

## EECS 233 Homework 3

### General requirements:

- Due at 11:00 PM on the posted due date.
- Include your name and network ID as a comment at the top of all of your programs.
- Create a typed document (.txt or .pdf) with answers to the questions.
- Upload your document and all .java files as a .zip file to Blackboard. Do not use other formats such as .rar.
- All work should be your own, as explained in the Academic Integrity policy from the syllabus.

**Instructions:** This assignment requires you to analyze the run times of two data structures provided with the textbook (and posted on Blackboard): the `MyArrayList` class and the `MyLinkedList` class. For both:

1. Display the  $N$  value you used and the run time (in seconds) for the following operations:
  - a. Add the values 1, 2, 3...,  $N$  with each new value added to the end of the data structure.
  - b. Add the values 1, 2, 3...,  $N$  with each new value added to the beginning of the data structure such that the final data is reverse-ordered ( $N, \dots, 3, 2, 1$ ).
  - c. Add the values 1, 2, 3...,  $N$ . Then remove each value from the end of the data structure without an iterator until all of the values are removed. Note: you will only analyze the “remove” operation.
  - d. Add the values 1, 2, 3...,  $N$ . Then remove each value from the beginning of the data structure without an iterator until all of the values are removed. Note: you will only analyze the “remove” operation.
  - e. Add the values 1, 2, 3...,  $N$ . Then remove each value from the beginning of the data structure with an iterator until all of the values are removed. Note: you will only analyze the “remove” operation.
  - f. Add the values 1, 2, 3...,  $N$ . Then compute the sum of all the values. Note: you will only analyze the summation.
  - g. Add the value 0 a total of  $N$  times (size  $N$ ) with each new value added to the end of the data structure.
2. Explain in a typed report how the results in #1 compare to the expected  $O(N)$  time discussed in class and the textbook. Discuss whether the run time for each operation is reasonable, as compared to the other operations for the same data structure and for the other data structure.

*Sample output:* Below is an example of partial program output (not complete):

```
MyArrayList...  
Add to end: N = 10000000, t = 4.144 sec
```

*Tip (how many programs?):* Use as many (or as few) program files as you wish. One example is to have one file for the array list tests and one file for the linked list tests. As demonstrated in lecture, the most straight forward approach for many students may be to copy/paste the classes from the sample code into new programs. However, you will want to move the main method inside the primary class, as demonstrated in lecture. You can either keep the original class name (overwriting the original sample program), or you may change the class name. Interested students can find another tip below regarding the question “Why won’t sample programs run?”

*Tip (empty arrays vs. full arrays):* Be careful to start with an empty array each time you start a new “add” test. You shouldn’t be adding to full array from the previous test. In contrast, tests for removing and summing require a full array to begin with. However, you should only test the run time for the “remove” or summation operations. For “remove” and summation, do not include the time required for creating the array.

*Tip (what  $N$  value should I use?):* Use  $N$  values that allow reasonable run time comparisons. Run times of 0 or 1 msec are not appropriate. You may use different  $N$  values for each operation.

*Tip (iterators):* There are two remove methods: a general “remove” and an iterator-based “remove”. Performing a remove without an iterator requires a loop and possibly the “size” method. Removing with the iterator requires

three steps: (1) declaring the iterator, (2) using the “next” method with iterator, and (3) using the “remove” method with iterator. Look carefully at MyLinkedList.java for an example.

*Tip (why won't sample programs run?):* As explained in class, you cannot just click “Run” to run the sample programs from the textbook. This is because the main method is not within the main class. If you compile MyArrayList.java, notice that it creates MyArrayList.class and TestArrayList.class. For the assignment, you can just copy/past the code into a new program, as suggested above. However, if you wish to use a more elegant approach, your two options are: (1) write a separate program that uses the compiled (but separate) .class file, or (2) run the “Test” class separately (e.g. TestArrayList) from the command line (java TestArrayList) or from your IDE.

*Grading rubric:*

Item	Points	Comments
1: declaration/initialization of data structure	20	10 for ArrayList, 10 for LinkedList
1: loop design	20	10 for ArrayList, 10 for LinkedList
1: performing primary operation (add, remove, etc.)	20	10 for ArrayList, 10 for LinkedList
1: timing and displaying results	20	10 for ArrayList, 10 for LinkedList
2: Discussion of O(N) results.	5	
2: Discussion of ArrayList vs. LinkedList	10	
Program design (arrangement of files, classes, etc.)	5	
<i>Total</i>	100	