Haoran Chen

U1060286

Assignment 7 Analysis

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

   Yes. I have already worked with two partners.

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.**

   Yes. We can add to first when push method is called. Then for peek and pop, we detect and delete the first node in the singly linked list. In this case, I can treat the first node in the SLL as the top of stack and the last node as the bottom.

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 as Java’s LinkedList is also a doublylinkedlist. The methods I used here could also be found in Java’s LinkedList class.

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

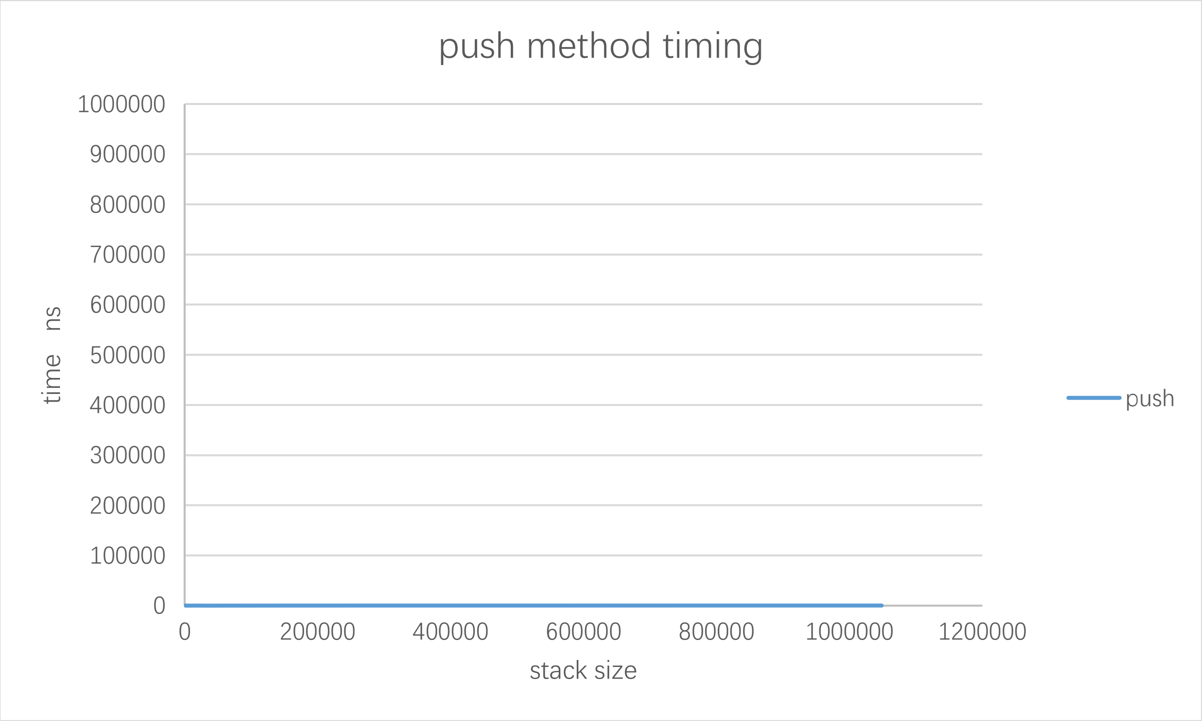
   I only used about five minutes for this class as I can just call the methods provided in doublylinkedlist class.

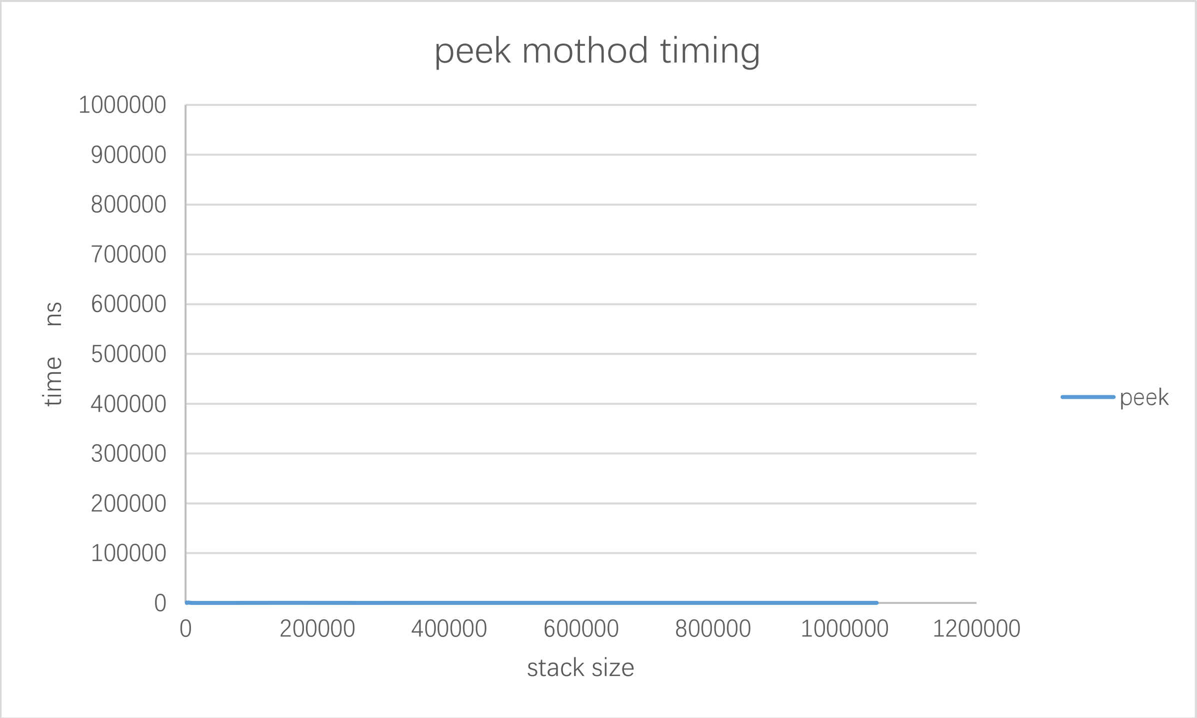
5. **Note that the line and column number given by BalancedSymbolChecker indicate the location in a file where an unmatched symbol is detected (i.e.,**

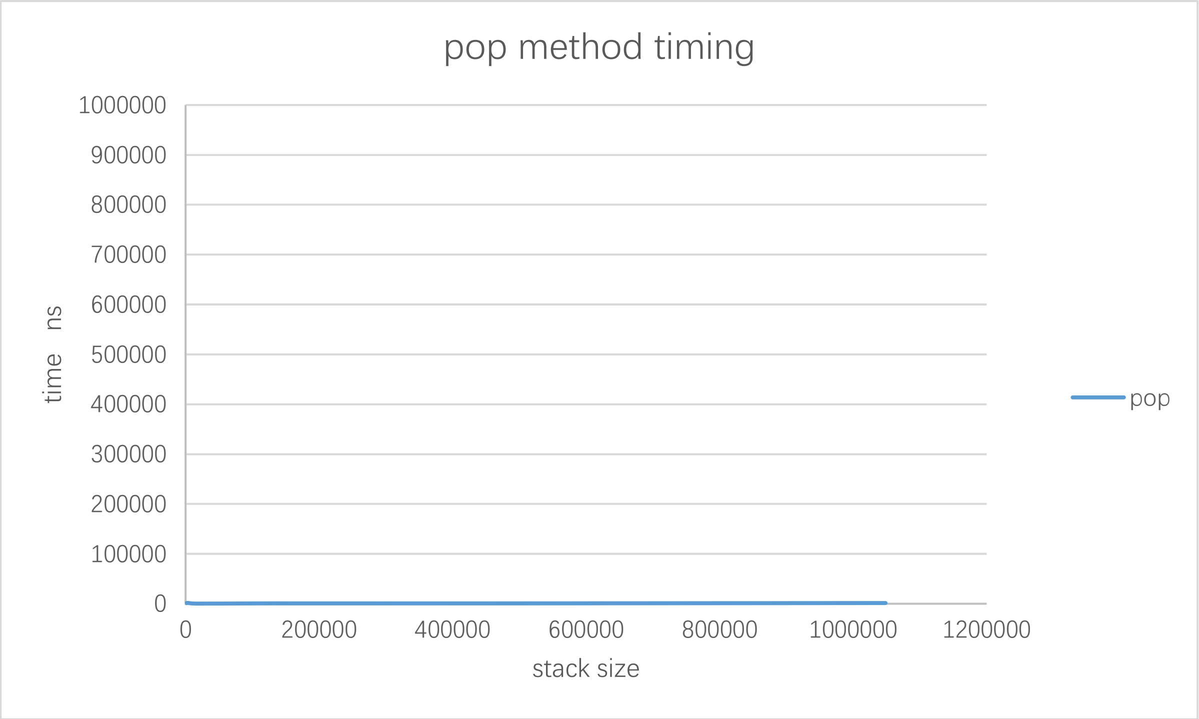
where the closing symbol is expected). Explain how you would also keep track of the line and column number of the unmatched opening symbol. For example, in Class1.java, the unmatched symbol is detected at line 6 and column 1, but the original '(' is located at line 2 and column 24.

As the doublylinkedlist used here is written in generic, I can generate an object containing the symbol, the line number and column number in it, and I can store that object in the linked list. So that whenever I need the location of the symbol, I can read it from the object.

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 shown above, the time taken for doing all the methods once are not influenced by the size of the stack, which is kind of constant compared to large time axis. The reason why I set a large time range is that the time I got shifted by noise of computer from around 100ns to 1300ns, the graph could not show any conclusions. The lines are horizontal with gradient of 0. Thus the Big(O) for all the three methods are O(1).

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

   About 3.5 hours.