Homework 3 - Due Saturday, June 18th

Eclipse Suggestion

We suggest you create a Homework3 Java project in Eclipse and unzip hw3.zip into the src folder.

Completing DoublyLinkedList

We have provided a skeleton for a DoublyLinkedList class that must be completed. All of the methods except the constructor, size, getSentinel, and isEmpty must be completed. The method spliceAfter is extra credit.

The particular implementation we will be using involves the use of a sentinel node. This is a dummy node that doesn’t store any value. If the list is empty, the next and prev of the sentinel node point back at the sentinel node. When the list is not empty, the last node’s next reference and the first node’s prev reference both point at the sentinel node. Furthermore, the next reference of the sentinel points at the first node, and the prev reference of the sentinel points at the last node. The purpose of a sentinel node is to make the implementation much cleaner since we will never have to deal with null references. Furthermore, we no longer need to maintain the error-prone references to the head and tail since they are easily accessible via the sentinel. The sentinel also may make our implementation faster since it allows us to remove many branches (i.e., if statements) from our code. The price we pay is the memory of a single node, but this is almost always worth it.

Here are some suggestions for implementing the DoublyLinkedList.

1. Implement addAfter, addBefore, and removeNode first. Once these are written almost every other method becomes trivial except getNode, equals, and spliceAfter. For example, once you have coded addAfter, the method addFirst is a single very short line of code.

2. equals: Model this after our SinglyLinkedList equals method we will discuss in class.

3. Draw pictures to make sure you have all of your links correct when implementing addAfter, addBefore, and removeNode.

4. It could be helpful to write your own toString to help you debug your list. This isn’t required.

Testing DoublyLinkedList

We will be using a library called JUnit to test our code. Included in the hw3.zip file is the jar files for the JUnit library (version 4.12). A jar (Java archive) simply contains a bunch of class files (and potentially other files). To use it do the following in Eclipse:
1. Right-click on your Homework3 project in the package explorer and go to properties.

2. Select Java Build Path, and then select the Libraries Tab.

3. Click Add External JARs, and look in the folder you unzipped hw3.zip into. Select all of the JAR files.

4. If you did this correctly all 4 of the JAR files should now be listed under libraries tab (along with JRE System Library). [Technically you didn’t need to include the source jars, but it is ok.]

5. Assuming everything went well, the files DoublyLinkedListTests.java and DoublyLinkedListListRandomTests.java should both compile. Right click on DoublyLinkedListTests.java in the package explorer and select Run As, JUnit Test. The tests should run but most of them should fail. You can do the same for DoublyLinkedListListRandomTests.

You have correctly finished the assignment when all the tests (except the splice tests) have passed. You have the extra credit correct when the splice tests have passed also. There are no other tests for this assignment.

What you submit

You should submit DoublyLinkedList.java a single zip file that is uploaded to NYU Classes. Please name your zip file hw3_yourNetID.zip

How to Test If You Aren’t Using Eclipse

If you want to run all the JUnit tests but you aren’t using Eclipse you can use the following command-line from the bin folder (place where the .class files are) of your Homework3 project (all on a single line). Windows Version:

```java
java -cp .;./junit-4.12.jar;./hamcrest-core-1.3.jar
   org.junit.runner.JUnitCore DoublyLinkedListTests
```

Mac Version:

```java
javac -cp .:./junit-4.12.jar:./hamcrest-core-1.3.jar *
```

The above will only work if your jar files are also in your bin folder. You can then do the same with DoublyLinkedListTests replaced by DoublyLinkedListListRandomTests.

If you are also unable to compile your files and wish to do this on the command-line use the following from your src directory (where you unzipped hw3.zip). Windows Version:

```java
javac -cp .:./junit-4.12.jar:./hamcrest-core-1.3.jar *
```

Mac Version:

```java
javac -cp .:./junit-4.12.jar:./hamcrest-core-1.3.jar *
```