GIT Department of Computer Engineering
CSE 222/505 - Spring 2022
Homework 7 Report

Abdurrahman BULUT
1901042258

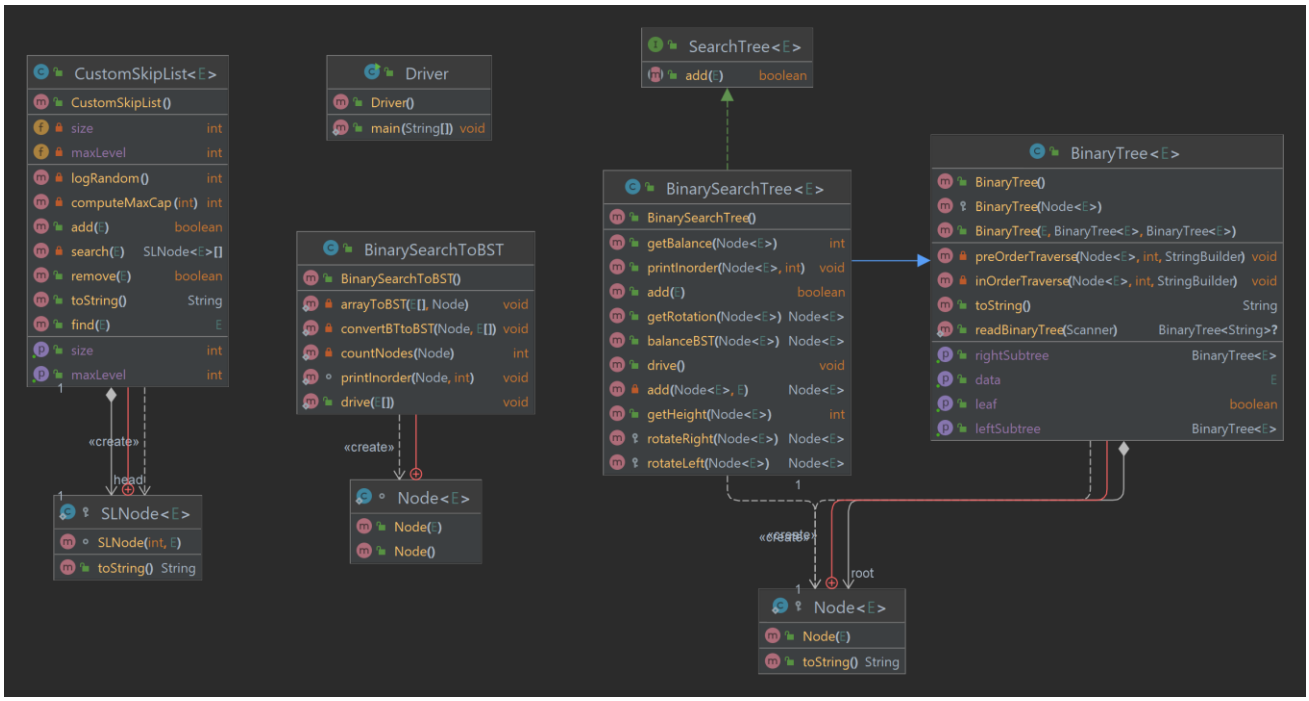
1. SYSTEM REQUIREMENTS

➢ Functional Requirements

♦ System

- openjdk 17.0.2 2022-01-18 LTS
- OpenJDK Runtime Environment Corretto-17.0.2.8.1 (build 17.0.2+8-LTS)
- OpenJDK 64-Bit Server VM Corretto-17.0.2.8.1 (build 17.0.2+8-LTS, mixed mode, sharing)

2. CLASS DIAGRAM



3. PROBLEM SOLUTION APPROACH

For first part, I created a temporary array and stored elements from taking binary tree as inorder traversal. Then I sorted that array. Finally, I copied array elements to the tree with inorder traversal one by one. Copying elements to the array takes O(n). I used quick sort to sort items, and it takes O(nlogn). In second question, we are converting a binary search tree to AVL tree which is balanced tree. I made it with rotation operation as requested. A binary search tree is balanced if the depth of the two subtrees of every node never differs by more than 1. I assume that the left and right sub-trees are balanced at each node. The two sub-trees must now be combined with the node. Condition for balanced tree is that abs (left height - right height)) <= 1. Do Nothing since the root node is already balanced. The root node is skewed to the left, and the left sub-tree has a height of or more. (Configuration of LL) -> Simply do a right rotation to balance this. Do two swaps in the LR configuration. To make the left sub-tree root an LL configuration, left rotate it first, then right rotate it. RR configuration -> Rotate the root left RL configuration -> Rotate the right sub-tree root first to make it an RR configuration Then rotate the root to the left. We must ensure that the swapped nodes form a balanced sub-tree whenever we rotate!

4. TEST CASES

Test 1:

System.out.println("** HW7 DRIVER CODE **");
System.out.println("** Question 1 Start**");

BinarySearchToBST bsttobst = new BinarySearchToBST ();
Integer [] values = new Integer [] {23, 45, 7, 46, 14};
System.out.println("integer array created with values [23, 45, 7, 46, 14]");
bsttobst.drive(values);

```
System.out.println("Binary tree structure is creating...");
root = new Node (0);
System.out.println("Head node created. Value: 23");
root.left = new Node(0);
System.out.println("left node created with value 45");
root.right = new Node(0);
System.out.println("right node created with value 7");
root.left.left = new Node(0);
System.out.println("left -> left node created with value 46");
root.left.right = new Node(0);
System.out.println("left -> right node created with value 14");
```

System.out.println("I was able to show the version rotated 90 degrees counterclockwise in console.");
System.out.println("It keeps the original structure of Binary Tree when converting BST");
System.out.println("** Question 1 END **");

Test 2:

```
Integer[] values2 = new Integer[]{7, 3, 1, 28, 55};
```

Same structure used with above

Test 3:

```
BinarySearchTree<Integer> bst = new BinarySearchTree<>();
System.out.println("Binary search tree created");
stead(12);
System.out.println("12 added to BST");
bst.add(23);
System.out.println("23 added to BST");
bst.add(66);
System.out.println("66 added to BST");
bst.add(1);
System.out.println("1 added to BST");
bst.add(4);
System.out.println("4 added to BST");
bst.add(95);
System.out.println("95 added to BST");
```

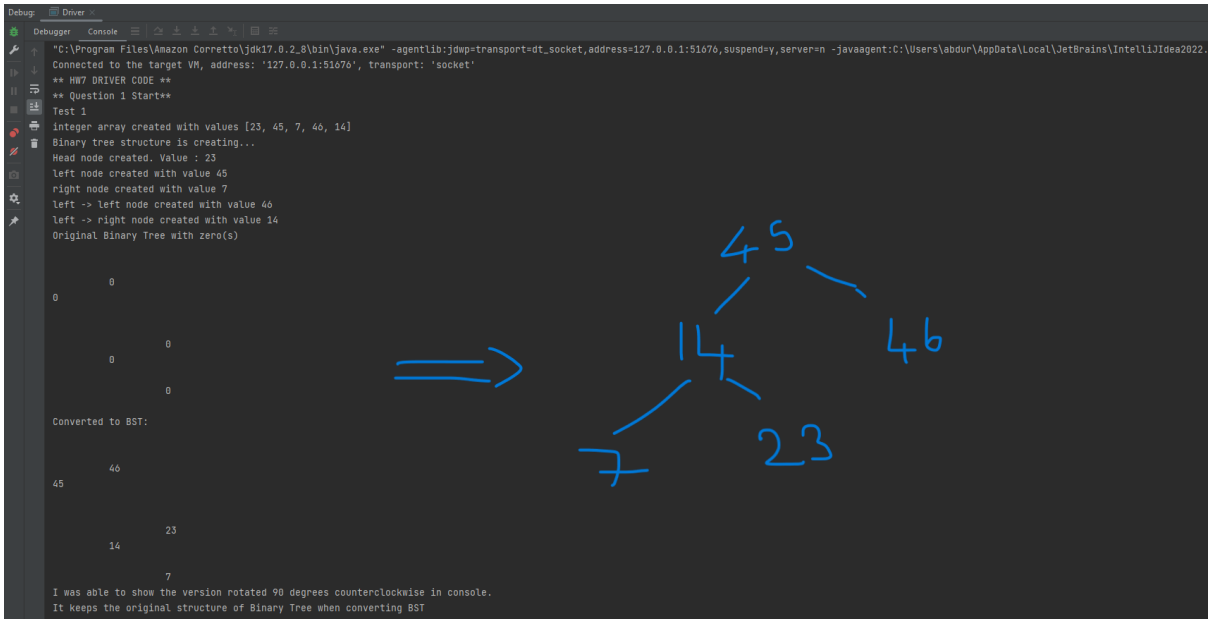
```
bst.add(63);
System.out.println("63 added to BST");
bst.add(33);
System.out.println("33 added to BST");
bst.add(34);
System.out.println("34 added to BST");
bst.add(35);
System.out.println("35 added to BST");
System.out.println("Created BST");
bst.drive();
System.out.println("BST is converting to AVL tree");
```

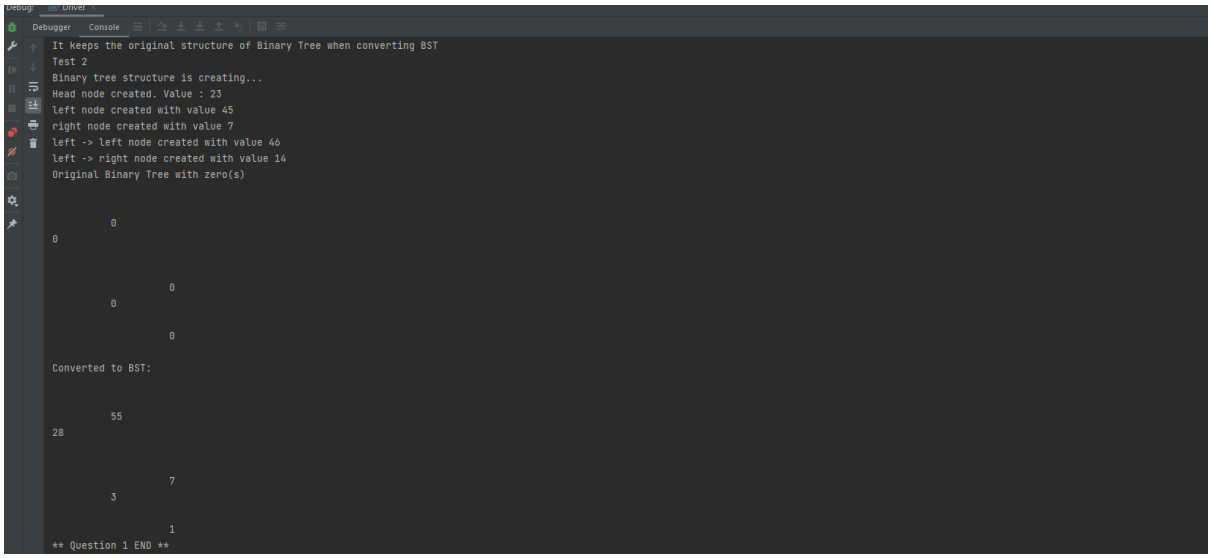
Test 4:

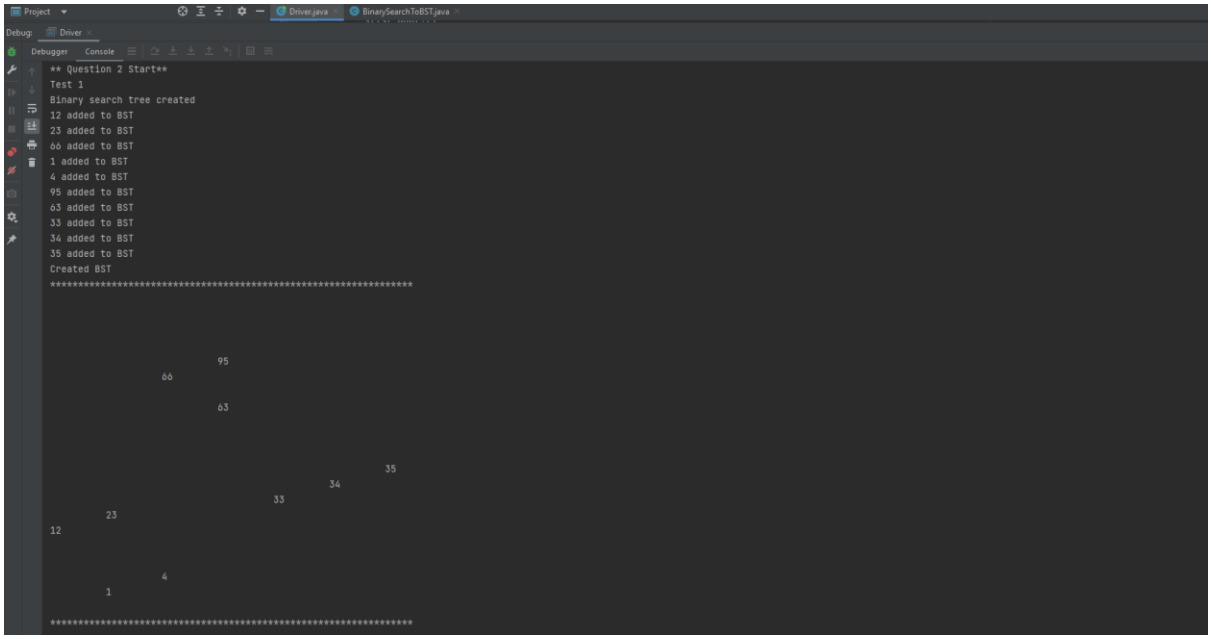
```
System.out.println("Test 2 | extended version of previous one");
System.out.println("Binary search tree created");
bst.add(12);
System.out.println("12 added to BST");
bst.add(23);
System.out.println("23 added to BST");
bst.add(66);
System.out.println("66 added to BST");
bst.add(1);
System.out.println("1 added to BST");
bst.add(4);
System.out.println("4 added to BST");
bst.add(95);
System.out.println("95 added to BST");
bst.add(63);
System.out.println("63 added to BST");
bst.add(33);
System.out.println("33 added to BST");
bst.add(34);
System.out.println("34 added to BST");
bst.add(35);
System.out.println("35 added to BST");
bst.add(12);
System.out.println("12 added to BST");
bst.add(23);
System.out.println("23 added to BST");
bst.add(66);
System.out.println("66 added to BST");
bst.add(1);
System.out.println("1 added to BST");
bst.add(8);
System.out.println("8 added to BST");
bst.add(6);
System.out.println("6 added to BST");
bst.add(156);
System.out.println("156 added to BST");
bst.add(28);
System.out.println("28 added to BST");
bst.add(33);
System.out.println("33 added to BST");
bst.add(38);
System.out.println("38 added to BST");
bst.add(24);
System.out.println("24 added to BST");
bst.add(5);
```

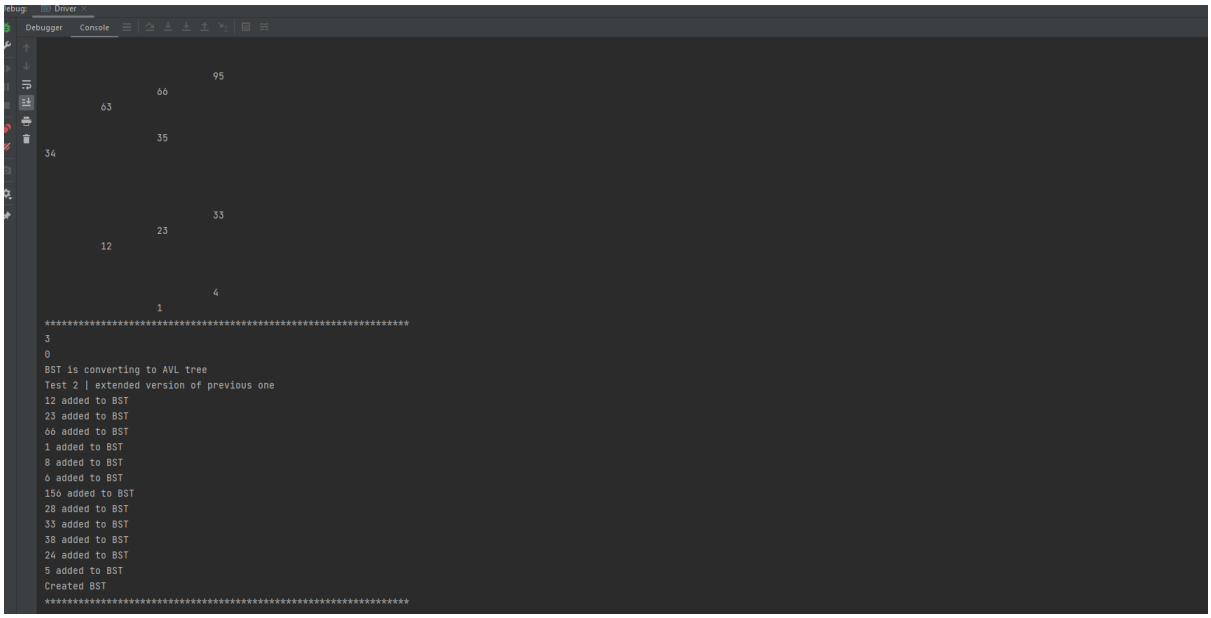
```
System.out.println("5 added to BST");
System.out.println("Created BST");
bst.drive();
System.out.println("BST is converting to AVL tree");
```

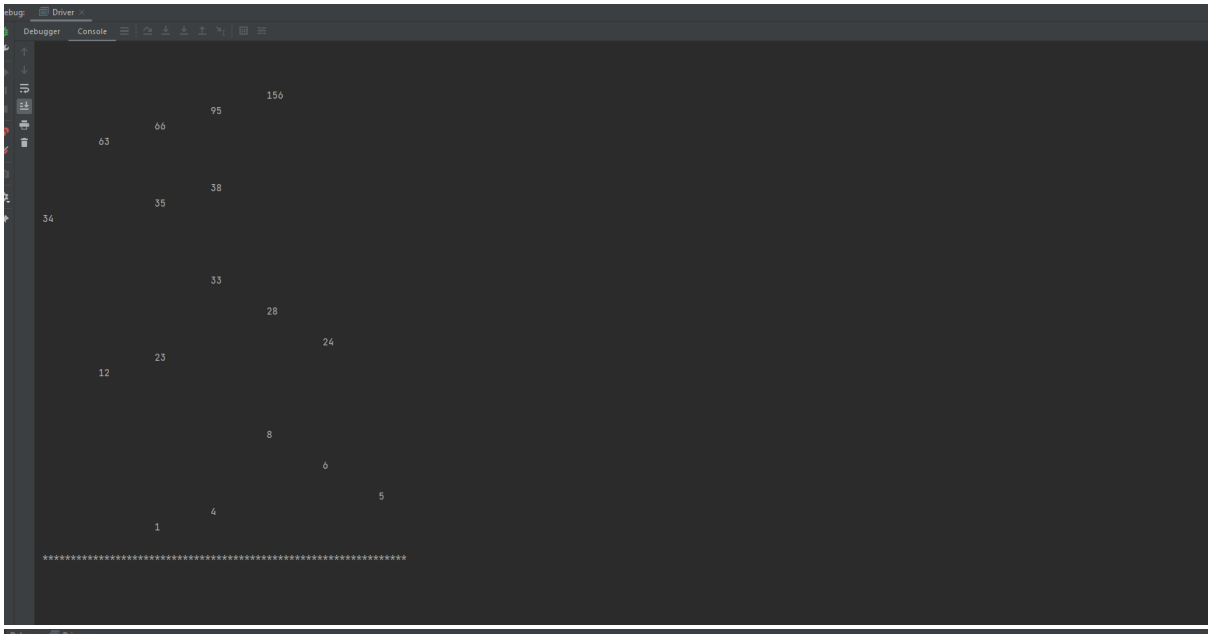
5. RUNNING AND RESULTS

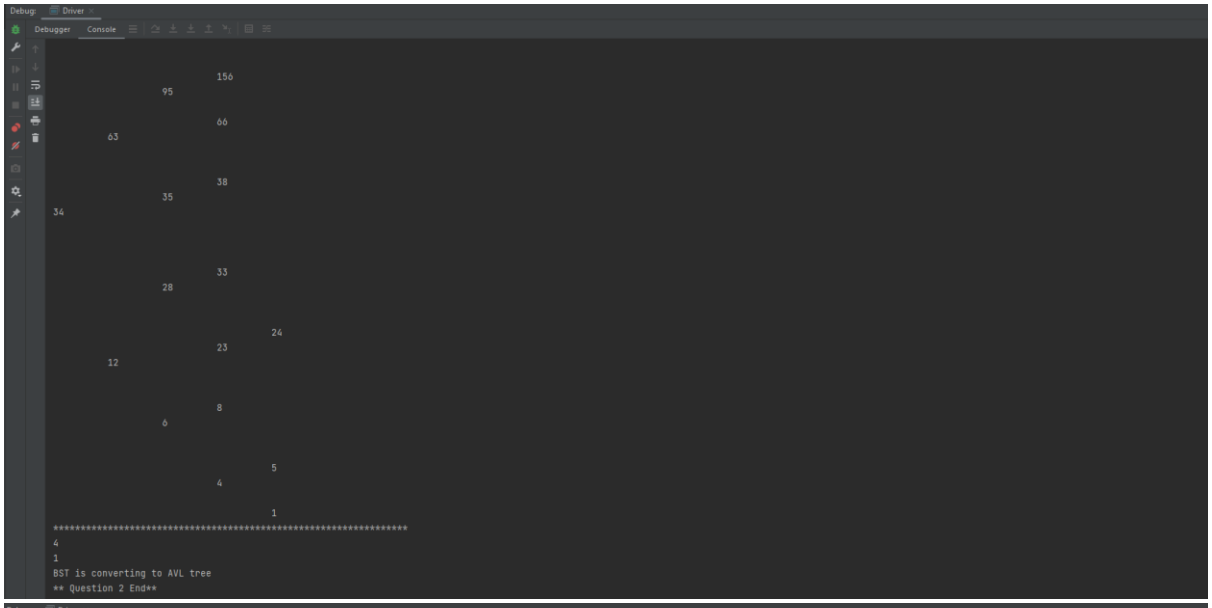


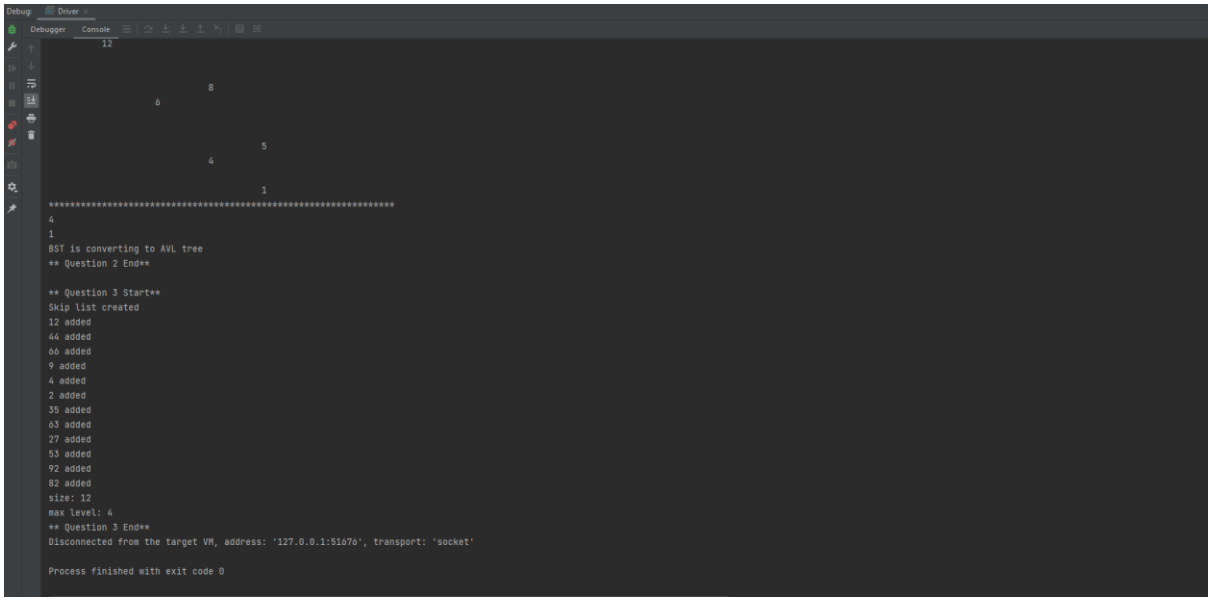



Project
Driver.java
BinarySearchToBST.java
Debug:
Driver
Debugger
Console
** Question 2 Start**
Test 1
Binary search tree created
12 added to BST
23 added to BST
66 added to BST
1 added to BST
4 added to BST
95 added to BST
63 added to BST
33 added to BST
34 added to BST
35 added to BST
Created BST
****
95
66
63
35
34
33
23
12
4
1
****


Debug:
Driver
Debugger
Console
95
66
63
35
34
33
23
12
4
1
****
3
0
BST is converting to AVL tree
Test 2 | extended version of previous one
12 added to BST
23 added to BST
66 added to BST
1 added to BST
8 added to BST
6 added to BST
156 added to BST
28 added to BST
33 added to BST
38 added to BST
24 added to BST
5 added to BST
Created BST
****


Debug: Driver
Debugger Console
156
95
66
63
38
35
34
33
28
24
23
12
8
6
5
4
1
****


Debug: Driver
Debugger Console
156
95
66
63
38
35
34
33
28
24
23
12
8
6
5
4
1
****
4
1
BST is converting to AVL tree
** Question 2 End**


Debug: Driver
Debugger Console
12
8
6
5
4
1
****
4
1
BST is converting to AVL tree
** Question 2 End**
** Question 3 Start**
Skip list created
12 added
44 added
66 added
9 added
4 added
2 added
35 added
63 added
27 added
53 added
92 added
82 added
size: 12
max level: 4
** Question 3 End**
Disconnected from the target VM, address: '127.0.0.1:51676', transport: 'socket'
Process finished with exit code 0