Andy Dao | uID: u0692334 | CS 2420

ANALYSIS DOCUMENT ASSIGNMENT 7

1. **Have you worked with more than one partner yet? Remember, you are required to switch at least once this semester.**

   I have not as of this assignment, but for our next assignment that we have partners for, I already have one lined up.

2. **In the LinkedListStack class, the stack data structure is implemented using a doubly-linked list.**
   Would it be better to use a singly-linked list instead? Defend your answer.

   I believe it would be better to just use a singly-linked list instead because we aren't needing to traverse both ways in a stack. With a stack, we only need to know what's on top of the stack, known as peeking, remove/pop, or add/push to that place in the list – so we are only really using the head of the linked list.

3. **Would it be possible to replace the instance of DoublyLinkedList in the LinkedListStack class with an instance of Java's LinkedList?**
   Why or why not?
   Yes, definitely. Essentially our DoublyLinkedList is implemented by our custom LinkedList class from a previous assignment, but it implements mostly the same methods as Java's LinkedList. Our custom implementation is just missing some methods is all, so it is very possible to replace the DoublyLinkedList class with Java's LinkedList class.

4. **Comment on the efficiency of your time spent developing the LinkedListStack class.**
   Since I knew it was going to be fall break, I got to start on the assignment fairly early to get it done. Procrastination definitely gets to me after not persistently doing work throughout my week. Since I got started even before we were let off

of fall break, I got done before fall break was over, which was really stress relieving for me.

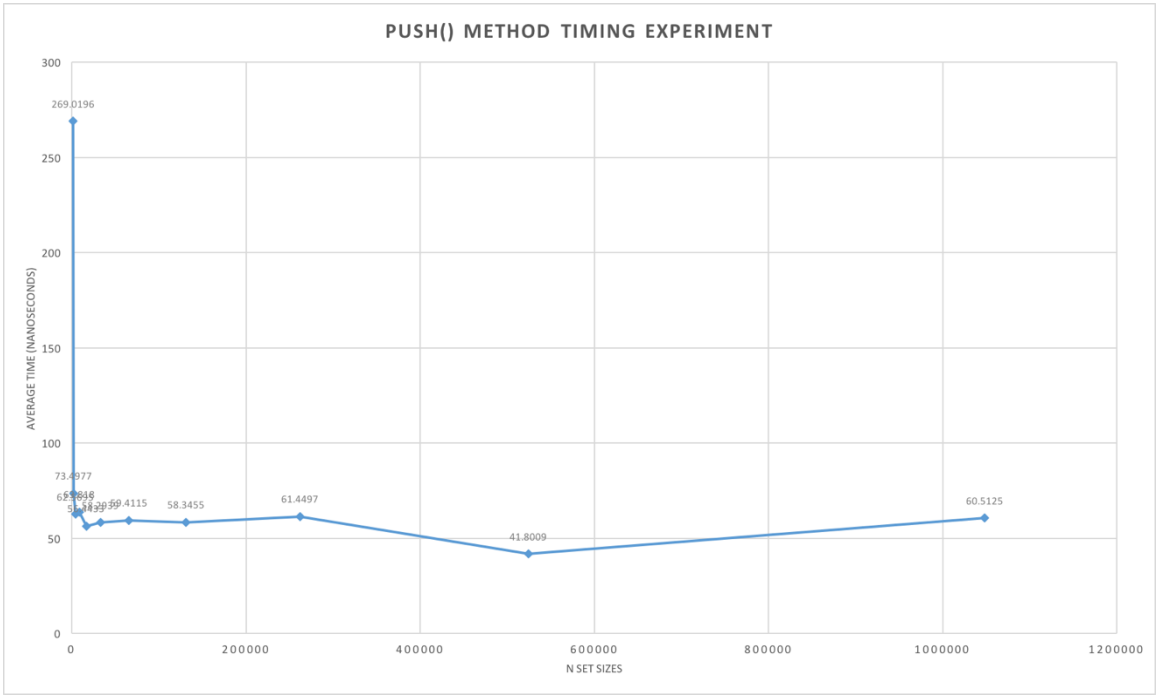
5. **Note that the line and column number given by BalancedSymbolChecker indicate the location in a file where an unmatched symbol is detected (i.e., where the closing symbol is expected).**

   Explain how you would also keep track of the line and column number of the unmatched opening symbol.

   For example, in Class1.java, the unmatched symbol is detected at line 6 and column 1, but the original '(' is located at line 2 and column 24.
   We could have two other variables keep track of the first original symbol(s) location in the file, then return them when the unmatched symbol is returned if that's what we want to do with the original symbol(s) location. I'm not sure if it'll work, but just thinking through, we could use HashKey's to keep track of the symbol and the line/column it's found in, as a whole object itself. I feel like this would be very efficient in memory, and the readability is more simple as well for debugging and testing.

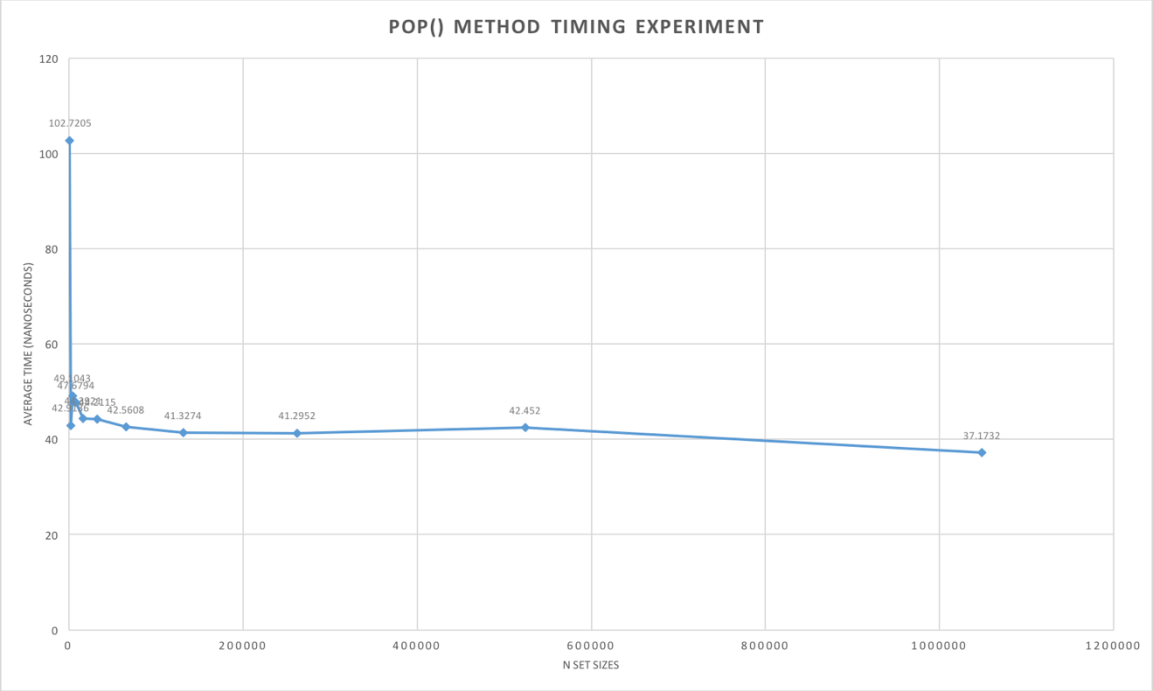
6. **Collect and plot running times in order to determine if the running times of the LinkedListStack methods push, pop, and peek are O(1) as expected.**
   Here for my push method is the data graph:

   Push Method:



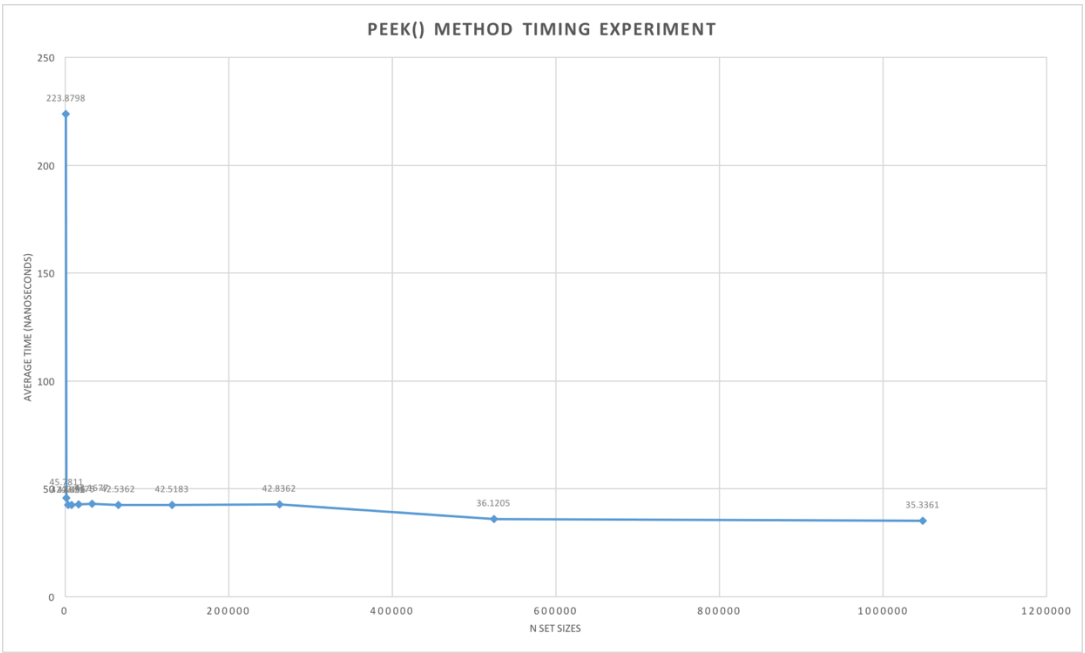
It definitely did return an O(1) complexity time as expected. It is super quick and efficient. *The data in the very beginning can be ignored as there's multiple things happening and the data warms up, giving inaccurate data in the very beginning.*

Pop Method:



Again, it ran at O(1) again. Very efficient, as needed for a stack implementation.

Peek Method:



Above shows the peek method runs at a O(1) complexity as well.

Conclusion:
A stack inserts and removes by something called the ***Last-In-First-Out*** principle**,** which is self-explanatory in its' name. Even though it's a limited data structure since we are only able to add and remove from the stack at the top, it has it's applications in the real world, and doing it very efficiently. Some of the things that come to mind is that if we want to reverse a world, we can push all the letters to the stack, and pop (returning the removed letter at the top of the stack) each one of those letters. Upon researching as well, one of our worlds greatest feature implemented by a stack, is the **undo** button in most applications.

7. **How many hours did you spend on this assignment?**
   About 12 hours of work was done during this assignment.