CSCE 221 Cover Page

Programming Assignment #1

First Name: Seth Last Name: Barberee UIN: 324009275

Any assignment turned in without a fully completed coverpage will receive ZERO POINTS.

Please list all below all sources (people, books, webpages, etc) consulted regarding this assignment:

CSCE 221 Students	Other People	Printed Material	Web Material (URL)	Other
1. Giusepee	1. Nathan Leake	1.	1.	1.
2. Edgar Martinez	2.	2.	2.	2.
3.	3.	3.	3.	3.
4.	4.	4.	4.	4.
5.	5.	5.	5.	5.

Recall that University Regulations, Section 42, define scholastic dishonesty to include acquiring answers from any unauthorized source, working with another person when not specifically permitted, observing the work of other students during any exam, providing answers when not specifically authorized to do so, informing any person of the contents of an exam prior to the exam, and failing to credit sources used. Disciplinary actions range from grade penalties to expulsion. Please consult the Aggie Honor System Office for additional information regarding academic misconduct – it is your responsibility to understand what constitutes academic misconduct and to ensure that you do not commit it.

I certify that I have listed above all the sources that I consulted regarding this assignment, and that I have not received nor given any assistance that is contrary to the letter or the spirit of the collaboration guidelines for this assignment.

Today's Date: 2/14/18

Printed Name (in lieu of a signature): Seth Barberee

Introduction

The purpose of this assignment is to implement and compare the different implementations of the Stack Abstract Data Type. Three of which are the ArrayStack, Doubling ArrayStack, and LinkedList Stack. To compare the implementations, we will push an abundance of random numbers onto the stacks and time how long it takes to push that many onto the stack.

Theoretical Analysis

In this section, you should provide an analysis of the complexity of a push operation.
Describe the effect of a push operation and the advantages and disadvantages of the three strategies.
ArrayStack: An element is added to the top of the stack. If full when trying to add it, the capacity is increased by a constant factor of 10. The data is then copied to a new array of that length. After, the new element is added to the top of the stack.
DoubleArrayStack: An element is added to the top of the stack. If full when trying to add it, the capacity is doubled and moved to a new array of that length. Then, the new element is added to the top of the stack.
Linked List: An node is appended the front of a singly-linked list. The list never is full and continues to grow as data is added.
What is the complexity of a push (on average) for the different implementations?
ArrayStack: O(1 **)**
DoubleArrayStack: O(1 **)**
Linked List: O(1 **)**
What is the worst case complexity for the different implementations?
ArrayStack: O(n^2 **)**
DoubleArrayStack: O(n **)**
Linked List: O(n **)**

Experimental Setup

a. Machine specification
Intel Core i7 6820HQ 4 cores/8 threads
16 GB DDR4 RAM
Arch Linux x86_64 4.15.1-2
b. How did you generate the test inputs? What input sizes did you test? Why? What parameters did you use for your initial Stack size and increment amount in the case of the ArrayStack?

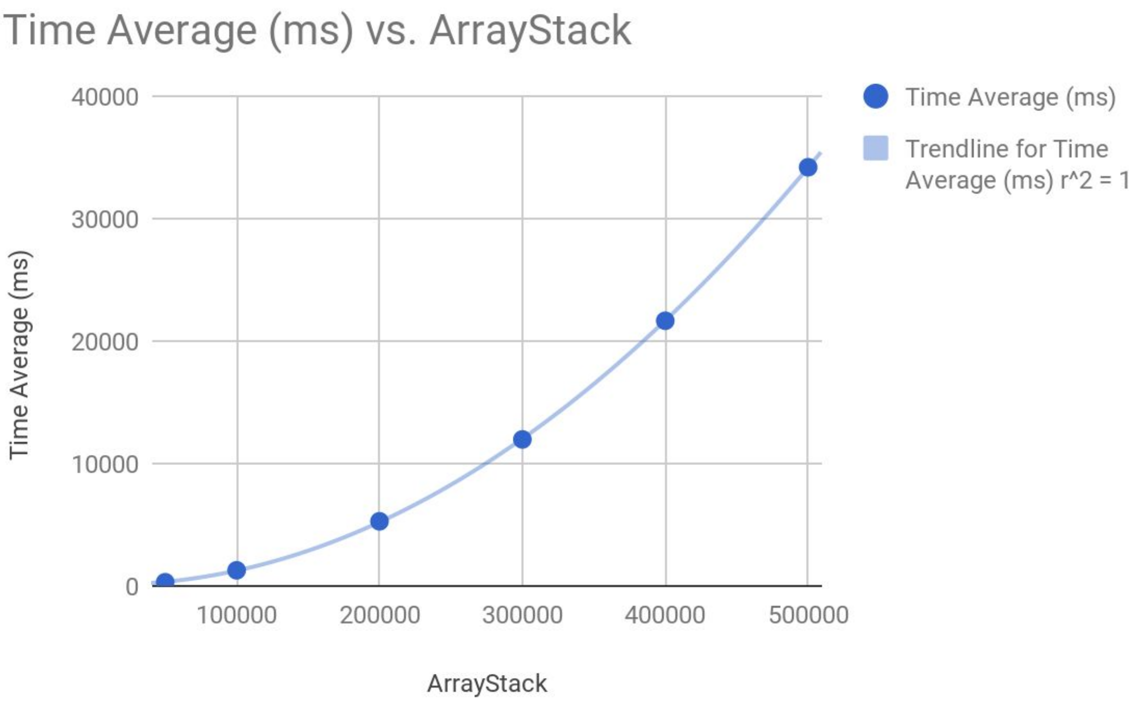
Inputs were generated with a for loop that started from 0 up to the targeted input size. The initial stack size was 5 and incremented by 10 in the ArrayStack.

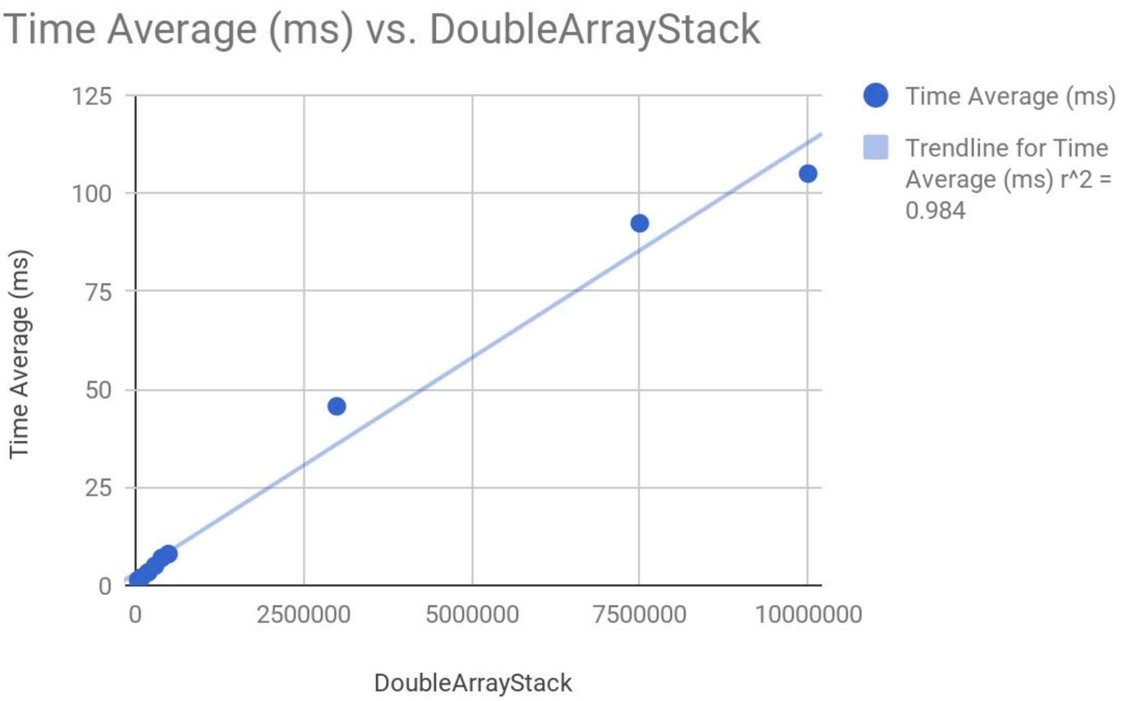
c. How many times did you repeat each experiment?
3 times and an average was taken for the data point for the graphs shown below.

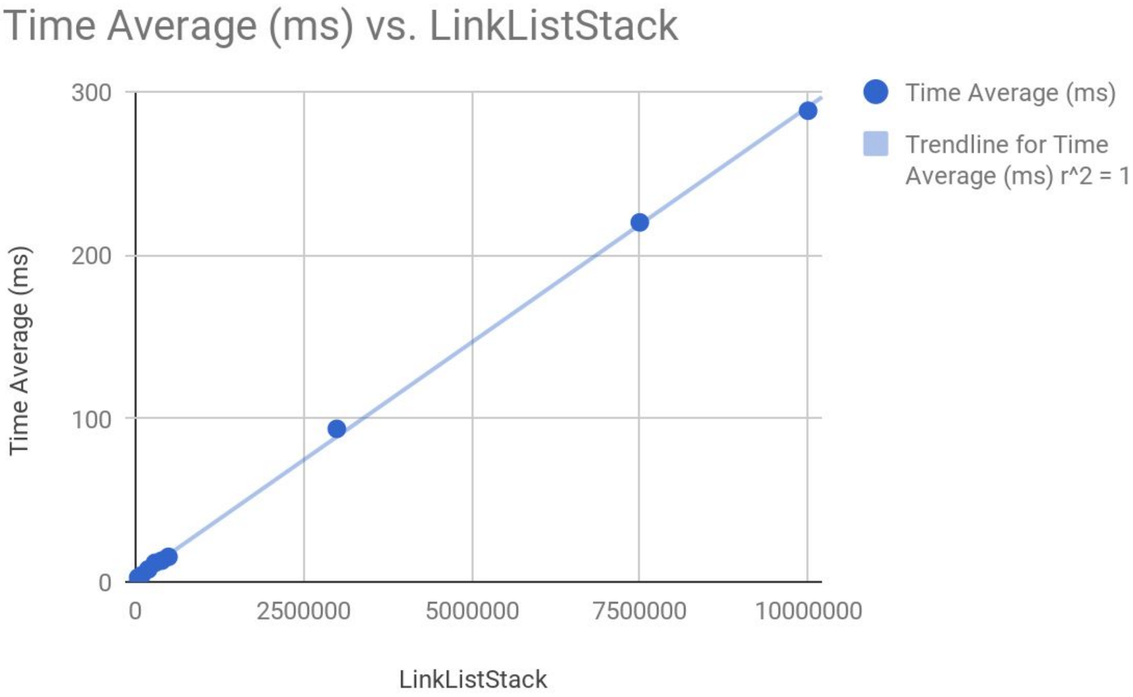
Experimental Results

In this section, you should compare the performance (running time) of the push() operation in the three different implementations to one another and to their theoretical complexity.

a. Make a plot showing the running time (y-axis) vs. the number of push operations (x-axis). You must use some electronic tool (matlab, gnuplot, excel, ...) to create the plot –hand-written plots will NOT be accepted.







b. Provide a discussion of your results, which includes but is not limited to:
   i. Which of the three Stack implementations performs the best? Does it depend on the input?

When the input is less than 50000 inputs, the DoubleArrayStack and LinkListStack become virtually tied. When greater than or equal to 50000, the DoubleArrayStack starts to separate itself as the fastest implementation.

ii. To what extent does the theoretical analysis agree with the experimental results? Attempt to understand and explain any discrepancies you note

All of the theoretical analysis agree with the experimental analysis. With the R^2 of each graph close to 1, it is shown to agree with the data.