

Points: 100

Due Date: Friday, November 30th, 2018 @ 11:59pm

do not, Do Not, DO NOT turn in any data files with your assignment.

- **Multiple data files this large may overwhelm Blackboard**
- **There will be a 100% point deduction if you turn in any data files with this assignment.**

For this assignment you may use any classes from the online version of the code for the textbook. The textbook code is in a jar (Java archive) file which uses the zip file format. One easy way to deal with a jar file is to rename the file with a .zip extension, use Windows Explorer to browse the zip archive, and copy and paste the necessary class files into your NetBeans project directory.

Note: I still believe that there is a learning benefit from transcribing the code from the textbook but you may copy and paste the code from the online jar file.

Note:

- The task is to sort a file of one billion numeric values.
- You may need to reduce the number of values in order to get a “reasonable” sort time.
- **Write your code so that the size of the data set can be specified by the user as the first instance variable in the main method.**
- Make sure that your program will work with a minimum of 100 data items and a maximum of one billion data items.

Task 1:

- Create a new NetBeans project named Lab113
- Write a method that:
 - creates a data file named “C:\\data\\data.txt” on your drive, and
 - writes one billion (1,000,000,000) numeric values of type int to the file,
 - writing one value per line.
 - These values should be in the range:
 - $0 < x < \text{Integer.MAX_VALUE}$.
 - **Output size of the data set (N) and the time it takes to create this data file.**

- Notes:
 - on my office computer this process takes between 10 and 12 minutes without doing anything “special.”
 - The file size is about 11GB.

Task 2:

- Write a method that sorts the data you wrote to the “C:\\data\\data.txt” file in ascending numeric order and save this sorted data to a file named “sortedData.txt”.
- The data set should be too large to fit into RAM so your sorting algorithm will need to perform an external merge sort.
- **Have your program output:**
 - **The size of the data set (N)**
 - **The time it takes to generate the random unsorted file**
 - **The size of each block, use a minimum of 5 blocks**
 - **Even if your computer has enough RAM to sort the data with a single internal sort you MUST still sort the data using an external merge sort.**
 - **Regardless of the size of the input data, your program must divide the data into a minimum of five blocks.**
 - **The time it takes sort each block**
 - **The total time it takes to sort the entire file.**
- Hints:
 - you may use any of the classes that you created or imported from the textbook code
 - you will want to use an efficient sort method to sort each block
 - you will want to make the blocks as large as possible, ideally your program should dynamically determine the best block size
 - regardless of the maximum block size on your machine write your program so that it breaks the data up into **a minimum of 5 blocks.**
 - you may want to review Java Illuminated sections 11.3 and 11.4 (pdf available in Course Content area)
 - you will want to read section 15.4 in the textbook.

Task 3:

- **Output the first 100 lines of the unsorted data from the “data.txt” file**
- **Output the first 100 lines of the sorted data from the “sortedData.txt” file**

Turning in your assignment:

- **Make sure that all of your code is properly documented.**
- Turn in your assignment using the standard method.
- Create a Word document and copy and paste the following into the document:
 - o A screenshot of your output
 - o Your client class
 - o Any other .java files written or modified for this assignment.
- Export your NetBeans project to a zip archive.
- Turn in the Word document, and zipped project file as separate files in a single Blackboard submission.
- **do not, Do Not, DO NOT** turn in any data files with your assignment.
 - o **Multiple data files this large may overwhelm Blackboard**
 - o **There will be a 100% point deduction if you turn your data file(s) in with this assignment.**