Intermediate/Advanced University Computer Science

ICS4U-Grade 12

What we have begun, we shall finish

Unit 4 Assignment*:* Recursion

Name:____ **Date:** April 18, 2018

Category	**A**	**C**	**Mark**
Marks	/8	/4	**/12**
Weight	*40%*	*20%*	***100%***

Marks have been normalized according to category weights

Part B. *This assignment replaces Part B of Unit 4 Quest of March 8, 2018*

Due date

- Tuesday April 24, 2018 start of class
- Late submissions lose 5% per day to a maximum of 20%
- No assignments may be submitted after Friday April 27, 2018

Hand-in

- A paper, plain English written or printed summary of the testing program
- A program file with student's name on it to be dropped in the DropOff folder

Scenario

We have tested the deleting of nodes from a binary tree by deleting one node at a time for each of cases 1 through 5 as given in the diagrams of

`Course Notes Unit 6 Indices Eserver Pseudo Code 20a.pdf`

Our code responds well with these very small trees that we have used. This is like doing testing in a laboratory. The purpose of this assignment is to put a strain on the `insertNode()`, `deleteNode()`, and `findNode()` methods that better simulates realistic circumstances. In doing so we will have higher confidence that our methods are robust and are able to take heavier loads.

(next page)

Task

Design a testing program that makes use of the three methods `insertNode()`, `deleteNode()`, and `findNode()` as needed. The constraints are:

1. The program grows and shrinks a binary tree continuously by adding and removing large amounts of nodes (in the tens of thousands) with random unique identifications for at least 15 seconds uninterrupted.
2. The first 50 nodes that get added to the tree must not be deleted at all throughout the entire test.
3. Somewhere in the process, the program must have had to delete at least 400 nodes and have reached at least a size of 8000 nodes.
4. After reaching the size of at least 8000 and having run for at least 15 seconds, the program adds or deletes nodes as needed so that the tree shrinks to exactly 50 nodes.
5. If the process is successful, then the remaining 50 nodes will be the first 50 nodes that got added to the tree at the start of the process. Use the `findNode()` method to test that all those 50 nodes are in the tree (though most likely not in the same location)
6. Include at least one other constraint of your own that will be of help for the testing of the `deleteNode()` code, or modify any of the other constraints in such a way that the testing is improved.

Be sure to read these 3 notes:

This assignment is meant to test the methods of the binary tree. Marks will be awarded primarily for creative thinking and testing, especially constraint 6. Any shortcuts used so as to only meet the bare minimum requirements of the constraints resulting in a poorly designed test will loose marks.

Some of the operations may be time consuming. At this point we are more concerned with the correctness of the result than with efficiency of the testing. The testing program may take minutes to run. As long as it runs within reasonable time (less than 15 minutes) we are ok.

Some of the constrains may not necessarily happen in the sequence given above (except possibly constraint 5). Use your judgment on this and feel free to move them around if it helps the ultimate goal of becoming more confident that the tree methods work properly.