COSC 222 Lab 7 – Timing Tests

In this lab you will test the runtime of some `String` operations (`.add()` `.contains()` and `.remove()`).

To record the runtime of a method, use something similar to this code snippet found on: https://stackoverflow.com/questions/180158/how-do-i-time-a-methods-execution-in-java

```
long startTime = System.nanoTime();
codeToTime();
long endTime = System.nanoTime();

long duration = (endTime - startTime);  //divide by 1000000 to get milliseconds.
```

You are given a custom implementation of a Trie (called `LowerCaseTrie` because it only maintains 26 references per node for the 26 lowercase letters in the English language). This data structure works like a Set: that is, it does not allow duplicated items.

Write unit tests to sufficiently test this class for correctness. You should make 6 tests that test .add and .contains together, and then 6 more tests of scenarios of using .add followed by .remove followed by .contains. Consider adding words which are prefixes of other words already added. And also testing for words which are prefixes of some added words, even if those prefixes are not added. Also consider adding words multiple times, removing the word, checking contains, adding the word again, checking contains. Report any errors in the data structure if you find any, but you do not have to fix the errors.

You will be given +0.5 points for each meaningful test, up to 6 points for unit testing.

The remaining 4 points will be given for performing a time test of this LowerCaseTrie compared to the other collections: LinkedList, HashSet, Skiplist.

See TimeTests.java for details. You are to measure (in milliseconds) the time required to perform 200,000 operations of various types for the above-mentioned data structures. The starter file reads in the English dictionary file and stores the >300k words in an ArrayList, and then you are expected to use .add() on a 200k random words (it is okay if you end up calling .add() on repeating words). You should get enough data to fill in the following table entries (although you do not have to present your findings in a table – you can present your findings in a block-comment in your .java file).

Data Structure	200000 .add()	200000 .remove()	200000 .contains()
Trie			
LinkedList			
HashSet			
SkipList			

Submit your `TestTrie.java` and `TimeTests.java` files.