CS526
Homework Assignment 8

Due: 11/25

The goal of this assignment is to give students an opportunity to observe differences among three data structures in Java – HashMap, ArrayList, LinkedList – in terms of insertion time and search time.

Students are required to write a program that implements the following pseudocode:

```
create a HshMap instance myMap
create an ArrayList instance myArrayList
create a LinkedList instance myLinkedList

Repeat the following 10 times and calculate average total insertion time and average total
search time for each data structure

    generate 100,000 random integers in the range [1, 1,000,000] and store them in
    the array of integers keys[ ]

    // Insert keys one at a time but measure only the total time (not individual insert
    // time)
    // Use put method for HashMap
    // Use add method for ArrayList and Linked List

    insert all keys in keys[ ] into myMap and measure the total insert time
    insert all keys in keys[ ] into  myArrayList and measure the total insert time
    insert all keys in keys[ ] into myLinkedList and measure the total insert time

    generate 100,000 random integers in the range [1, 2,000,000] and store them in
    the array keys[ ] (or in a different array).

    // Search keys one at a time but measure only total time (not individual search
    // time)
    // Use containsKey method for HashMap
    // Use contains method for ArrayList and Linked List

    search myMap for all keys in keys[ ] and measure the total search time
    search myArrayList for all keys in keys[ ] and measure the total search time
    search myLinkedList for all keys in keys[ ] and measure the total search time
```

Print your output on the screen using the following format:

```
Number of keys = 100000
```

```
HashMap average total insert time = xxxxx
ArrayList average total insert time = xxxxx
LinkedList average total insert time = xxxxx

HashMap average total search time = xxxxx
ArrayList average total search time = xxxxx
LinkedList average total search time = xxxxx
```

You can generate *n* random integers between 1 and N in the following way:

```
Random r = new Random(System.currentTimeMillis() );
for i = 0 to n - 1
    a[i] = r.nextInt(N) + 1
```

When you generate random numbers, it is a good practice to reset the seed. When you first create an instance of the Random class, you can pass a seed as an argument, as shown below:

```
Random r = new Random(System.currentTimeMillis());
```

You can pass any long integer as an argument. The above example uses the current time as a seed.

Later, when you want to generate another sequence of random numbers using the same Random instance, you can reset the seed as follows:

```
r.setSeed(System.currentTimeMillis());
```

You can also use the *Math.random*( ) method. Refer to a Java tutorial or reference manual on how to use this method.

We cannot accurately measure the execution time of a code segment. However, we can estimate it by measuring an elapsed time, as shown below:

```
long startTime, endTime, elapsedTime;
startTime = System.currentTimeMillis();
// code segment
endTime = System.currentTimeMillis();
elapsedTime = endTime - startTime;
```

We can use the *elapsedTime* as an estimate of the execution time of the code segment.

Name the program *InsertSearchTimeComparison.java*.

Documentation

You need to submit a separate documentation. In the documentation, you need to write your conclusion/observation/discussion of this experiment. As usual, you must include sufficient inline comments within your program.

Grading

There is no one correct output. As far as your output is consistent with generally expected output, no point will be deducted. Otherwise, up to 4 points will be deducted for wrong insert times and up to 4 points will be deducted for wrong search times. If your conclusion/observation/discussion is not substantive, points will be deducted up to 4 points. If there are no sufficient inline comments, points will be deducted up to 4 points.

Deliverables

Submit the *InsertSearchTimeComparison.java* file to Blackboard. If you have multiple files, combine them into a single archive file, such as a *zip* file or a *rar* file, and name it *LastName_FirstName_hw8.EXT*, where *EXT* is an appropriate file extension (such as *zip* or *rar*). Upload it to Blackboard.