Jordan Gardner u0566259

Assignment 07 Analysis Document

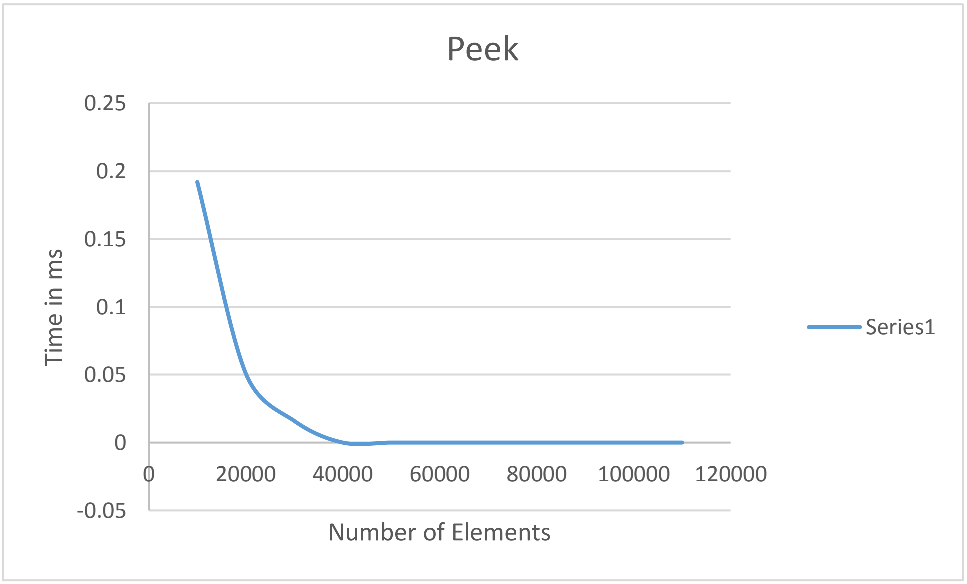
No I have not worked with another partner yet but I plan to do so for assignment 8 or 9.

I believe it would be more efficient to use a single linked list just due to the fact on the stack you are only concerned with the top of the stack. You can only access one point of the list therefore a singly linked list would suffice.

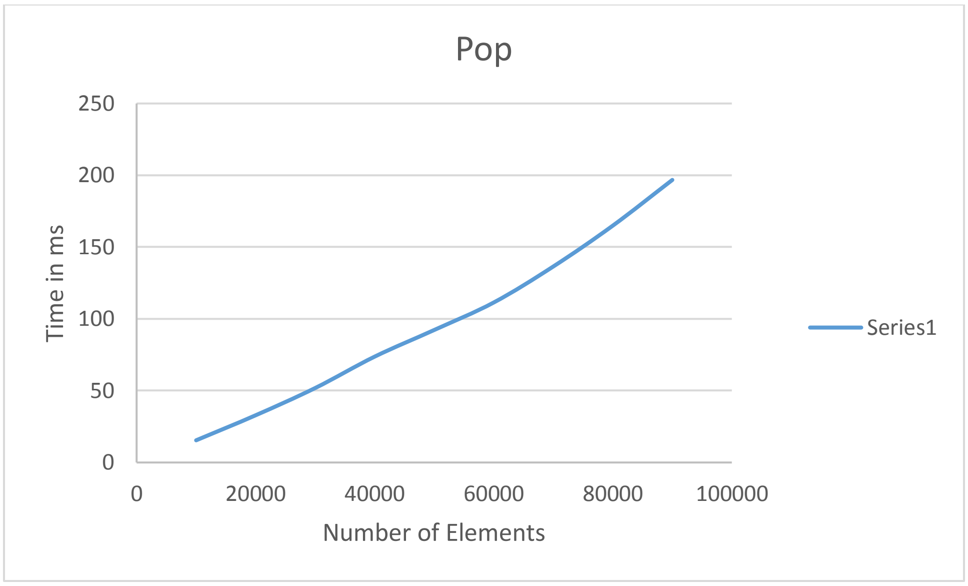
I don't think this will work because LinkedListStack requires implementation of an interface and LinkedList is not an interface. I tried to get this to work and it said that there was not a proper superinterface implemented. LinkedList implements List which is an interface.

The LinkedListStack class was very easy because we had imported the DoublyLinkedList class. This part of the assignment took about 30 minutes to complete. However, the BalancedSymbolChecker class is another story. This class took me the majority of my time to figure out and debug. To answer the question my time was well spend on the LinkedListStack class and use of this class in other parts of the program ran smoothly.

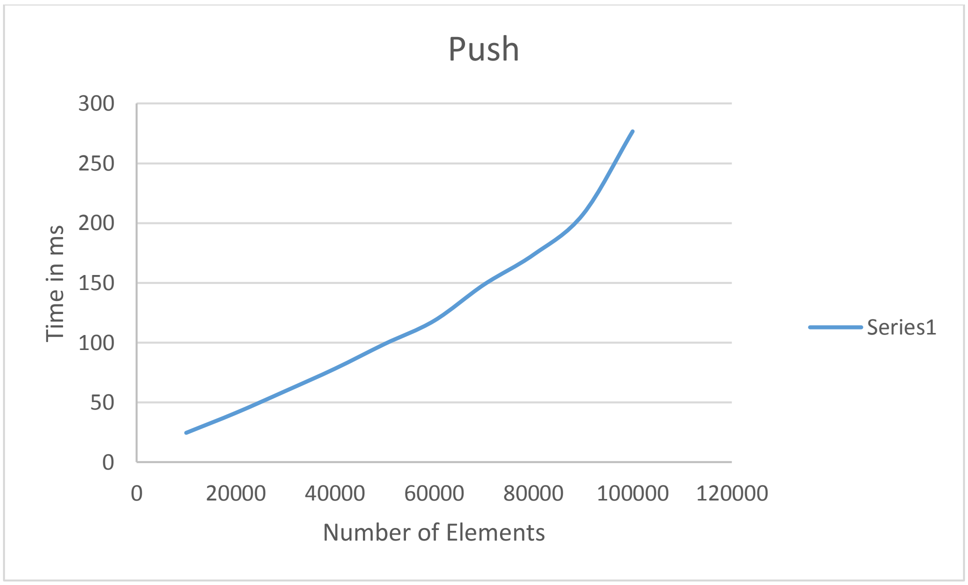
How you would do this is keep track of each symbol until it was closed. Therefore you would have to keep pushing things onto the stack as well as saving the location of that symbol. If the symbol has a match then it would pop the stack and try to find the next closing symbol. If there isn't a closing symbol there would be an opening symbol on the stack with the location of that symbol. You may have to use a separate linkedlist stack to keep track of the locations as well as the symbols.



Peek looks to be O(1) after the first couple iterations.



Pop seems to be close to O(n) as the Number of Elements increases the amount of time it takes to pop increase. I would expect to see a more drastic movement in time if I did something wrong. This is still very little time to pop off the stack at 100,000 elements.



Push has similar data to Pop in that time seems to increase as we increase number of elements. This is intriguing to me and might warrant some extra consideration. With that being said I think that time would have to increase much more for me to worry about a code problem. This may demonstrate a worst case scenario, but it seems to follow an O(n) behavior.

Let me know thank you.

9-11 hours