CMPT225, Spring 2023
Homework Assignment 3
Due date: Sunday, March 5, 2023, 23:59

You need to implement the following classes:

- *binarytree.BinaryTree*
- *binarytree.BTNode*

Note that all files must be under the correct package (folder)**.**
You may add more classes to your solution if necessary.

Submit all your files in ***assignment3.zip*** to CourSys.
Make sure your zip file can be unzipped using the command "*unzip assignmen3.zip*" in CSIL.
The zip file needs to contain exactly one folder: ***src***.
In the ***src*** folder there must be one folder/package: ***binarytree***.
The ***binarytree*** folder needs to contain the following files (and any additional files if needed):

- src/*binarytree/BTNode.java*
- src/*binarytree/BinaryTree.java*

Discussion with others: You may discuss the assignment with your classmates/tutors (or anyone else), but coding must be entirely your own.

References: You may use textbooks, wiki, stack overflow, geeksforgeeks, etc. If you do, specify the references in comments. Posting questions online asking for solutions (e.g. using chegg.com) is prohibited.

Readability: Your code should be readable using the standard Java conventions. Add comments wherever is necessary. If needed, write helper functions or add classes to improve readability.

Compilation: Your code MUST compile in CSIL using javac.
Make sure that your code compiles without warnings/errors.
If the code does not compile in CSIL the grade on that part will be 0 (zero).
Even if you can't solve a problem completely, make sure it compiles.

The assignment will be graded mostly **automatically**, with some exceptions.

Do not add main() to your solutions. The main() method will be in the test files.

Warnings: Warnings during compilation will reduce points.
More importantly, they indicate that something is probably wrong with the code.

Testing: Test your code. Examples of tests are included. Your code will be tested using the provided tests as well as additional tests. You should create more tests to check your solution.

Good luck!

Write the following methods in the class BinaryTree.

a) *[10 points]* **int** numberOfLeaves()
The method returns the number of leaves in the tree.

b) *[15 points]* **int** countDepthK(**int** k)
The method returns the number of nodes in the tree with depth=k.
If k is greated than the depth of the tree, the method returns 0.

c) *[15 points]* **void** map(Function<? super T, ? extends T> mapper)
The method gets a mapper, and for each node in the tree applies it to the node.data.
In order to apply the mapper on the data in a node you may use the line
node.setData(mapper.apply(node.getData()))
For the API see https://docs.oracle.com/javase/8/docs/api/java/util/function/Function.html

d) *[15 points]* List<BTNode<T>> pathFromRoot(BTNode<T> node)
The method returns the path from the root to the given node.
If the node is not in the tree, the method throws IllegalArgumentException

e) *[15 points]* **int** distance(BTNode<T> node1, BTNode<T> node2)
The method returns the distance from node1 to node2 in the tree.
If one of the nodes is not in the tree, the method throws IllegalArgumentException

f) *[30 points]* **public** Iterator<T> preOrderIterator()
The method returns an iterator that performs the preorder traversal on the tree.

The iterator must be dynamic in the following sense: if after the iterator is created, and the tree changes in some part that has not been processed by the iterator yet, then the iterator "will see" these changes and output the values in the updated tree when reaching that part of the tree. However, if you change the nodes that have already been processed, then it will have no effect on the iterator.
** Be careful, do not change the nodes that are currently being processed, as this may have unexpected effects

This means you cannot simply create the list with the preorder traversal of the tree in the constructor of the iterator. It will not work in the dynamic sense.

You will probably need to implement the iterator in a new class. Please declare the class in the package ***binarytree***. Make sure to submit the java file implementing the iterator.