COMP 2313 DATA STRUCTURES CODE ASSIGNMENT

|  |  |
| --- | --- |
| ***Assignment Name:*** Create an array-based implementation of a binary tree | ***Student Name :****Mustafa S Topsakal* |
| ***Assignment Date:*** *10/30/2020* |  |

# Problem

# Create an array-based implementation of a binary tree

# Challenges

* Understanding the logic of binary tree
* Implementing a recursion func to reach nodes

# Code

public class Node {  
 //Those nodes are the places that we conenct datas and implement our datas //  
 public Integer data;  
 public Node leftChild;  
 public Node rightChild;  
  
 //main constructor if nothing implemented at first//  
 public Node(){  
 data = null;  
 leftChild = rightChild = null;  
 }  
  
}

public class TreeArray {  
 //I chose 64 for max size because it should be 2^n//  
 public static final int *size* = 64;  
 public Integer[] arr;  
  
 public TreeArray(){  
 arr = new Integer[*size*];  
  
 //this for loop initialize the array null at first if the user didnt implement anything at first  
 //also for further adding and removing  
 for(int i= 0; i <arr.length;i++){  
 arr[i] = null;  
 }  
 }  
 public void insert(Integer data){  
 //checksd if the array is empty//  
 if(arr[0]==null){  
 arr[0]=data;  
 return;  
 }  
 // otherwise we will use other func to add datas //  
 add(0,data);  
  
 }  
 public void add(int root, Integer data){  
 //if the root is null which means we cant add so just return;  
 if(arr[root]==null){  
 return;  
 }  
 // with this if, we can reach the left child because data is smaller than root //  
 if(data<arr[root]){  
 if(arr[1+ root\*2]==null){  
 arr[1+ root\*2] = data;  
 return;  
 }  
 else{  
 add(1+root\*2,data);  
 }  
 }  
 else{  
 //in this else, we will reach the right child//  
 if(arr[2+ root\*2]==null){  
 arr[2+ root\*2] = data;  
 return;  
 }  
 else{  
 add(2+root\*2,data);  
 }  
 }  
  
 }  
 public String toString(){  
 String temp = "";  
 for(int i=0;i<arr.length;i++){  
 if(arr[i] != null){  
 temp = temp + arr[i] + " ";  
 }  
 }  
 return temp;  
 }  
  
}

public class Main {  
 public static void main(String args[]){  
 TreeArray tree = new TreeArray();  
 tree.insert(0);  
 tree.insert(1);  
 tree.insert(2);  
 tree.insert(5);  
 tree.insert(3);  
 tree.insert(8);  
 tree.add(2,7);  
   
  
 System.*out*.println(tree);  
 }  
}

# Pseudo Code

Class Node

Implement all the variables

Data, leftchild , rightchild

Node()

Initialize all of them as null

Class TreeArray

TreeArray()

Set the array size to 64

For(make all of them null)

Void insert(takes varaiable)

If array is empty, first element = 0

Called the add function

Void add (take root and data)

If(root= null)

Return nothing

If(the data we want to add < root)

We add data to 1+root\*2 which is left node of the root

Else

Add(called the function with same data to implement)

Else

We add data to 2+root\*2 which is right child of the door

Than called the add fucn with same variable

ToString

Print all of them one by one

# Outputs

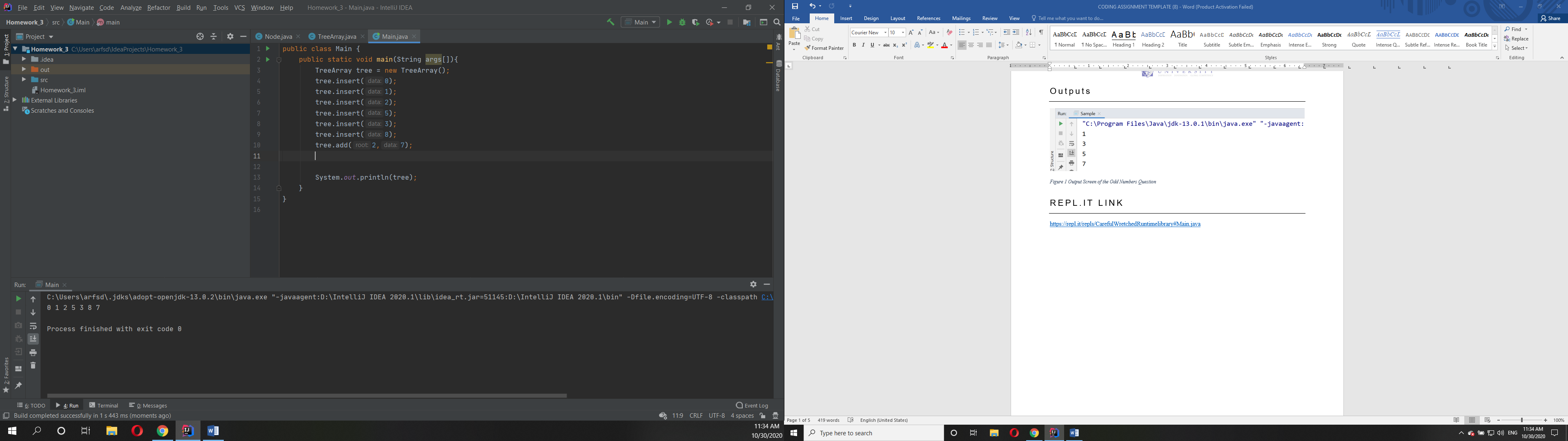


Figure Output Screen of the Odd Numbers Question