

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

Not yet, but I have a new one for assignment 8.

*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.*

The stack structure depends on the topmost item, which can be assigned as the head or tail of a singly-linked or doubly-linked list. The references are then assigned or reassigned based on the operations, but there is no traversal up and down and no change in efficiency in dealing with this end node whether you're using a singly- or doubly-linked list, so it wouldn't make much of a difference to use a singly-linked list over a doubly-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 it would be possible. `LinkedList` has push, peek, and pop methods, which we need. It has the functionality of keeping track of a "top" node. It is virtually the same as Doubly Linked List in terms of usability for a stack structure. The `isEmpty` method is missing, but it is easily implemented by checking for the size of the `LinkedList` being zero or not.

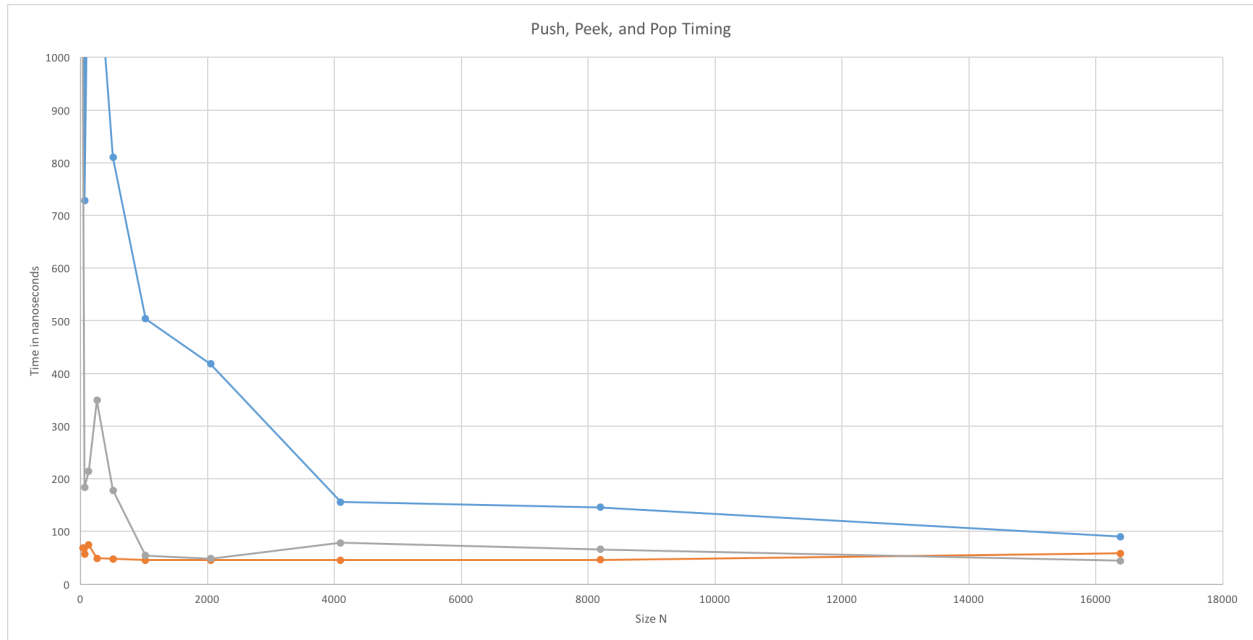
*4. Comment on the efficiency of your time spent developing the `LinkedListStack` class.*

I feel I was pretty efficient. It seemed like a simple and straightforward assignment that I got done in a few small spurts of work.

*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.*

When symbols are added to the stack, it could be implemented so that the node contains not only the data information but also the location in two variables called `startRow` and `startColumn` for example. That way when an unmatched symbol is encountered, it can return the location of the first symbol by calling on the data and the location of the node that doesn't have a closing symbol.

*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 is the timing for all methods, push being blue, peek being grey, pop being orange, on a sample set of  $2^5$  to  $2^{14}$ . Aside from the initial noise that was quite big for all of them and was cut out of the graph, the methods do operate on  $O(1)$  time, operating extremely close to the same by the last few iterations, within nanoseconds of each other.

7. *How many hours did you spend on this assignment?*

I spent about 3 hours on this assignment.