**Design your own sort - Activity** *(revised April 2014)*

Your task is to write a Java program that will:

1. Import a text file called **source.txt**

The file source.txt contains an unknown number of integers. Write code to:

* + Check how many lines are in the text file
  + Create an appropriately sized 1-dimensional array
  + Load the integers from the file into the array

1. Create a **long** variable called **steps** and set it to zero. (Long is just like an integer, but capable of holding much larger numbers).
2. Write a **method** called **mySort()** that takes an array of integers as a parameter, and sorts them into numerical order, and returns the sorted array.
   * You will need to move items into and out of your array in order to swap them. Remember, you can't just "swap" two elements - you have to move the first element into a temporary variable, move the second element into the first slot, and then move the first element from the temporary variable into the second slot. *Every time you perform an* ***assignment operation*** *or a* ***comparison (if) operation****, increment your****steps****variable*. You are trying to track how many total operations you have to perform to get the data sorted.
   * You may NOT use Arrays.sort() or any other predefined sorting method. You may NOT borrow code for this assignment.
3. Import the provided class **Timer.java**. Examine the API. Implement the Timer into your code so that you can time how long it takes for your sort to perform its operations.
4. Write the sorted data into a new text file called **sorted.txt**. The FIRST line should have the value of your steps variable, and the sorted data will be below this.
5. How long did it take your program to sort 5,000 values? 25,000 values? Note that this might take a LONG time. You should use the Timer class, as described in step 4, to accurately time the process. **Record your results**
6. If your sort took more than 15 minutes, identify possible ways that

The text files provided consist of one integer per line with no extra white space. The first file contains 100 random integers between 0 and 999, the second file contains 5,000 and the third file contains 25,000.