

COMP251 - Lab5

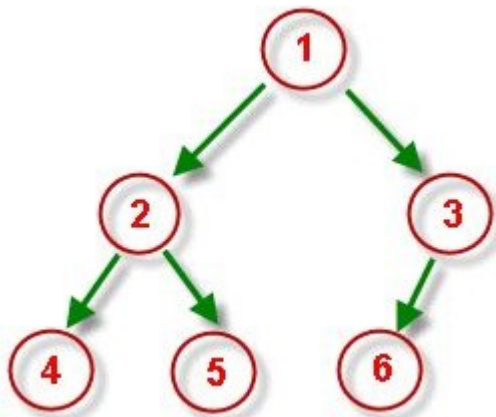
Goal: This lab will give you practice with binary trees.

Getting Started

You can have a partner for this lab, but submit individually. You can read the lectures to answer the question.

Download the files for this lab from blackboard (Week 8). Classes `BinaryTree` and `BinaryNode` implement a binary tree. Create an empty project in Eclipse and call it 'BinaryTree'. Add classes: `BinaryTree`, `BinaryNode`. The `main()` method of `BinaryTree` includes test code. Read `BinaryTree.java` and `BinaryNode.java` to find out what methods are available.

The `test1()` method in `BinaryTree` creates a tree like this:



1. Write a function `test2()` that generates a tree with 7 nodes (including root) minimum height.
2. Write a function `test3()` that generates a tree with 7 nodes (including root) maximum height.
3. Implement `printPostOrder()` for `BinaryTree` (you need to define the helper function in `BinaryNode`)
4. Implement `printPreOrder()` for `BinaryTree` (you need to define the helper function in `BinaryNode`)

Optional (for more practice only)

5. Implement height() for BinaryTree (you need to define the helper function in BinaryNode).
6. Write a method for BinaryTree class to check if the tree is perfect or not.

This is the signature of the method that you should add:

```
public Boolean isPerfect();
```

Hint: the recursive definition of perfect trees is useful to implement this method.

7. A binary tree is said to be "**balanced**" if both of its subtrees are balanced and the height of its left subtree differs from the height of its right subtree by at most 1. Write a method for BinaryTree class to determine whether the binary tree is balanced. This is the signature of the method that you should add:

```
public Boolean isBalanced();
```

Note that the definition of being balanced is recursive like many other definitions in binary trees.

8. Given the in-order and pre-order traversals of a tree (following), draw the tree.

Pre-order: A B C D E F G H I J K M

In-order: D C B F E G A J I K H M

Submission

In addition to submission of the files for question 1-4, you need to record a video and explain your answer.

Submit the link to your video clip (roughly about 4min, not more than 5min) that **explains clearly your approach, run your code, and demonstrate your solution.** You could provide the link to your video (on Youtube/Vimeo/...) in the body of your submission.