Programming Project 3 - Due: Sunday, October 16 at 11:59 PM

Having learned of ADTs and algorithmic complexity, you can't wait to see how this works in practice. You will implement a simplified list interface. The list interface has been given to you in ISimpleList.java.

Once implemented, you will use your list implementations to store data read from a file and do a simple complexity analysis of that code.

Part 1:

You will write a class ArrayBasedList.java which implements the ISimpleList interface.

Your class will store the data in an array with an **initial capacity set to 2** and it will have a “double up” growth policy. There is no automatic capacity reduction.

All your implemented methods should be as time and space efficient as possible. **Give the time big O for each in the method header.**

Part 2:

You will write a class SinglyLinkedList.java which implements the ISimpleList interface.

Your class will store the data in a singly linked list. All your implemented methods should be as time and space efficient as possible. **Give the time big O for each in the method header.**

Part 3:

You will implement the body of the two `getFileContentsInReverse(A/B)` methods in Project3.java. Your implemented methods should be as time efficient as possible. You will provide a time complexity analysis of each method assuming a file with n lines of text. **Give the big O and explain clearly how the big O was obtained (~ 3-5 lines brief and to the point analysis)**

Hints:
- Make sure you review week 5 notes
- For Part 1, make a method ensureCapacity which doubles up the array when needed
- For Part 2, draw simple box and arrow node diagrams on paper before you implement your methods

Deliverables:

You should submit a zip file named project3_first_last.zip (where first and last are your first and last name) containing **ONLY the files below.**

ArrayBasedList.java
SinglyLinkedList.java
Project3.java
report.txt : a text (not Word, Power Point, ...) file containing:
A 1 to 5 lines paragraph from you saying “I have tested this program and there are no known issues.” if you believe that to be the case, or a brief description of known issues in case your program has known problems or you could not fully implement it.

How you get points:

Part 1	40 points
Part 2	40 points
Part 3	20 points

How you lose points:

- You do not follow the given directions and decide to make changes “for fun”. Specifically, **do not change the interface or the skeleton code given to you**. If you do, the code you submit will not compile. Remember that you are not submitting the interface, only the classes that implement it.
- **If your implementation is inefficient. Your solution should be efficient to a level seen in class for this ADT.**
- If your implementation is not well organized and clear.
- If any of your code prints anything at all on the console. **Remove all your print outs, debug statements, etc. Clean up your code and do not leave clutter behind.**
- If your code has no comments where needed. **Comment your code appropriately. Brief and to the point.**
- If you submit your whole workspace or executable files. **Submit only the files the project asks for.**