Zachary Cutler, u1025642

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

Not yet, no

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.

A single linked list would be slightly better, since theres no situation where you'd have to navigate through the stack backwards, and thus no need for that implementation. In reality the difference in team would be very minor, but in theory it would be better.

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?

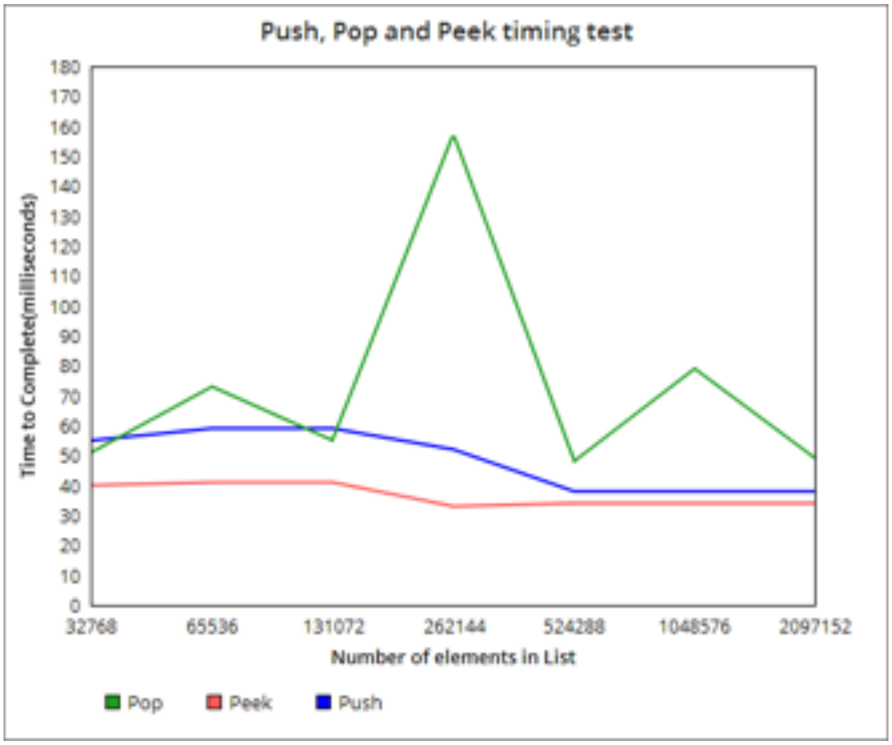
Yes, you could easily replace it with java’s LinkedList because it has essentially the same functionality of our current DoublyLinkedList, the only real difference you'd fine is java's class would likely be slightly better optimized.

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

It was exceptionally efficient, since almost the entirety of its methods had already been essentially created in DoublyLinkedList, and the ones that weren't used already implemented methods, so it was quite easy.

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.

You could modify DoublyLinkedList slightly so that the items popped onto the list would have 2 additional fields for their location within the node, and those could be stored and seen in the node whenever you want after putting them into the list. So when you go to pop an item off the list you could return its location which is stored in the node. The solution would be specific to this problem and would not aid DoublyLinkedList in other situations, but would do the job.



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.

As the graph above shows, all of the methods are indeed O(1) as expected across a wide variety of N sizes.

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

Around 15 hours