

## CS2050 Fall 2017

### Homework 6

**Due: Thursday, Oct 19, 2017**

More practice with sorting and timing different data types.

#### Requirement:

Put your name and assignment number at the top of each program in comments

If you're using an IDE that creates a boilerplate (e.g. NetBeans) make sure your name and assignment is included at the top

#### Part 1 - Integer Sorts

1. Create a Mergesort class. that takes an array as input and returns a sorted array as output.
2. In HW6.java, write a user-input routine that asks the user for two integers. The first integer will be SIZE, the number of integers in the array to sort. The second integer will be BIG, which is the largest number (range) that the random numbers can come from. Example, if I want 3 random numbers from the list 1,2,3,4,5,6,7, the SIZE is 3 and BIG is 7. Make sure you have the appropriate exception-handling code to keep the user from getting into trouble. For example, SIZE < BIG.
3. In HW6.java, generate the input array of random integers of SIZE with values between 1 and BIG. Use the Random function.
4. Using the timing functions from HW5 to time the Mergesort and built-in sort for comparison. Print the times for each run.
5. Test #1 - Use SIZE = 10 and BIG = 100 and print the list before and after the sort to verify the sort works.
6. Test # 2 and #3 use larger numbers for SIZE and BIG. **Do NOT print the list of numbers, just the times.**

#### Part 2 - Real Number Sorts

Do the same as Part 1 but with floating point numbers.

#### Part 3 - String Sorts

Do the same as Part 1 but with short strings (6 characters). BIG will not be used, just SIZE. Upper & lower case characters and numbers are valid. This means write a random string generator.

#### Notes:

- There should be only one Mergesort class. Do not create 3 different classes.
- Create the class files with stubs. Compile and verify at each step the program works. Design and build incrementally. Once you have something working, go on to the next method. Don't try to do everything at once.
- The only user involvement is in HW6.java class.
- Do not hard-code.
- Avoid duplication of code. If the same code is used twice or more, make it a function.
- Remember, only **HW6.java** will contain the **main()** method.
- Make sure this program will compile and run without depending on the IDE you use. I test all code at the command line using Java 8.

#### To Turn In:

Upload to your github account the following:

- 1) All files for this program
- 2) Plain text output.