William Frank
10/19/16
Assignment 07 Analysis

1. I have not yet worked with more than 1 partner, but I will be working with a new partner on the next project.

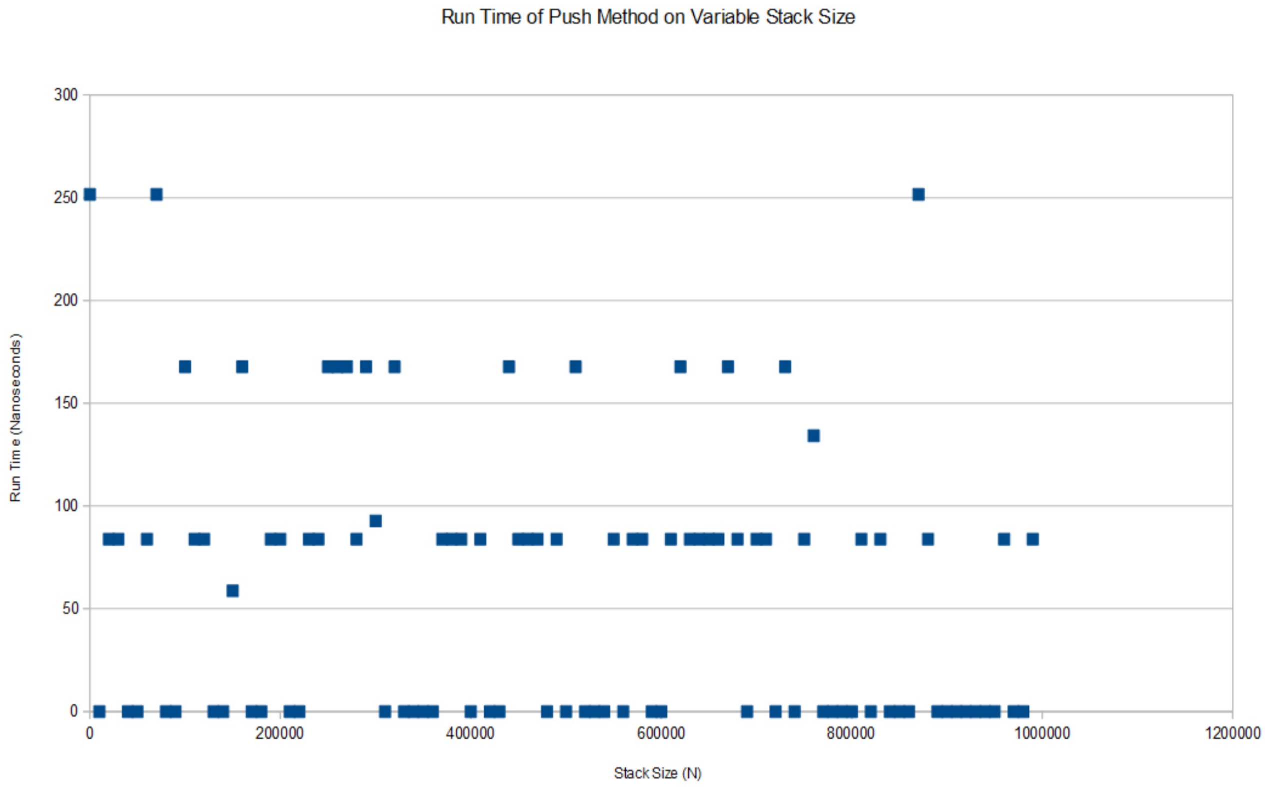
2. Using a singly-linked list could potentially be better. When using a stack you never have to access the “bottom” of it, so using a doubly-linked list is rather pointless. By using a singly-linked list adding elements would be ever so slightly faster because you wouldn't have to assign previous nodes. For this reason using a singly-linked list would be better, but by an almost immeasurable amount. It would also require less memory as there are only pointers in one direction.

3. Because Java's LinkedList class is a doubly-linked list, it should be able to easily replace the instance of DoublyLinkedList that is used in our stack.

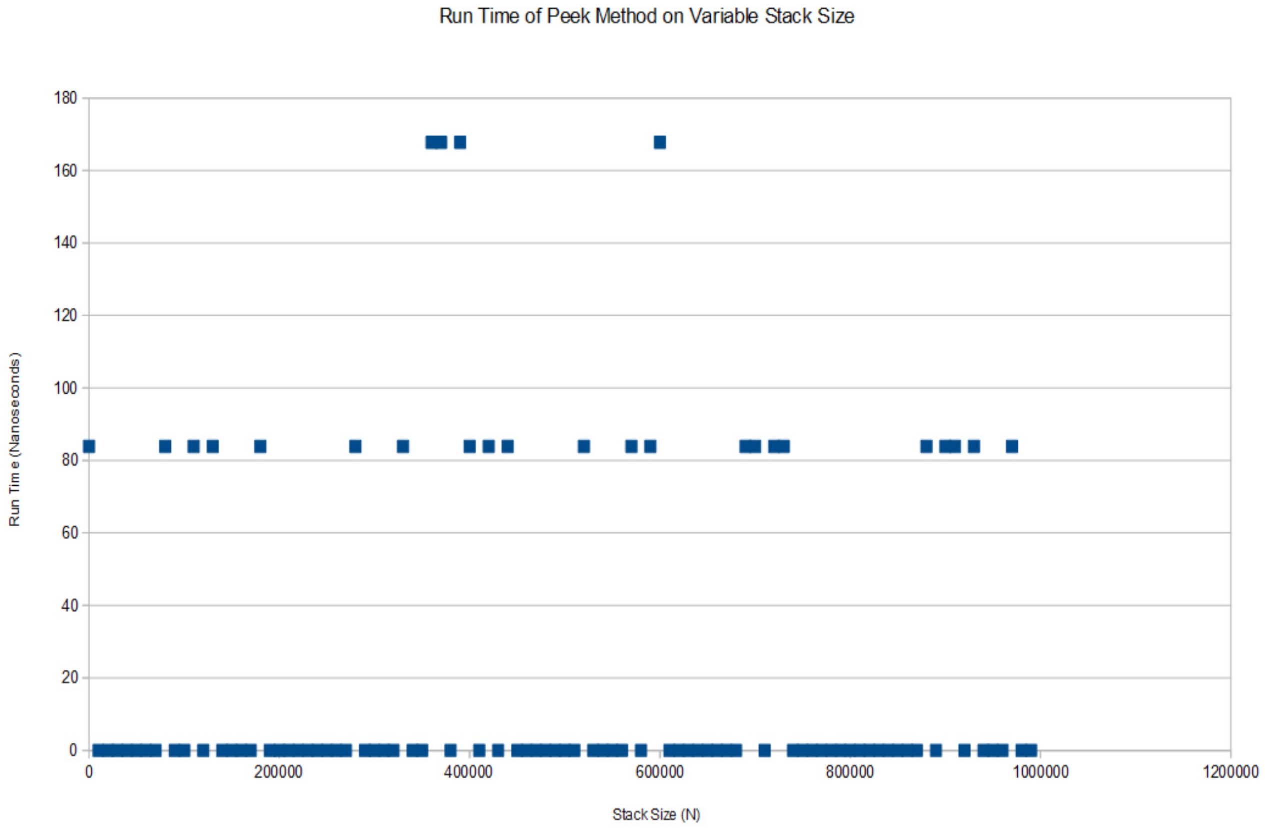
4. I was able to implement the LinkedListStack very quickly. All the stack methods had an analogous doubly-linked list method, so they were all one line of code. It was very easy to implement a stack backed by a doubly-linked list.

5. In order to keep track of the location of the original unmatched symbol you could store it's location along with what the character is in the node. This way when you look at the top of the stack you will know what character needs to be matched next, and where in the file that character is originally found.

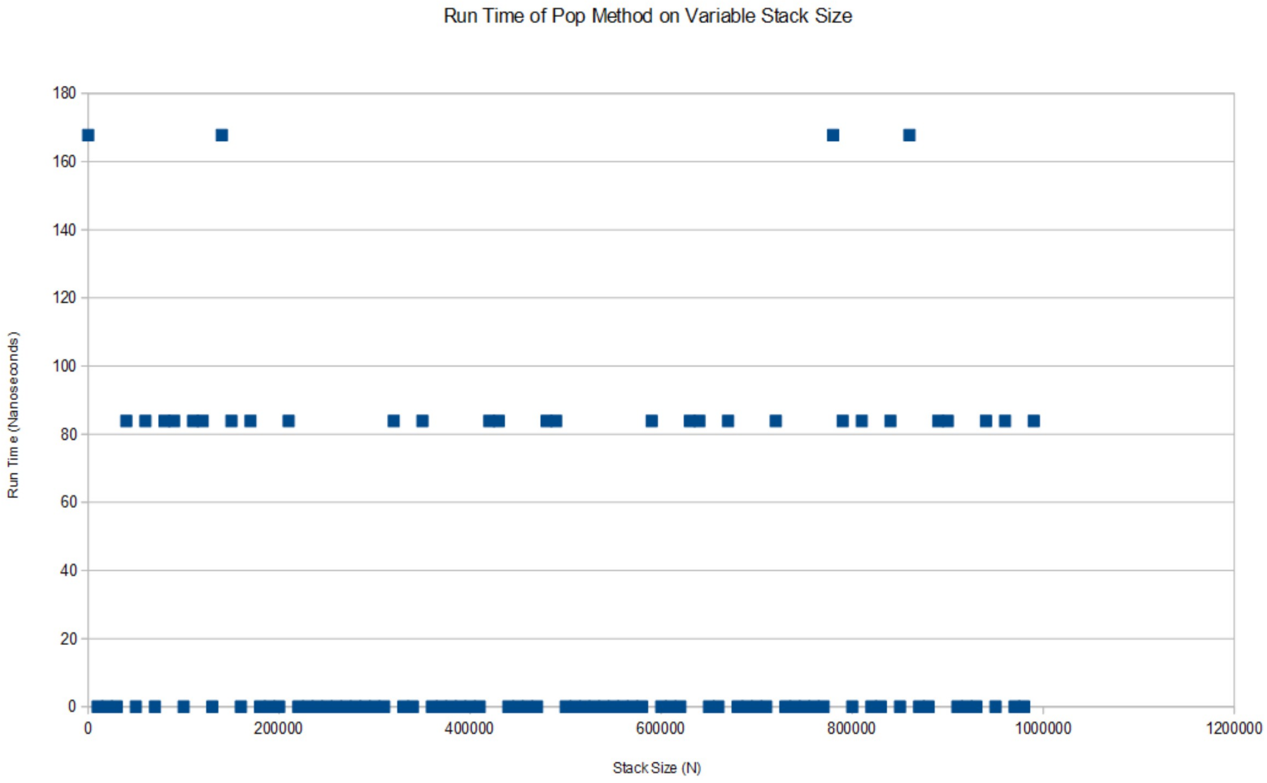
6.



Run Time of Push Method on Variable Stack Size
Run Time (Nanoseconds)
300
250
200
150
100
50
0
0
200000
400000
600000
800000
1000000
1200000
Stack Size (N)



Run Time of Peek Method on Variable Stack Size
Run Time (Nanoseconds)
180
160
140
120
100
80
60
40
20
0
0
200000
400000
600000
800000
1000000
1200000
Stack Size (N)



As you can see by these graphs, the run time of the push, peek, and pop methods is constant.

7. I spent 5-10 hours on this assignment.