UNIVERSITY OF CALIFORNIA, MERCED

BERKELEY · DAVIS · IRVINE · LOS ANGELES · MERCED · RIVERSIDE · SAN DIEGO · SAN FRANCISCO   SANTA BARBARA · SANTA CRUZ

CSE 30: Data Structures
Laboratory 5 (Midterm Project)

Strict Due Date: Friday, October 16, 2020 at 9:00 pm

Introduction

This lab project is a little different from others because it will be a replacement for your midterm examination. As such, there will be no deadline extensions granted for any reason whatsoever. Not even medical or technical reasons, just as there are nom extensions given for midterm examinations.

The lab is designed to take 2 hours of work, which is similar to the time it takes to complete a midterm examination. Since we are giving you an entire week to complete a 2-hour assignment, there will be no other extensions given.

The lab is completely open-book, which is the same policy as in any other lab in the course. While you are allowed to use whatever resources you want, your solution *must* be your own work.

Support Code

In your `inc` folder, there is a file called `LinkedList.h`, and `Array.h`. All the functionality you need to perform your experiments has already been implemented. There is also a file called `TimeSupport.h` that allows you to measure the running time of code segments. See Lecture 9 for examples of this. There is also a file called `RandomSupport.h`, which you can use to generate good random numbers in your code. There is an example of how to do that in the scratchpad of this project.

Lab Report

Answer the following questions, type your answers and save the document as a PDF. Only PDFs will be accepted. Upload the PDF file into your CodeSync project and perform a git commit and a git push. (You can do this from outside CodeSync if you want, but your PDF file must end up in the git repo)

1. With respect to resizable arrays, establish the relationship between the size of the array and the time it takes to perform random access (read a the value stored at a given position somewhere in the array). Provide empirical evidence to support your answer.

2. With respect to linked, establish the relationship between the size of the list and the time it takes to perform random access (read a the value stored at a given position somewhere in the list). Provide empirical evidence to support your answer.

3. With respect to arrays, establish the relationship between the size of the array and the time it takes to perform an insertion at the end of the array. Provide empirical evidence to support your answer.

4. With respect to arrays, establish the relationship between the size of the array and the time it takes to perform an insertion at the beginning of the array. Provide empirical evidence to support your answer.

5. With respect to linked lists, establish the relationship between the size of the list and the time it takes to perform an insertion at the end of the list. Provide empirical evidence to support your answer.

6. With respect to linked lists, establish the relationship between the size of the list and the time it takes to perform an insertion at the beginning of the list. Provide empirical evidence to support your answer.

Programming Task

In your `app.cpp` file, create a demo illustrating a scenario where storing numbers in a linked list is more efficient than an array. Your demo should generate a sufficiently large number of random integers and insert them into both the list, and the array. Your demo should also provide the time it took to complete the operations in the array, and in the linked list. It should show that the linked list was faster.