SPECIFICATIONS

DASTRUC 3rd MACHINE PROBLEM (MP2)

33% of the Final Project Grade

I. INTRODUCTION

This document contains the specifications of the 2nd Machine Problem. **MAKE SURE THAT YOU UNDERSTAND AND FOLLOW THE SPECIFICATIONS PROPERLY. FAILURE TO DO SO WILL AUTOMATICALLY FAIL THIS MP.**

II. OBJECTIVES OF THE MP

a. To learn how to implement and apply the trees data structure concept in Python.

III. PROBLEM DESCRIPTION

A. INPUT

The input is a text file containing a sequence of integer values that will be inserted (in the order that they appear in the text file) into an initially empty Binary Search Tree (BST).

We use as example the following data values as sample file contents:

500
250
700
100
400
600
300
650

B. SIMPLIFYING ASSUMPTIONS

We make the following assumptions to simplify our lives :-) /* actually, your life */

a. The contents of the text file are valid values.
b. Each value is unique (that is no two lines will contain the same value).
c. We do not know in advance the number of nodes (vertices) stored in the text file.

C. PROCESSING and OUTPUT

Your program should ask first for the name of the text file, for example:

Input the name and extension of the file: DATA.TXT

Your program should read each value stored in the data file, and insert it as a new node into an initially empty BST. You are NOT required to graphically display the BST. If the contents of "DATA.TXT" correspond to sample data values above (see Section II-A INPUT), the resulting BST will be:

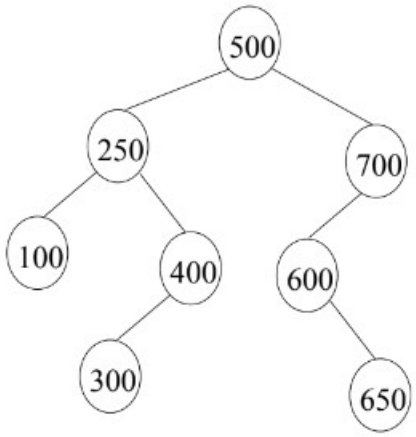

Figure 1. Sample BST

The required output of your program are:

a. Traversals in the following sequence
   i. Pre-order traversal (output in a line, use space/s to separate values)
   ii. Post-order traversal (output in a line, use space/s to separate values)
   iii. In-order traversal (output in a line, use space/s to separate values)

Using the sample BST in Figure 1, the output (of the traversals) will be:

500 250 100 400 300 700 600 650
100 300 400 250 650 600 700 500
100 250 300 400 500 600 650 700

b. A table indicating information (arranged into seven columns):
   i. Node
   ii. Node's parent
   iii. Node's sibling
   iv. Node's left child
   v. Node's right child
   vi. Node's degree
   vii. Node's depth

The number of rows correspond to the number of nodes. The node's in the first column are listed in ascending order (i.e. from lowest to highest (hint: In-order traversal).

Using the sample BST in Figure 1, the output will be:

```
100   250    400    NULL   NULL   0   2
250   500    700    100    400    2   1
300   400    NULL   NULL   NULL   0   3
400   250    100    300    NULL   1   2
500   NULL   NULL   250    700    2   0
600   700    NULL   NULL   650    1   2
650   600    NULL   NULL   NULL   0   3
700   500    250    600    NULL   1   1
```

The first line (i.e. first row) of output means that the node is 100, its parent is 250, its sibling is 400, its left child is NULL, its right child is NULL, its degree is 0 and its depth is 2.

The fifth line (i.e., fifth row) of output means that node is 500, its parent is NULL (which means 500 is the root node), its sibling is NULL, its left child is 250, its right child is 700, its degree is 2 and its depth is 0.

IV. REQUIREMENTS

You are required to implement the following on your program.

1. Based on problem description above, your output should be similar on the sample output (shown in blue color) above.
2. You program should execute from the command line (CMD) in Windows.

V. DELIVERABLES AND SUBMISSION DEADLINE

You need to submit **one (1)** item, namely:

a. Your Python script for the main script (filename should be <*lastname*>_main.py)

It is the student's responsibility to check the solution **(several times!)** for syntax and semantic errors before submission. Once submitted, no replacement will be accepted.

The source code should be **RECEIVED** via e-mail as a .zip file (along with the other MP files) before **June 19, 2023, 12:00 noon**. The recipient of your e-mail should be one of the following, depending on your DASTRUC professor. Make sure to CC your and your groupmate's e-mail addresses before sending the e-mail.

IT223, IT225 - John Raymound Javier: jljavier@nu-fairview.edu.ph

IT221, IT222 and IT224 - Jian Michael Wu: jowu@nu-fairview.edu.ph

The e-mail subject should be:

DASTRUC_<section>:
[space]<LASTNAME_MP1>[space]<FIRSTNAME_MP1>[space]MP1_<LASTNAME_MP2>[space]<FIRSTNAME_MP2>[space]MP2_<LASTNAME_MP3>[space]<FIRSTNAME_MP3>[space]MP3

where:

- LASTNAME_MP1 and FIRSTNAME_MP1 = the one who was assigned MP1
- LASTNAME_MP2 and FIRSTNAME_MP2 = the one who was assigned MP2
- LASTNAME_MP2 and FIRSTNAME_MP3 = the one who was assigned MP3
- You should replace [space] with " " (one space bar press)

Ex. DASTRUC_IT123: PEÑA_RYAN_MP1_CANLAS_JOMARY_MP2_LO_HANS_MP3

INCOMPLETE SUBMISSION and **NON-COMPLIANCE** with the instructions and specifications will automatically result in point deductions.

LATE SUBMISSIONS WILL BE ACCEPTED but with a deduction of ONE point for every minute or a fraction thereof in fairness to all students who submitted on time.

VI. CHECKING, TESTING, AND GRADING SCHEME

Your Python Scripts will be compiled and tested. Note that by default you will receive a perfect grade of 100% (for this MP only). It will remain as 100% if there are no warnings, syntax errors, and semantic errors; and that you complied properly with all the MP specifications. **A program with a syntax error will automatically be given a grade of ZERO (0).**

The contribution, in percentage, for each of the required functions are indicated in the table below:

Requirements	Percentage
INSERTION OF NODE IN A BST	10
PRE-ORDER TRAVERSAL	10
POST-ORDER TRAVERAL	10
IN-ORDER TRAVERAL	10
DETERMINING A NODE'S PARENT	10
DETERMINING A NODE'S SIBLING	10
DETERMINING A NODE'S LEFT CHILD	10
DETERMINING A NODE'S RIGHT CHILD	10
DETERMING A NODE'S DEGREE	10
DETERMING A NODE'S HEIGHT	10

** Please see your professor if you have any questions regarding this MP **