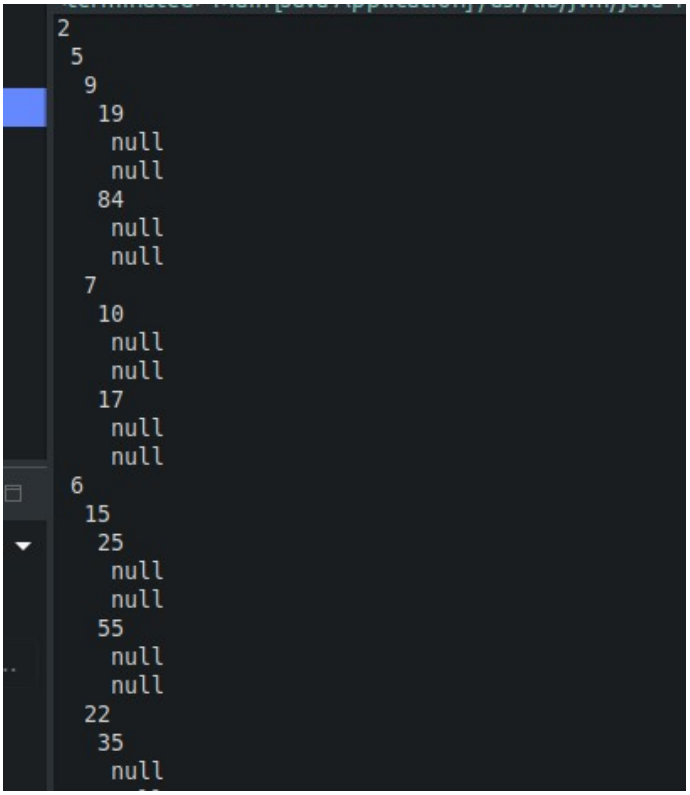
Find Method Test

```
System.out.println("Contains 80: " + arrayTree.find(80));
System.out.println("Contains 100: " + arrayTree.find(100));
```

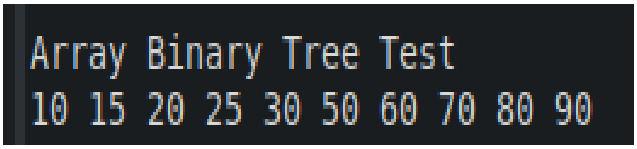
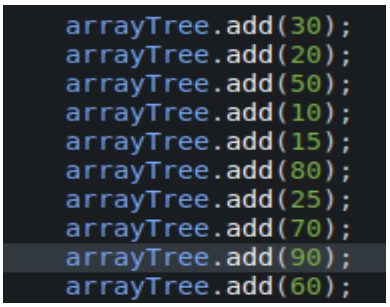
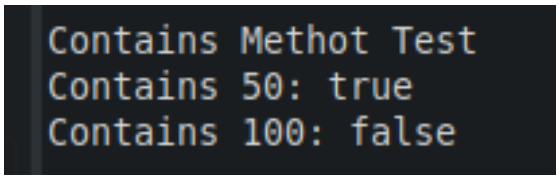
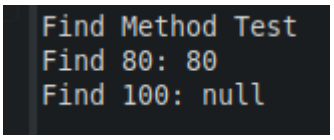
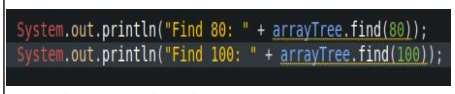
4)Test And Results

Question 3

Output	Test	Result
	 Insert test	Pass
	 Remove test	Pass

	 Merge() method Test	Pass
	Merge() method test	

Question 4

Output	Test	Result
	  Array Add and InOrder printTest	Pass
	 Contains() Method Test	Pass
	 Find() Method test	Pass