

PROGRAMMING ASSIGNMENT 3

Binary Tree

Introduction

In this assignment you will implement a program to create a binary tree that can be used to easily search for values.

Binary Tree

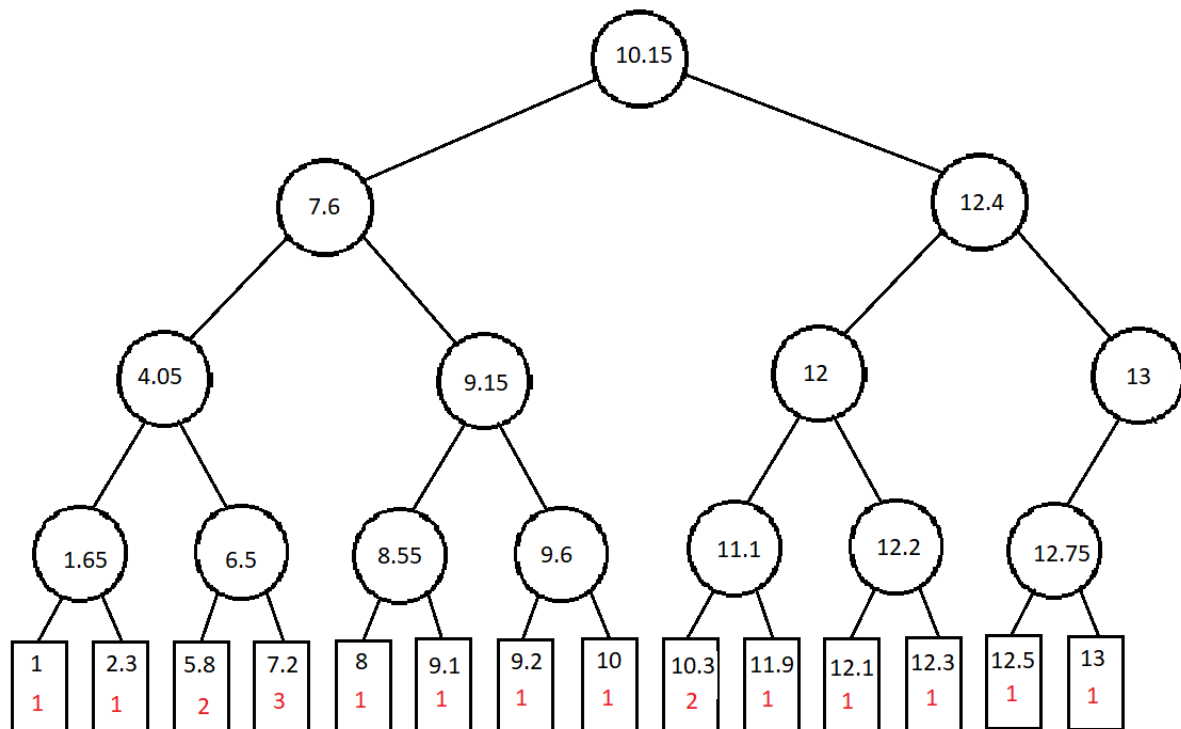
The data stored in the binary tree are `doubles`. The binary tree will store the data **only on the leaves**. The internal nodes are used for searching only. Your tree will work like a normal BST, to find a target element, your program will compare it to the root of the current subtree and if the target is less than the value of the root you will recursively search on the left subtree, otherwise, you will search on the right subtree. Your tree will allow duplicate values, by keeping track of the number of times that a particular value appears (this data is kept only in the leaves).

Your binary tree is always created from scratch given a sorted array of `doubles`, in such a way that the root of every subtree (not a leaf) contains the average of the maximum element in the left subtree and the minimum element in the right subtree.

Example:

Suppose that you are given the following array:

1, 2.3, 5.8, 5.8, 7.2, 7.2, 7.2, 8, 9.1, 9.2, 10, 10.3, 10.3, 11.9, 12.1, 12.3, 12.5, 13



The tree that is generated has a root with a value of 10.15, which is the average of the maximum value of the left subtree (which is 10) and the minimum value of the right subtree (which is 10.3), so the root has a value of $(10+10.3)/2 = 10.15$

The leaves have the actual values and the number of times each value appears.

Downloading Instructions

Download the *assignment3.jar* file, it contains a “project” that you must import into Eclipse so that you can start working on your programming assignment.

First download the *assignment3.jar* file to your computer.

In Eclipse select *File -> Import -> JavaEE -> App Client JAR file*.

Then click on Next and use (*Application Client file*) Browse to find the file that you downloaded.

In *Application Client Project* write down the name of the new project that you will create in Eclipse. Press Finish

Implementation and Grading

You will implement the following methods in the *assignment3* class. The instructions of what to do are in the source files (comments):

- **[40 points]** `public static Queue<BTNode<dataNode>> makeQueue(double[] a)`
- **[40 points]** `public static Queue<BTNode<dataNode>> join(Queue<BTNode<dataNode>> myQueue)`
- **[20 points]** `public static int search(BTNode<dataNode> root, double target)`

What to turn in

In Eclipse select *File -> Export -> Java -> JAR file*. Then select the Project that you want to export (the project that you have been working on), on the options below, select *Export Java source files and resources*, and also select *Compress the contents of the JAR file*. After selecting the destination of the JAR file, click on *Finish*.

The file that you just created (above) must be submitted on Sakai. Make sure that your project compiles and works correctly before turning it in. A project that does not compile will receive a grade of 0.