Nathan Milot
CS 2420
10/12/16

Assignment 07 Analysis

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

No I have not.

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.

I believe it would be slightly more efficient to use a singly-linked list due to the fact that you only have to keep track of one link per node. This would slightly reduce the amount of storage per node which could be useful if you are dealing with massive stacks. Additionally, it could slightly improve runtime performance since the clear method would only have to reset a head and size and no a tail as well. These optimizations however are only slight and would not affect the asymptotic behavior for either runtime or storage. Especially in the case of storage, it could be beneficial when dealing with large N and very limited storage.

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?

Yes, it would be. Java’s LinkedList is also a DoublyLinkedList and has much of the same functionality with similar, if not slightly better performance. Additionally, Java’s LinkedList has methods like pop, push, and peek built into it. This would make the implementation of a LinkedListStack class very simple because for each of those methods you would simply call the LinkedList’s method.

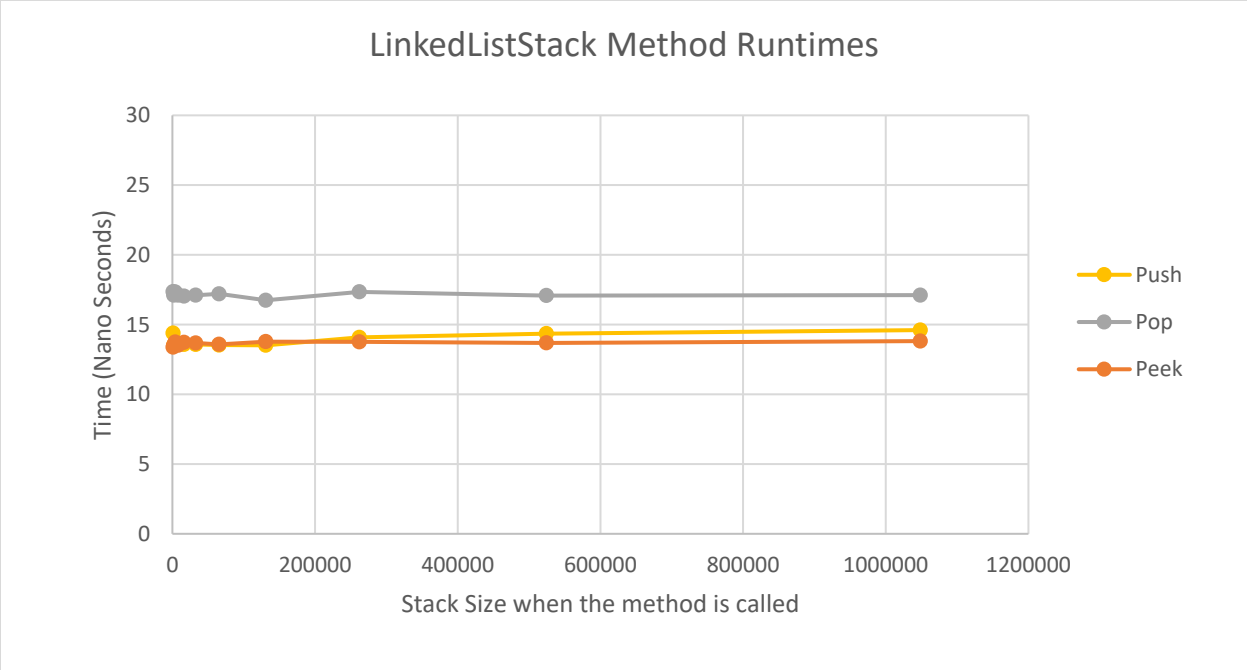
4. Comment on the efficiency of your time spent developing the LinkedListStack class.

I feel like I was able to efficiently spend my time on this project. Because it was a solo assignment, I could work more flexibly around my schedule and for shorter periods of time. So instead of two hour sessions with a partner I could spend half an hour before class or take a break whenever I felt I needed one. This was really helpful to keep me focused and highly on task when I was working, and to help me not get too stuck on any one problem.

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.

I would create a SymbolItem, or some other object, to store the character and location of each item pushed to the stack. If there was ever a discrepancy the line and column location would be stored in the object and could be used in the error message. An example of what I would do is included in my code as a private SymbolItem class.

6. Collect and plot running times in order to determine if the running times of the LinkedListStack methods push, pop, and peek are O(c) as expected.



As you can see, the runtime of each push, pop, and peek all have O(c) behavior. Pop has a slightly higher runtime of about 17 Nano seconds whereas peek and push have a runtime of about 13 and 14 Nano seconds respectively. Though there are slight differences in each methods runtime at different points this is due mainly to processing speed, or other noise. In spite of that noise, there is a very good trend line showing the constant behavior of each method.

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

About 6-8 hours.