Gebze Technical University
Computer Engineering

CSE 222 - 2018 Spring

HOMEWORK 6 REPORT

EBRU KARDAŞ
141044049

Course Assistant: Fatma Nur Esirci

1 Worst RedBlack Tree

1.1 Problem Solution Approach

1.2 Test Cases

1.3 Running Commands and Results

2 binarySearch method

2.1 Problem Solution Approach

Classic binary search algorithm is used but user gives just target. A helper method searches for the target if it is in the tree. In the code, children are not taken as a list. Links are shown as left and right reference so binary search is like in binary tree. In pdf, we don't have to take Btree code from the book so algorithm is different from in the book.

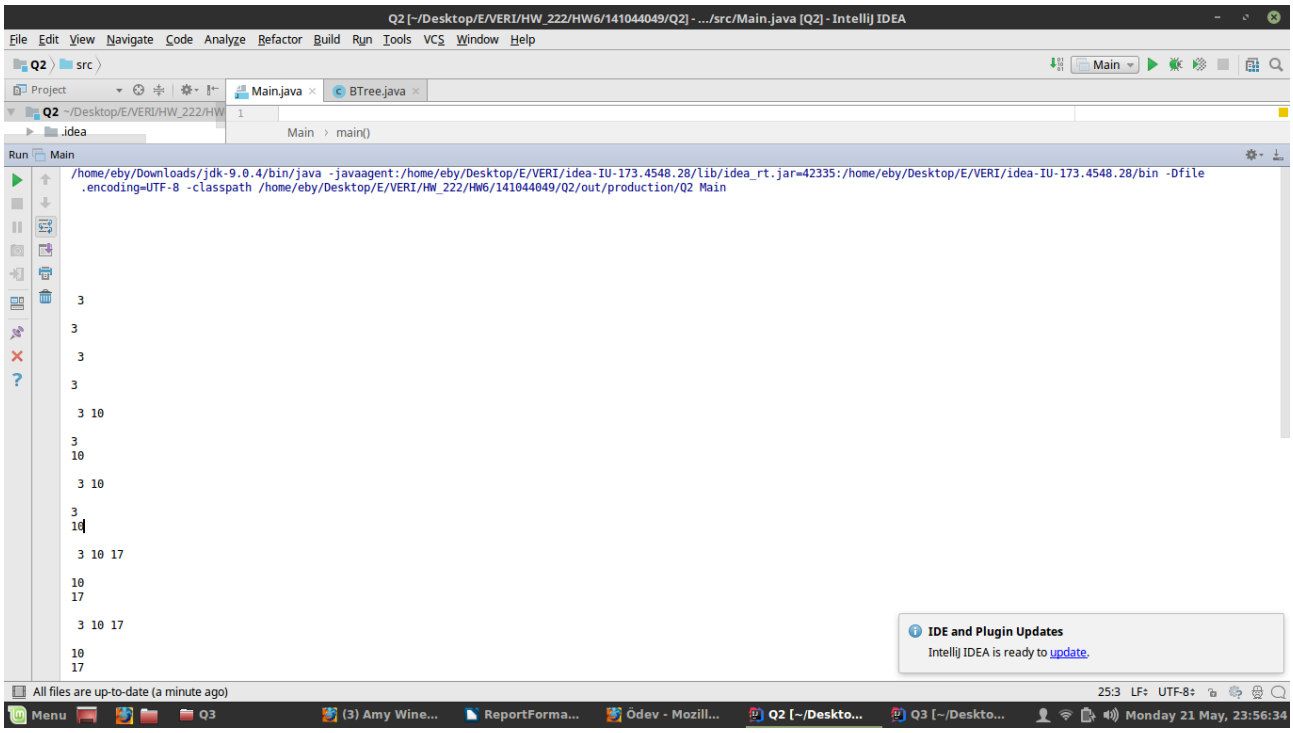
Method binarySearch(item)
call binarySearch(item, root)

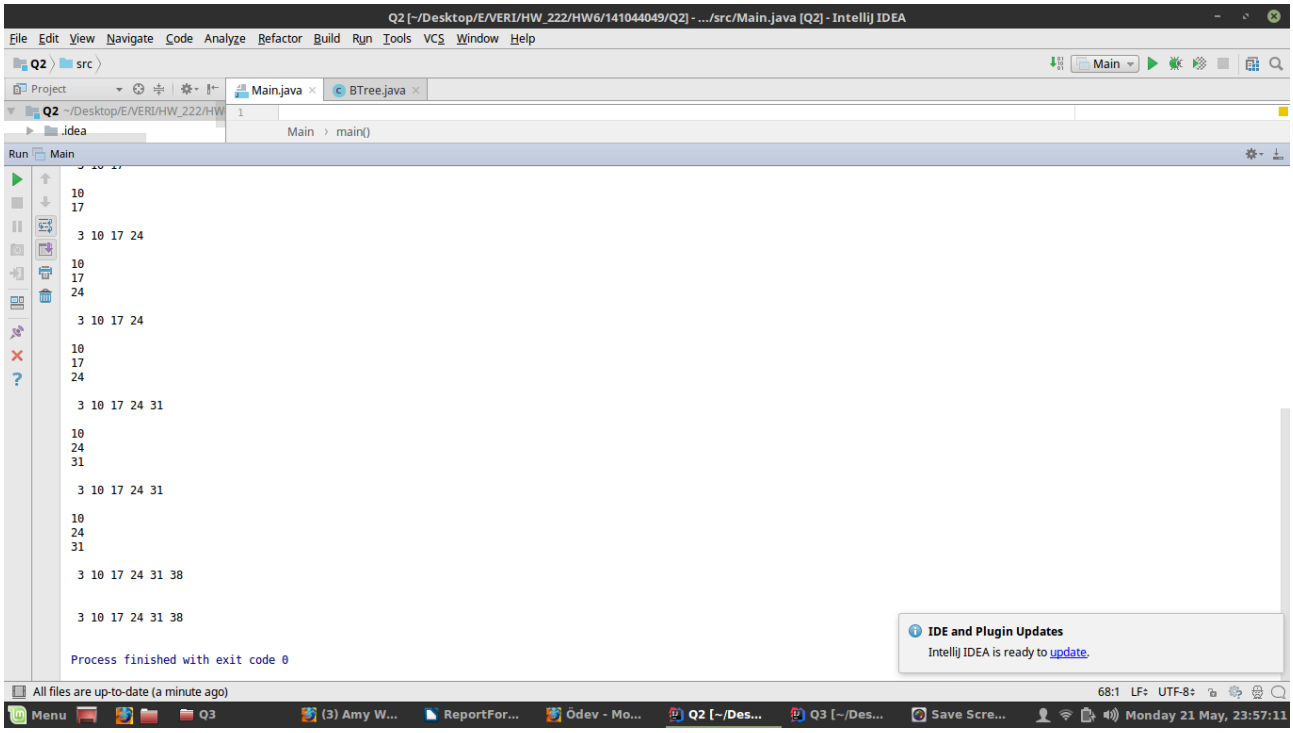
Method binarySearch(item, node)
var returned is false
if it is null, returns false
if node data is equal to item, return true
if item is less than node data, binarySearch(item, node.left)
else (item is less than node data) binarySearch(item, node.right)
Btree is taken from internet (jbixbe.com/doc/tutorial/Btree).

2.2 Test Cases

All these in Main.java

2.3 Running Commands and Results





3 Project 9.5 in book

AVL Tree which should be completed is implemented. Code is taken from given link in homework pdf but some properties was missing or not complete. Removal method(s) was missing and decrementBalance, incrementBalance, rebalanceLeft, and rebalanceRight methods were not complete. Also a new constructor that takes a binary tree and checks whether this tree is AVL tree or not is wanted.

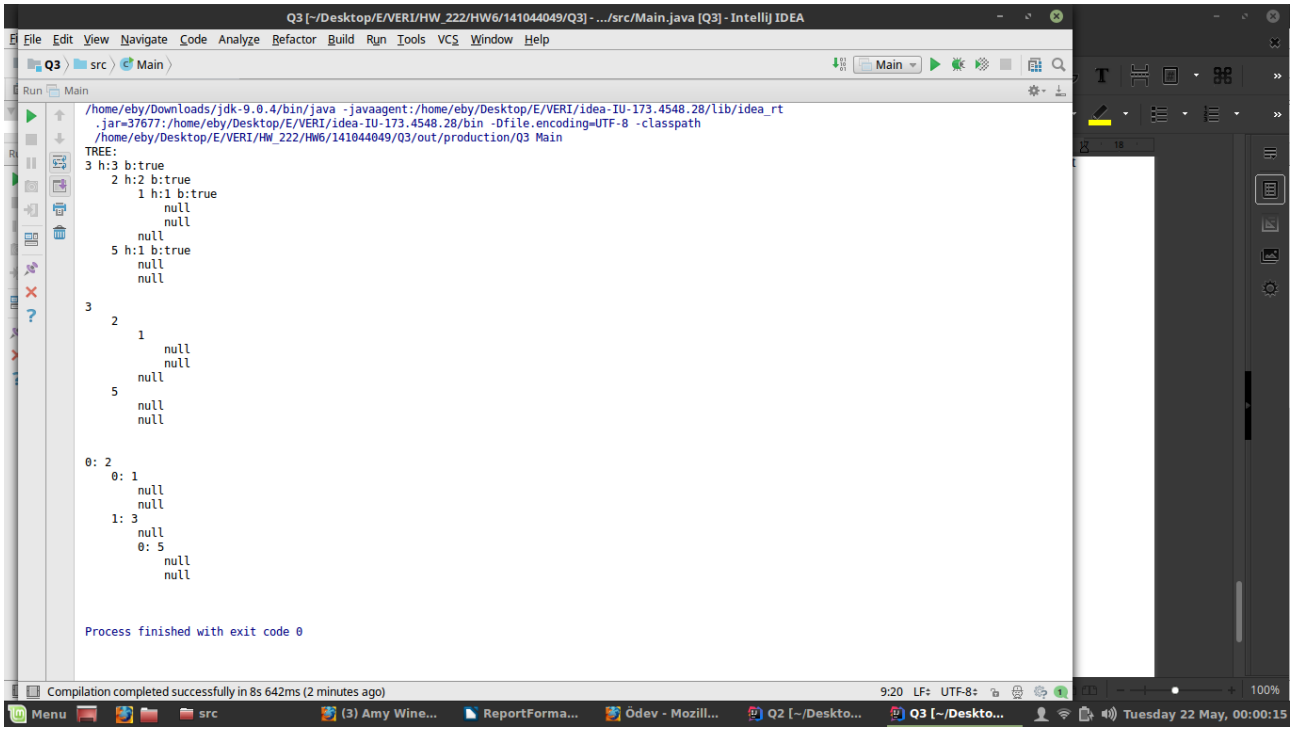
3.1 Problem Solution Approach

For remove an item, we should create a boolean member variable named decrement like used for addition. DecrementBalance and rebalanceLeft are same but incrementBalance and rebalanceRight methods are a little changed for ‘decrement’ variable.

For constructor that takes BinaryTree and checks if AVL Tree or not, two new methods are used. One of these find the height of the tree. Other method checks recursively if tree and its subtrees balanced which means height is less or equal to 1.

Deletion and rotate methods doesn’t run always as it should be but constructor works an checks is it is AVL (balanced) tree. Code is taken from book (given in the link)

3.2 Test Cases

In Main.java

3.3 Running Commands and Results