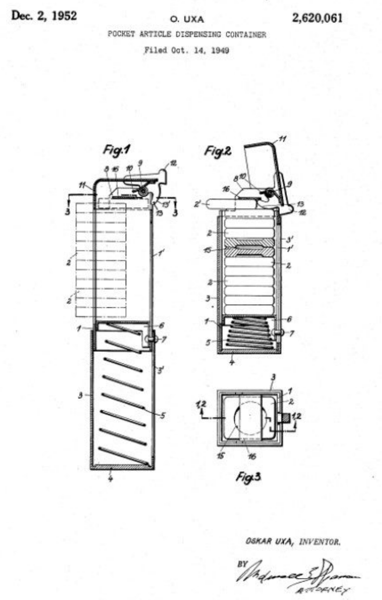

PA4 Inside Stacks and Queues

Due: Wednesday, October 19th, 11:59PM

Submission:

Six java files:

- DynamicArray.java, LinkedList.java
- ArrayStack.java, ListStack.java
- ArrayQueue.java, ListQueue.java

Overview

The purpose of this assignment is to implement a stack and a queue in two different ways and to see how this affects the performance of typical operations such as push and pop.

Requirements

Many data structures are built with either arrays or linked structures. You will implement a stack and a queue in both of these ways and then compare their performance.
Two files are provided, StackInterface.java and QueueInterface.java which define the interfaces for a stack and queue. Six classes need to be implemented:

DynamicArray.java - a dynamic array, an array that can be resized.
LinkedList.java - a singly linked list.
ArrayStack.java - implementation of the StackInterface using *DynamicArray*.
ListStack.java - implementation of the StackInterface using *LinkedList*.
ArrayQueue.java - implementation of the QueueInterface using *DynamicArray*.
ListQueue.java - implementation of the QueueInterface using *LinkedList*.

In addition to the methods defined in the interfaces, the following methods need to be implemented:

- *toString*
- *equals*
- *copy constructors*

The *toString* method should return a string in the format: ”{0,1,2,3,4,5}”.
There are no spaces in the string and if there are no elements at all, then print ”{}”.
In the above example, ’0’ would be at the bottom of the stack and ’5’ would be at the top.
If it was a queue, ’0’ would be at the front of the queue and ’5’ would be at the back.

Two stacks (or queues) are *equal* if their sizes and all of their elements are equal.

For each method in the stack and queue interfaces that you implement, put in the comment section the time complexity for the method, using big-O notation.

Only simple error handling is used in this PA. Athough later we will use exceptions, here the ‘error’ return values are specified in the comments of the interfaces.

You should not print anything in any of your classes. i.e. do not call ’System.out.println’ anywhere in your code. You can use print statements to debug but be sure to remove them all before submitting your files.