

Analysis Document

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

I been working with only one partner. For the next assignment I will work with another partner.

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.

It will be the same since we only work with the last element added to the stack. If we were to take an element from the middle then LinkedList would be faster since it has index while DoublyLinkedList have to find the element since it depends from nodes. And if it were to pop the first element than DoublyLinkedList will be faster. In conclusion, working with stack we just have to pop the last element inserted and removing last element for SingleLinkedList and DoublyLinkedList takes same amount of time.

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 possible, because both can held types. Both LinkedList and DoublyLinkedList can add, remove, elements and both have same complexity if removing (pop) last element of list.

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

To understand how stack works it took me about 1 hour to understand, for the push method it took me a bit to understand like if to add at first or at last of the list; have to draw and understand where to add since in the picture that I see in slides to me it looks like we have to add at first but then I found out that we have to add at last. And that will be faster for us to peek and then to pop. With the pop method was fairly understandable, with the peek method it was understandable too since we just have to get the last element inserted to the list and if match found pop or do something else.

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 variables called openSymbol and closing-Symbol that way I will keep track of both open symbol and closing symbol. And whenever find the error I would get the openSymbol instead of the closing symbol. Other way to get the symbol I would create a variable called symbol which will get the number of column and line of where the symbol is and whenever find the error, call the variable. This variable will act as stack if variables match pop that way I will still have the other open symbols with its location.

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.

The running for these three method came to be as I was expecting since we're just adding elements to last and removing last element. For DoublyLinkedList removing last element is constant therefore pop method is constant and adding as last element also makes it constant. And the peek method is also constant since we're just getting the last element. What surprise me with these methods is that they run at different time, push takes less time than pop and peek, I thought they would be about the same, but as nanoseconds I can see that removing takes a bit more time than adding to last element.

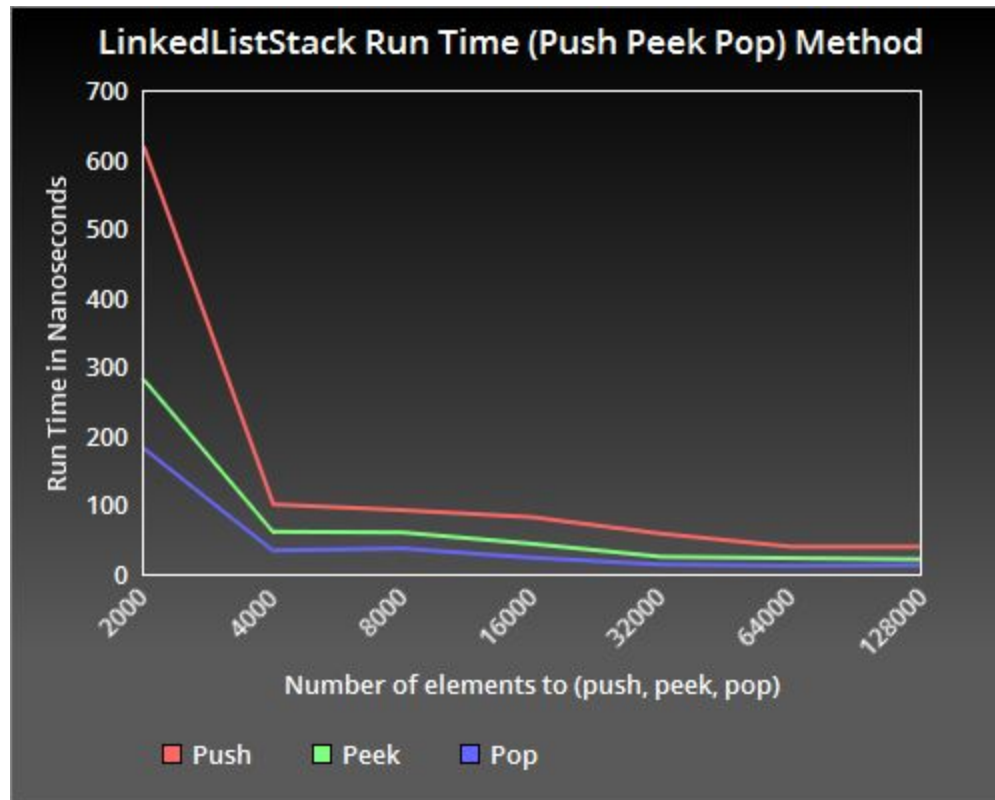
For this running time I have to change my DoublyLinkedList addLast, getLast, and removeLast methods or it will take more time than I was expecting. Without the change its complexity was $O(n)$ and is because my add last method start with head and has to go to the end to add the element so that take more time to add elements. What I change in my doublyLinkedList is just that instead of looking from head to last element I just create a new node and which will be the tail and so every time adding las element it will create a new node at the end of the list. I found that error but now is fixed.

Maycol Vilchez
U0832923
Assignment 07
10/19/16

```
public void addLast(E o) {
    if (isEmpty()) {
        tail = new Node(null, o, null);
    }
    else
    {
        Node tmp = new Node(tail, o, null);
        tail.mNext = tmp;
        tail = tmp;
    }
    size++;
}

public E getLast() throws NoSuchElementException {
    if (isEmpty()) {
        throw new NoSuchElementException();
    } else {
        return tail.mElement;
    }
}

public E removeLast() throws NoSuchElementException {
    if (isEmpty()) {
        throw new NoSuchElementException();
    }
    // if size == 1 head gets null
    if (size == 1) {
        Node tmp = tail;
        tail = null;
        size--;
        return tmp.mElement;
    } else {
        Node tmp = tail;
        Node a = tail.mPrev;
        tail = null;
        tail = a;
        size--;
        return tmp.mElement;
    }
}
```



7. How many hours did you spend on this assignment?
For this assignment I spent about 6 hours to finish it.