CSCI3200 Spring 2023 LAB3 --BinarySearchTree

***Objective***: In this lab project, the student will demonstrate their understanding on the reference-based implementation of a Binary Search tree. After completing this project, the students should be able to:

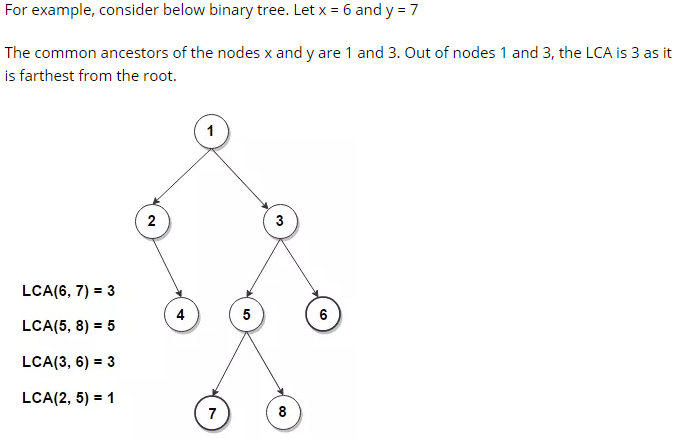
* Confidently implement binary search tree using reference-based implementation
* Comfortable in implementing binary search tree operation with three reference variables only: element, left, right (no direct reference to its parent. This will help the students better adjust their learning to popular interview questions they may have to deal with in their future interviews.)
* Implement varies operations on binary search tree based on the requirement
* Comfortable in implementing required methods using recursion

You are allowed to work with a partner in one of your remaining LABs (only one) if you have not used it. If you choose to work as team, all team members are expected to contribute to the completion of the LAB. You will receive a deducted grade if your team member complains about your contribution.

1. Download the compressed folder LAB3 from D2L. There are three “.java” files in this folder, including:
   1. Node.java: this class defines the node structure, no modification needed
   2. BinarySearchTree.java: this is the class you are going to work on. Please based on the requirement to implement each method.
   3. mainClass.java: based on your implementation of the binary search tree, complete this class to test your methods. You are required to test each method at least once. Starting by randomly inserting three more nodes with no duplicate.
2. The following question is a very popular interview question. You are encouraged to challenging yourself. This question will **not** be graded.

***public Node findLCA(Node p1, Node p2)***: find lowest common ancestor (LCA) of p1 and p2 in it.

<http://www.techiedelight.com/find-lowest-common-ancestor-lca-two-nodes-binary-tree/>



1. Remark:
   1. The implementation of the methods either covered in our in-class practice or can be found on the slides. Feel free to contact the instructor if you see a need to change any of the method signature.
   2. If you want to add extra methods to support the implementation of the required methods, please put them also at the end of the file for clarity.
   3. If you choose to work with a partner, make sure your team members name can be clearly identified via submission

**What to Submit (one submission per team is good enough):**

• BinarySearchTree.java

• mainClass.java