

GTU Department of Computer Engineering
CSE 222/505 - SPRING 2022
HOMEWORK 7 REPORT

BURAK ÇİÇEK
1901042260

System Requirements

```
public BinarySearchTree arrayToBST(BinaryTree inputBinaryTree, E [] inputArray){
```

Method for Q1 to convert array to BST.

```
public boolean add(E item) {
```

 BST add method.

```
private E find(Node < E > localRoot, E target) {
```

 BST Find method.

```
    /** Return value from the public add method. */  
    protected boolean addReturn;  
  
    /** Return value from the public delete method. */  
    protected E deleteReturn;
```

```
protected Node < E > rotateRight(Node < E > root) {
```

 right rotate for AVL.

```
protected Node < E > rotateLeft(Node < E > localRoot) {
```

 Left rotate for AVL.

```
private AVLNode < E > rebalanceRight(AVLNode < E > localRoot) {
```

 Right

Balancer method for AVL

```
private AVLNode < E > rebalanceLeft(AVLNode < E > localRoot) {
```

 Also left

balancer method for AVL.

```
public boolean add(E item)
```

 add method for AVL.

```
public AVLTree BSTToAVL(BinarySearchTree input){
```

 Method for Q2.

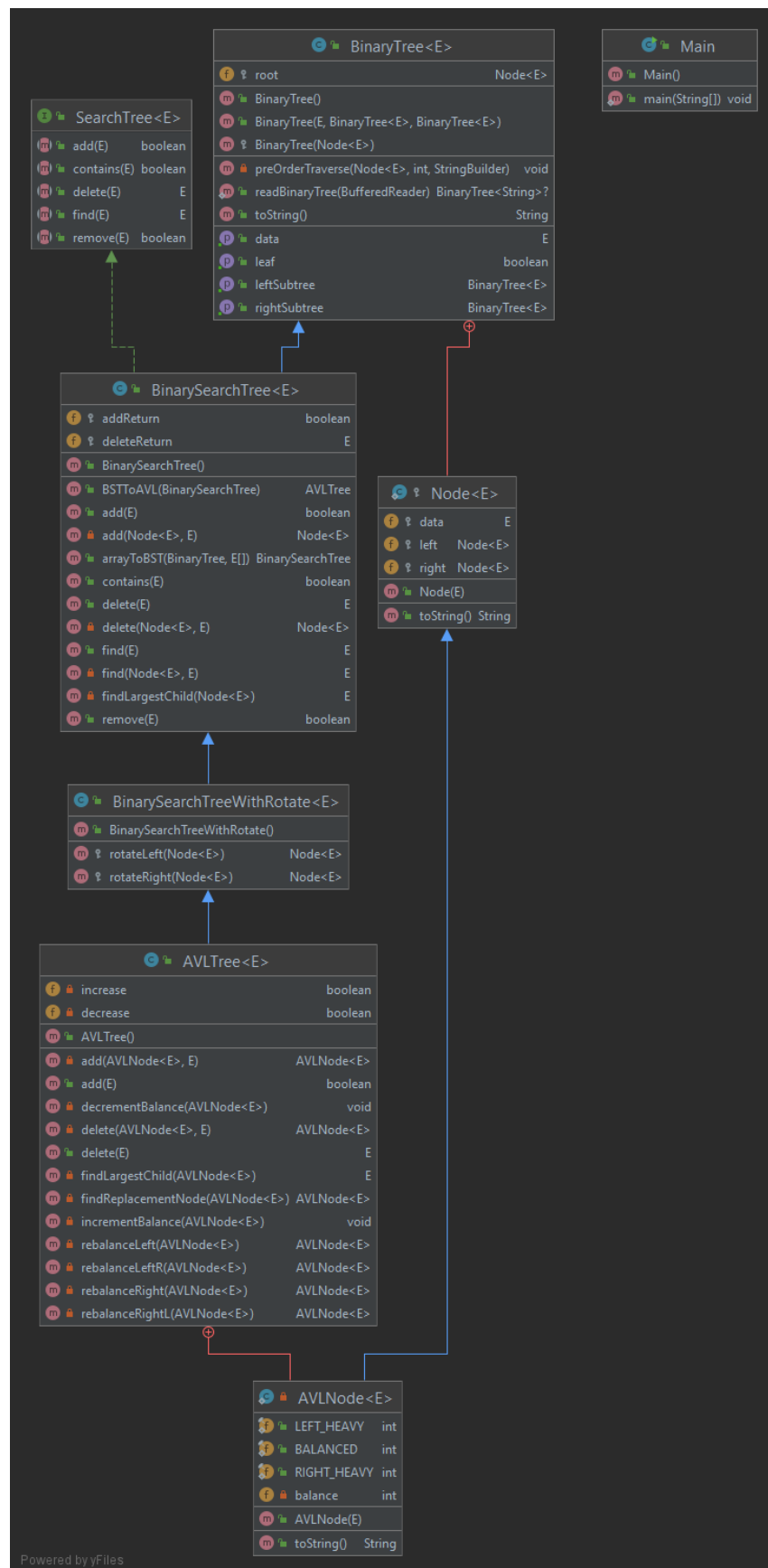
Problem Solution Approach

Speaking for the first question, I had some difficulty in casting and returning. I've never written a method that returns BST as a return, I can say that it forced me a little. Other than that, I had no difficulty in doing the first question.

In the second question, since it uses the left and right rotate methods of AVL, the add method did not need to be written separately or used in my method. Again, as in the first question, returning the AVL Tree made it difficult for me. With casts on it etc. I tried a lot but finally I got the right result. Both of these methods work fine and I can get the desired result.

I print trees with toString to test the accuracy of both results.

Class Diagrams



Test Cases

```
BinaryTree<Integer> myPart1TestBinaryTree = new BinaryTree<Integer>();
Integer testArray[] = {12,5,7,4,3,11,6,8,24,9};
BinarySearchTree<Integer> result = new BinarySearchTree<Integer>();
System.out.println("Print Q1 Method Test Tree");
result = result.arrayToBST(myPart1TestBinaryTree,testArray);
System.out.println(result.toString());
```

```
Print Q1 Method Test Tree
12
 5
  4
   3
    null
    null
  null
 7
  6
   null
   null
 11
   8
    null
   9
    null
    null
  null
24
  null
  null
0: 8
  -1: 6
    0: 4
      0: 3
        null
        null
      0: 5
        null
        null
    0: 7
      null
      null
1: 11
  0: 9
    null
    null
  1: 12
```

```
1: 12
  null
0: 24
  null
  null
```

```
result = result.arrayToBST(myPart1TestBinaryTree, testArray);
//System.out.println(result.toString());
AVLTree<Integer> myPart2AVLTest = result.BSTToAVL(result);
//myPart1TestBST.BSTToAVL(myPart1TestBST, myPart2AVLTest);
System.out.println("Print Q2 Method AVL Test Tree");
System.out.println(myPart2AVLTest.toString());
}
```

Print Q2 Method AVL Test Tree

```
0: 8
  -1: 6
    0: 4
      0: 3
        null
        null
      0: 5
        null
        null
    0: 7
      null
      null
1: 11
  0: 9
    null
    null
1: 12
  null
  0: 24
    null
    null
```

Process finished with exit code 0

```

    BinaryTree<Integer> myPart1TestBinaryTree = new BinaryTree<Integer>();
    Integer testArray[] = {-7,0,1,4,9,13,5,6};
    BinarySearchTree<Integer> result = new BinarySearchTree<Integer>();
    //System.out.println("Print Q1 Method Test Tree");
    result = result.arrayToBST(myPart1TestBinaryTree,testArray);
    //System.out.println(result.toString());
    AVLTree<Integer> myPart2AVLTest = result.BSTToAVL(result);
    //myPart1TestBST.BSTToAVL(myPart1TestBST , myPart2AVLTest);
    System.out.println("Print Q2 Method AVL Test Tree with another array");
    System.out.println(myPart2AVLTest.toString());

```

Print Q2 Method AVL Test Tree with another array

```

1: 4
  0: 0
    0: -7
      null
      null
    0: 1
      null
      null
1: 6
  0: 5
    null
    null
  1: 9
    null
    0: 13
      null
      null

```

Process finished with exit code 0

```

    BinaryTree<Integer> myPart1TestBinaryTree = new BinaryTree<Integer>();
    Integer testArray[] = {-7,0,1,4,9,13,5,6};
    BinarySearchTree<Integer> result = new BinarySearchTree<Integer>();
    System.out.println("Print Q1 Method Test Tree with another array.");
    result = result.arrayToBST(myPart1TestBinaryTree,testArray);
    System.out.println(result.toString());

```

Print Q1 Method Test Tree with another array.

-7

 null

 0

 null

 1

 null

 4

 null

 9

 5

 null

 6

 null

 null

 13

 null

 null

Process finished with exit code 0