CMPS 261: Project 3

Due Date: Check on Moodle

Instructions

1) You are allowed to discuss the problem and solution design with others, but the code you submit must be your own. Your solution must include the certification of authenticity as given in the course syllabus.
2) Create project in NetBeans 8.1 using Java 8
3) Name your project as P3_<your clid>
    a. Example: if clid = abc1234 then project name = P3_abc1234
4) Follow the instruction on Moodle under "Submitting to Moodle".

General Requirements:

- Project submissions not including a Certification of Authenticity will not be graded until such a written and signed submission is provided by the student.
- Code, even if substantial, that fails to compile will not receive a grade higher than 25%
- Code, even if substantial, that fails to run will not receive a grade higher than 50%
- A penalty of up to 10% may be assessed for lack of required documentation, lack of descriptive identifiers, lack of indention of blocks of code, failure to follow naming requirements of the project and associated tar archive.
- Failure to submit the complete NetBeans project may result in a penalty of up to 10%.

Problems

1) (100 Points) Write a Java project that implements selection, bubble, insertion and merge sort algorithms that sort lists of random doubles in the range [0, 1000). Use array objects or objects created from API classes that implement java.util.List<E> to contain the lists. Using the sorting algorithm implementations, compare execution time performance of these sorts for random double lists of the following lengths:

- 100,000

- 200,000

- 400,000

- 800,000

Output should be in the form of a table of number of elements sorted and type of sort. Hint: Use System.out.printf.

2) (100 Points) Read the words found in the file "fstein.txt" into an array object or an object created from an API class that implements java.util.List<E>. Then

- sort the list using merge sort, then remove all duplicate words.
- sort the list using insertion into an initially empty list. Do not insert duplicate words.

Output the execution time performance of these two processes and the number of words remaining after duplicates have been removed from the list.

Notes:

- If you choose to use an array, you will need to allocate at least 17,077 elements (this is the number of words in the given "fstein.txt" file).
- When inserting into the initially empty list, do not do this when reading directly from the file, as the input time will skew your results.

Programing Tips:

1. Test your sorts with short lists first.
2. *double Math.random()* returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.
3. Be careful not to sort a sorted list.
4. *System.gc();* calls immediate garbage collection.
5. If your JVM needs more memory for execution than the default 128 MB:

   (a) In NetBeans, right click on the project name (under the Projects tab) and select Properties, then select Run. By Arguments, add "-Xms*size*m" (without the quotes), where size is the maximum number of megabytes of memory to be allocated. Example: `-Xms256m`

   (b) From the command line, add the argument "-Xms*size*m" (without the quotes), where size is the maximum number of megabytes of memory to be allocated, to the call to "java". Example: `java -Xms256m etc`

6. To time the execution time of a task, use the following code template:

```
long startTime = System.currentTimeMillis();
... perform task ...
long endTime = System.currentTimeMillis();
long executionTime = endTime - startTime;
```

Additional Requirements

[1] All output must be labeled.

[2] Identifiers must be descriptive, i.e. must self-document. The only exception granted is in the case of a "for variable", that is a variable created as a simple counter as in the control variable in a "for" loop statement.

[3] Indention of all code blocks (compound statements, anything in braces), including single statements following selection or while statements, is required. NetBeans will do this fairly automatically as you type if your syntax is correct. In NetBeans, ALT-SHIFT-F will re-format a whole file if your syntax is correct.

[4] The main ".java" file [the one with the method *public static void main(String[] args)*] of your project must contain this minimal documentation:

```
// Your Name
// Your CLID
// CMPS 261
// Program Description: description of actions of code
// Certificate of Authenticity:   (Choose one of the two
following forms:)

     I certify that the code in the method functions
     including method function main of this project are
     entirely my own work.

     {or}

     I certify that the code in the method functions
     including method function main of this project are
     entirely my own work., but I received some assistance
     from {name}.  Follow this with a description of the
     type of assistance. (For example, if you consulted a
     book, and your solution incorporates ideas found in the
     book, give appropriate credit; that is, include a
     bibliographical reference.)  Note:  You do not have to
     list the text, the author of the text or the
     instructor's examples.
```

[5] Each class method must include documentation immediately above the method header in the form:

```
/* method name
 * a brief definition
 * pre-conditions
 * post-conditions
 */
```

Example:

```
/* RemoveElement
 *
 * Removes an element from a list of strings stored in an
 * array by compacting the list.  First locates the list
 * element to be removed, then copies each list element
 * located at an array index larger than the array index of
 * the list element to be removed to the next lower index,
 * proceeding from lowest to highest array index.
```

```
 *
 * Pre:   The array passed is populated, the item parameter
 *             contains a value that is not the null string and
 *             the length of the list passed is > 0.
 *
 * Post:  If the item passed was found in the list, the
 *        matching list element has been removed, the list
 *        compacted and the length of the list reduced by 1 and
 *        returned.
 */
public int RemoveElement(String[] list, String item, int len)
...
```