Dylan Northcutt
U1055102

1. Have you worked with more than one partner yet? Remember, you are required to switch at least once this semester.
I have not worked with more than one partner in the paired programing. I will find a new partner for the upcoming assignments.

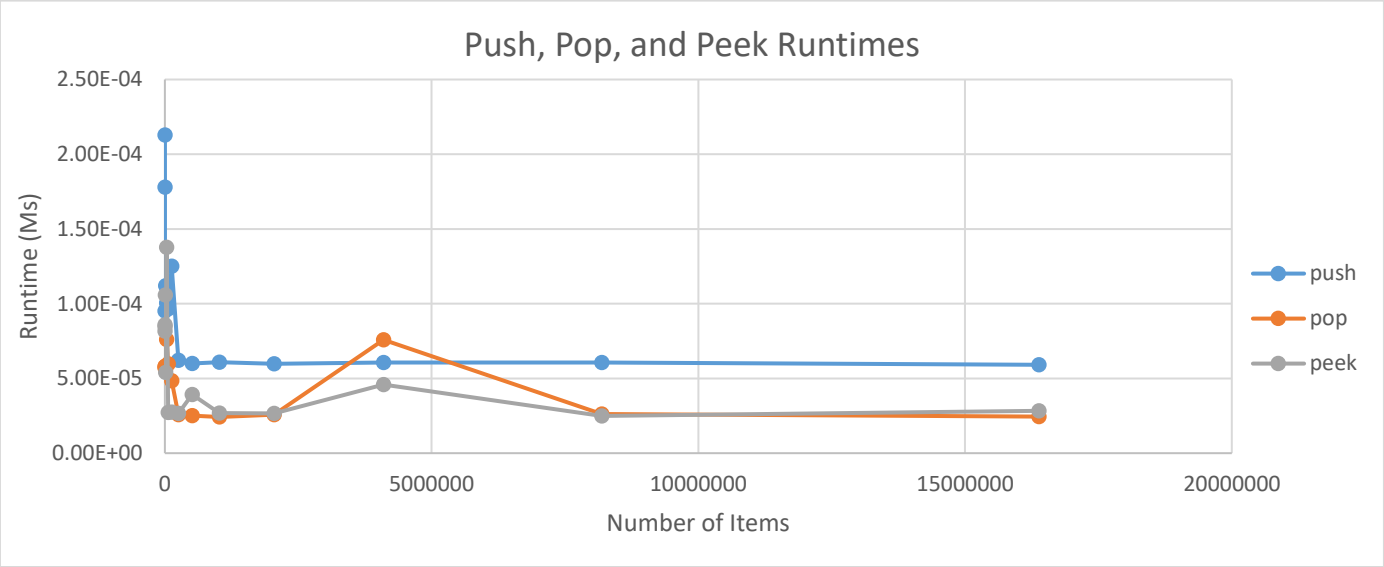
2. In the LinkedListStack class, the stack data structure is implemented using a doubly-linked list. Would it be better to use a singly-linked list instead? Defend your answer.
For a stack both doubly-linked lists and singly-linked list would work. Both work about the same. For a stack that is only accessing the first element in the stack without need for a tail a singly-linked list is more appropriate. The only difference is that there is no need to assign a previous node for a stack making the double part of doubly-linked list pointless. In terms of efficiently assigning a previous and tail causes so little runtime that it is essentially the same as a singly-linked list. Both doubly-linked list and singly-linked work about the same for this assignment however a singly-linked list seems slightly more practical.

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?
It is possible to replace the DoublyLinkedList that we used with java's LinkedList. Both of these options have many of the same methods used for a stack. They both have an addFirst method to uses for push and a removeFirst to use for pop. It should also be noted that Java's LinkedList may be more effective as it has a push and pop method of its own.

4. Comment on the efficiency of your time spent developing the LinkedListStack class.
My time was efficiently spent in this assignment. Very little time was spent initially programing the assignment. Most time was spent debugging and adding lines for unexpected inputs such as escape characters and characters within a string.

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.
The position of the opening brace could be kept track of by adding the character as a new object with variables for symbol itself, the line number, and the position. Instead of storing a node with one piece of data it is stored with three so when it is popped the line and place are also available.

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.



The runtimes of push, pop, and peek appear to be O(1) as expected. All the runtimes are constant aside from variations where the gaps weren't as larger and fewer items were tested. Had the test continues with more, larger items the chart would likely show a continuation of the constant runtime.

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

I spent about 6 hours on this assignment.