Ella Moskun | u0897242
Prof Miriah Meyer
CS 2420-001 | Intro Alg & Data Struct
19 October 2016

Assignment 7: Symbol Matching

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

I have NOT yet worked with more than one partner yet. I WILL be working with a new partner on the next project, though.

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.

Seeing as an implementation of LinkedListStack only operates on the underlying list from one direction, a singly-linked list could be used in precisely as a doubly-linked list. A singly-linked list would be a little better because:

- The amount of space taken would be less (but still O(N)) due to a singly linked list only having to store one node-reference in each node rather than two
- The push, pop, and clear methods would be faster (but still O(1)) due a singly-linked list having fewer references to update

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?

Java's LinkedList exposes all of the methods (clear, isEmpty, getFirst, removeFirst, addFirst, and size) used in my implementation of LinkedListStack. As such, replacing the instance of DoublyLinkedList with an instance of Java's LinkedList would be trivial.

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

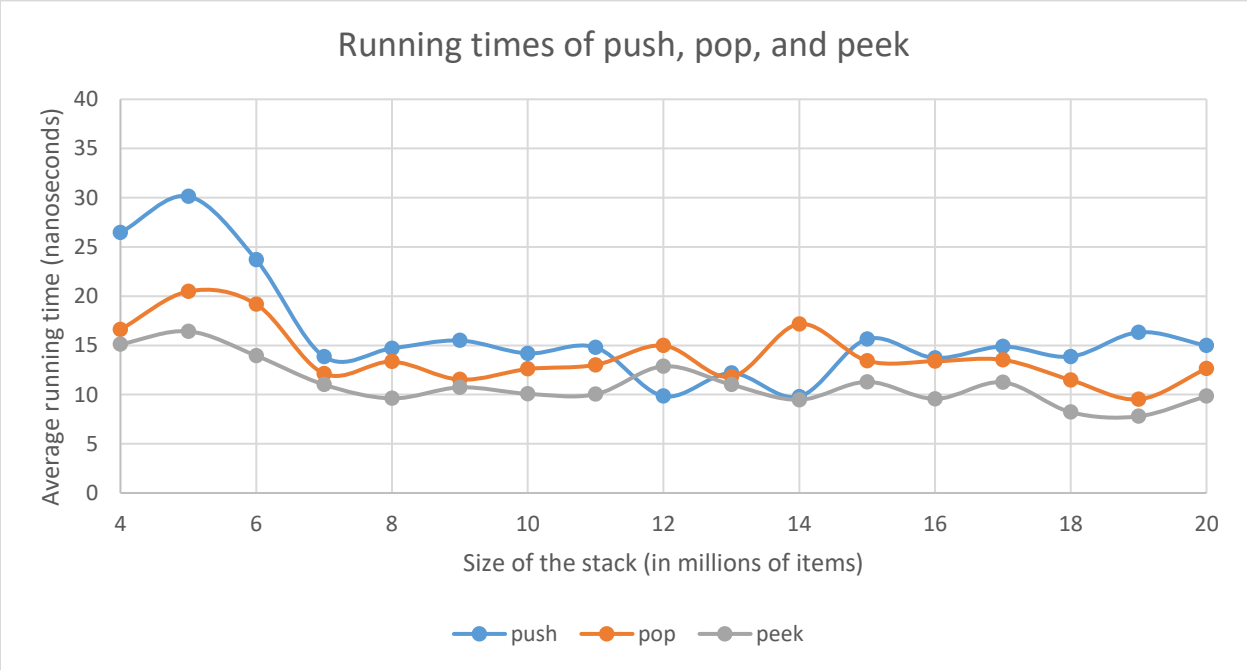
Developing LinkedListStack was incredibly easy. Using the existing DoublyLinkedList allowed me to implement all of LinkedListStack's methods with simple one-liners. I cannot imagine being any more efficient at implementing a data structure; I probably spent no more than ten minutes on this part of the assignment.

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.

To also keep track of the line and column number of the unmatched opening symbol, I would create a Symbol class. This class would include fields for character, line number, and column number. Instead of pushing opening characters to the stack, I would push a Symbol object created from the opening character as well the line and column numbers where the character was encountered. That way, should it come time to report an error, the line and column numbers of the opening character could be determined by simply accessing the fields of the Symbol 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.

Plotting the running times of push, pop, and peek, we get:



The push, pop, and peek methods are all O(1) as expected.

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

I spent a breezy five hours on this assignment.