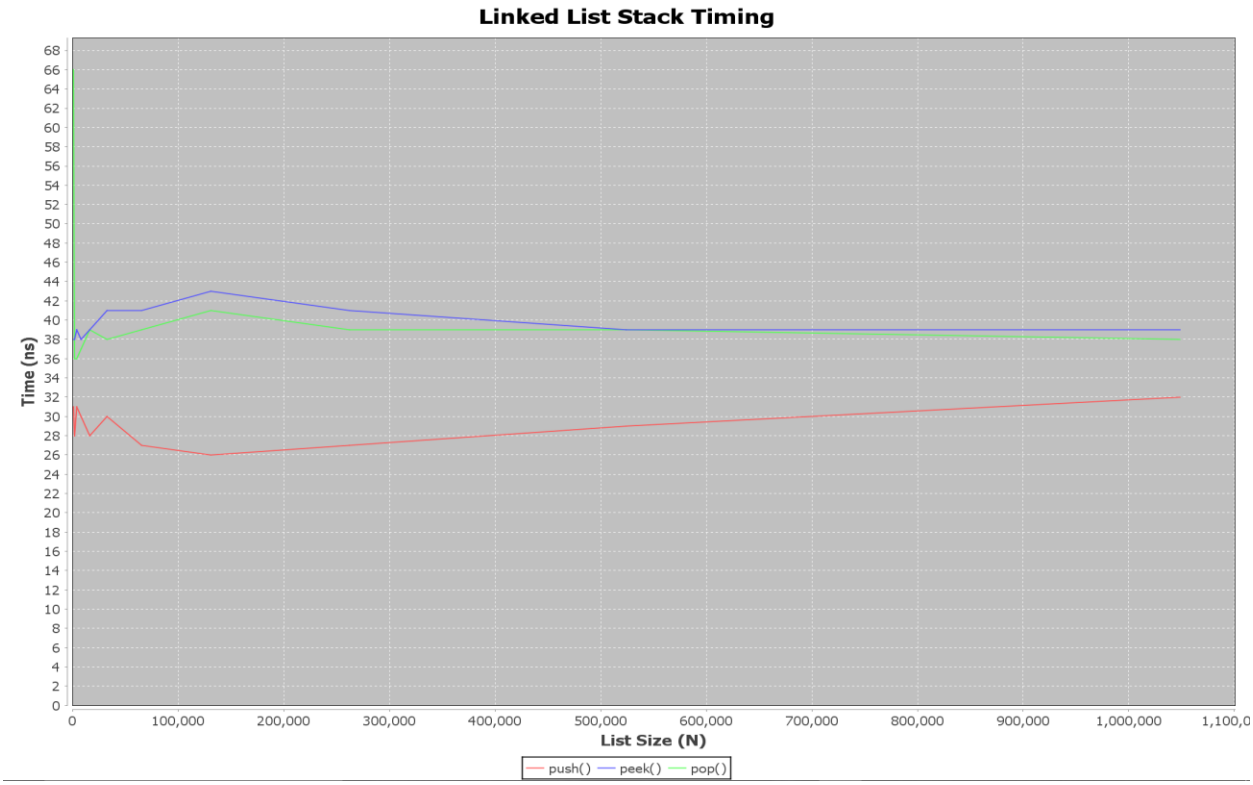
Braeden Barwick

1. I have worked with more than one partner this semester.
2. Using a singly linked list wouldn't affect the performance too much, but it would affect the implementation efficiency, as each node only needs to store its .next. A singly linked list would still have the functionality required for a stack.
3. It would be possible to implement Java's LinkedList class to back our LinkedListStack. It has the same method functionality that is required. Furthermore, LinkedList has its own implementation for push(), peek(), and pop() when used to back a stack.
4. I spent less than a minute developing the LinkedListStack class. Nearly all the implementation just required a method call to the backing list, so most methods had a single line of code.
5. You could have a second column and row counter, and every time you read an open symbol, copy those row/column coordinates into the new counters. Then when you run into an unmatched symbol error, you can also return those counters to get the coordinates of the last open symbol.

6. 

   Even though the list increases in size exponentially, the timing of push(), peek(), and pop() all remain constant, as expected.

7. I spent about 5 hours on this assignment.