**CIS 263 Assignment 2 - Experimental Performance Analysis**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Due Date**

* at the start of class on Monday, March 30.

**Before Starting the Project**

* Read the entire project description before starting. This is an individual assignment.

**Learning Objectives**

After completing this project you should be able to implement and evaluate the performance of C++V11 data structures.

**Rubric**

25 pts commented and readable code \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

25 pts elegant source code \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

25 pts concise design \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

25 pts results during testing \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Background**

As a class, we will determine the performance of several data structures on store and find tasks. You will utilize several of the common data structures and report results to the class.

**Step 0: Pick a Data Structure**

* In class, choose five from the following options: vector (unsorted), linked-list (unsorted), stack (unsorted), queue (unsorted), BST, AVL tree, Red-Black tree, Map, Set, Binary Heap, Quadratic Probing Hash Table, Cuckoo Hash Table, Leftist Heap, and Binomial Queue.

**Step 1: Gather data (5 data points)**

* Read in the file “RomeoAndJuliet.txt”. Ignore any empty lines and trim any leading whitespace. Store the entire line as a string in your data structure. Record how many data points each of your structures stored, e.g., 4000 lines (AVL tree), … You should have 5 data points, one for each structure.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 2: Insertion function (15 data points)**

* Add timing code to your program to determine how long it takes to read the file and build the data structure. Test your code on a computer not running any other program (preferably EOS) and preferably on a computer not being shared by others. You have 3 files to store into 5 data structures (15 cases). Provide the timings to store:
  + RomeoAndJuliet.txt \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  + RomeoAndJulietx10.txt \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  + RomeoAndJulietx100.txt \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 3: Find function (15 data points)**

* Write a function with timing code to determine the how long it takes to find a data point for each data structure. You should test the 5 structures with every query (only for RomeoAndJulietx100.txt). Find the following lines.
  + THE TRAGEDY OF ROMEO AND JULIET \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  + Rom. Tush, thou art deceiv'd. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  + THE END \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 4: Bundle your program**

* Bundle your program and turn in all files required to run your program.
* Turn in output that answers all of the questions without changing and recompiling your program. E.g.,

…

RomeoAndJulietx.txt - building AVL tree took: 1 ms

RomeoAndJulietx10.txt - building AVL tree took: 12 ms

RomeoAndJulietx100.txt - building AVL tree took: 1203 ms

…

* Ensure you include suitable documentation for your 1) overall project (e.g., an appropriate comment block at the top of your program, state which structures you used, and which performed best for these tasks) and 2) source code.

**Grading Criteria**

* There is a 50% penalty on programming projects if your solution does not execute or if it generates errors.
* There is a 50% penalty for not turning in a hardcopy (both pages of this document and printed code) and softcopy of code (zipped if needed) to blackboard.
* Any options/approaches/requirements not specified in this document are left for your own decision making, in keeping with the spirit of the assignment.

**Late Policy**

Projects are due at the START of the class period and not accepted later. Not turning in the hard copy or soft copy by the due date is considered a late/missing project unless PRIOR arrangements are made.