UNIVERSITY OF VIRGINIA, DEPARTMENT OF COMPUTER SCIENCE

Linked Lists - Linked List Analysis

Nada Basit and Mark Floryan

January 27, 2022

1 SUMMARY

The goal of this homework is to write a report comparing the runtimes of the various methods of your LinkedList class. The requirements for this report are VERY simular to the previous reports.

1. Write some test code to time the various methods of your Linked List
2. Run a small experiment timing some of the methods (see below)
3. Write a report summarizing and analyzing your findings
4. **FILES TO DOWNLOAD:** None
5. **FILES TO SUBMIT:** LinkedListAnalysis.pdf

1.1 TIMING YOUR CODE

For this homework, we'd like you to write a tester that executes each of your methods in your Linked List and times how long each takes. We are doing this to gain some intuition about which operations are slow and which operations are fast. You should consider the following when timing your code:

- Each method, if invoked only once, will run VERY quickly. Thus, instead we are going to invoke each method of our Linked List many times (you can adjust this number) to get a better measurement. We want the total time for each operation to be at least 1 second.

- Likewise, if methods are invoked on very small lists (e.g., finding an element in a list of size 5), then the operations will be VERY fast. Thus, make sure to make the lists large enough until you start to see a slowdown. Part of your investigation involves figuring out how big these lists must be before a slowdown is observed.

- Make sure you are ONLY invoking the method of interest when testing that method. Don't inadvertantly invoke find() while testing insertAtTail() as that will throw off your measurement.

To actually time the code itself, Java provides a system call that returns the number of milliseconds between now and January 1, 1970. The line of code is:

```java
/* In java.lang.System */
public static long currentTimeMillis();
```

To use this in order to time some code, one can make this method call to get a timestamp, then run some code, then take another timestamp (using currentTimeMillis). The difference between the two timestamps is the number of milliseconds that have passed between the two method calls. So, you can test a method by doing something like:

```java
import java.lang.System;

//Ready to begin measuring
int t1 = System.currentTimeMillis();

/* Lot's of code that runs a method on my list a bunch of times here */

//Difference in now and first time stamp is total time taken
int time = System.currentTimeMillis() - t1;
```

For this experiment, you should time your Linked List calling each method i times on a list of size n. You may play around with i and n until you find numbers that work well, but they should be the same for each method. You should test the following methods:

```java
/* You will test these methods: */

public void insertAtTail(T data);
public void insertAtHead(T data);
public void insertAt(int index, T data);
public void insert(ListIterator<T> it, T data);
public T removeAtTail();
public T removeAtHead();
public T remove(ListIterator<T> it);
public int find(T data);
public T get(int index);
```

1.2 Report

Summarize your experiment and your findings in a report. Make sure to adhere to these general guidelines:

- Your submission MUST BE a pdf document. You will receive a zero if it is not.
- Your document MUST be presented as if submitted to a professional publication outlet. You can use the template posted in the course repository or follow Springer's guidelines for conference proceedings.
- You should write your report as if it is original novel research.
- The grammar / spelling / professionalism of this document should be sound.
- When possible, do not use the first person. Instead of "I ran the code 60 times", use "The code was executed 60 times...".

In addition to the general guidelines above, please follow the following rough outline for your paper:

- **Abstract**: Summarize the entire document in a single paragraph
- **Introduction**: Present the problem, and provide details regarding the two strategies you implemented.
- **Methods**: Describe your methodology for collecting data. How large were the lists, how many times did you call each method when timing, how many executions, how you averaged things, etc.
- **Results**: Describe your results from your execution runs.
- **Conclusion**: Interpret your results. Which methods were fast and which were slow? Did this surprise you? Does this align with the theoretical runtimes of those methods?

How large did the lists need to get before you witnessed a slowdown?

Lastly, your paper MUST contain the following things:

- A table (methods section) summarizing the different experimental groups and how many execution runs were done in each group.
- A table (results section) summarizing each experimental group and the averages / std. dev. for each (as well as any other data you decided to collect).
- Some kind of graph visualizing the results of the table from the previous bullet.