PROG2121 – WINDOWS PROGRAMMING

A03 – DATA STRUCTURES

OVERVIEW

Data structures are an integral part of most computer programs. Many popular structures are already implemented in the .NET Framework.

In this assignment, you will design tests to measure the performance of some data structures.

OBJECTIVES

- Implement various .NET supported Data Structures in desktop applications.

ACADEMIC INTEGRITY AND LATE PENALTIES

- Please refer to the SET Policies document regarding Academic Integrity Information
- Please refer to the SET Policies document regarding Late Policy

EVALUATION

- Please see the Rubric for this assignment in eConestoga

PREPARATION

- Review the lesson content on Data Structures

REQUIREMENTS

Submission: Individual submission

You will use randomly created data to measure the performance of some data structures. Here are the specific requirements:

1. The application must be a Console application written in C#. Use the .NET Framework. DO NOT use .NET Core.
2. The application must use good object oriented programming techniques.

3. The only user input allowed is through command line arguments.
   The first argument is the number of elements in each of the data structures. It must be at least 100 and less than 5,000,000.
   The second argument must be the number of elements in the test arrays. It must be at least 1 and must be no more than 1% of the total number of elements in the data structures (value from first argument).
   If there is any error in the command line, the application must report the error, display the correct usage, and end the program.
4. The application should be broken up into the following general structure:
   a. Process command line arguments
   b. Create the data structures
   c. Load the data structures with random data
   d. Create string arrays with valid test data
   e. Create string arrays with invalid test data
   f. Perform the tests
   g. Display the results of the tests
5. The three data structures that will be measured for performance are:
   a. String array
   b. List of type String
   c. Dictionary
6. The random data must be GUID types. Each created GUID must be added to all three data structures. For the Dictionary, you may add the GUID as both the key and the value.
7. The array of valid strings must be randomly chosen from one of the data structures.
8. The array of invalid strings can be created in any way you choose. They should be GUID types as well.
9. Measure the time it takes to find all the valid test data in the data structures. The time measured should be the total time it takes to iterate through all the elements in the test array.
   a. For the string array, you can use the Array.Find() method.
   b. For the List, you can use the BinarySearch() method.
   c. For the Dictionary, you use the Key to find the data
10. Perform the same test as the previous one, using the invalid test data.
11. Display the results of the tests. Please make sure you show the command line arguments as part of the results. Also, show the average time it takes to perform the search for each data structure.
12. SET programming standards must be adhered to. This includes proper commenting, including header comments and comments for classes and methods.

13. You must submit regular commits to the GitHub repository for this assignment.

Things to consider:

- Eliminate sources that may affect your results in a negative way. For example, screen output is very time consuming. In addition, although your programming may be more elegant, the more layers of code a test traverses, the more inaccurate the results
- You should use the Stopwatch class in the System.Diagnostics namespace for accurate measurement of code execution.

FILE NAMING REQUIREMENTS

- There are no specific file naming requirements. Please keep the name short, but meaningful.
- The submitted file must be a .zip file (or fully compatible).

SUBMISSION REQUIREMENTS

- The full C# solution should be submitted in one zip folder.