1 Instructions

For this assignment you will be submitting a report of the timing analysis for the faster comparison sorts, there is no coding or code submission for this assignment. The report is to be a single PDF document that includes all the requested timing data, graphs, and commentary. As usual you will submit your work through the MyClasses page for this class.

2 Exercise

This homework has no coding portion, just timing analyses of the faster sorts we discussed in class, specifically the quick sort, merge sort, and the sort that is included in the algorithm library. All the code you need is included in the zip file that is with this assignment. You can find the same code in the class code archive, in `ChapterX/SearchingSorting/Sorting`. This code uses a different method for getting the system time, it uses a macro in place of using the chrono library. The chrono library has more options but this macro is easier to use if you just want time in seconds with a microsecond resolution.

Recall from the previous discussions on timing. All timings are to be on the same hardware and it should be using a system where the operating system is running directly on the hardware and not through a virtual machine. The HPCL Linux Lab would be the best place to do these timings.

1. Use the main timing program to do the following for the quick sort, merge sort, and the sort in the algorithm library.

   (a) Sort integer arrays with sizes 1,000,000, 2,000,000, 5,000,000, 10,000,000, 15,000,000, 20,000,000, 25,000,000, 30,000,000, 35,000,000, 40,000,000, 45,000,000, and 50,000,000. For each case, use the maximum element size of `RAND_MAX` by selecting 0 for this option.

   ```
   Input the maximum element size (0 for RAND_MAX): 0
   ```

   (b) Using a spreadsheet, such as LibreOffice Calc or Excel, plot the sort times verses the number of data items. Put all three lines on the same chart. You will want to use an XY scatter chart with the sizes on the x-axis and times on the y-axis.

   (c) In a word processor, you can use LibreOffice Writer, Word, or any word processor that will allow you to export the document to a PDF file. LibreOffice Writer has a toolbar button for this. In the document include tables of all the data you acquired from the runs you did above, the charts you created from the data and the answers, in complete sentences, to the following.

   - How do the times for these three algorithms compare?
   - Do the curves have approximately the same shape? That is, do they increase along a parabolic-like curve, are they all straight lines, do they have the same types of curves in about the same places?

- What do you think is accounting for the differences?

2. Use the main timing program to do the following for the quick sort, merge sort, and the sort in the algorithm library.

   (a) Sort double arrays with sizes 1,000,000, 2,000,000, 5,000,000, 10,000,000, 15,000,000, 20,000,000, 25,000,000, 30,000,000, 35,000,000, 40,000,000, 45,000,000, and 50,000,000. In each case, use the interval of $[0, 1]$ for the element range.

   (b) Using a spreadsheet, such as LibreOffice Calc or Excel, plot the sort times verses the number of data items. Put all three lines on the same chart. You will want to use an XY scatter chart with the sizes on the x-axis and times on the y-axis.

   (c) In a word processor, you can use LibreOffice Writer, Word, or any word processor that will allow you to export the document to a PDF file. LibreOffice Writer has a toolbar button for this. In the document include tables of all the data you acquired from the runs you did above, the charts you created from the data and the answers, in complete sentences, to the following.

      - How do the times for these three algorithms compare?
      - Do the curves have approximately the same shape? That is, do they increase along a parabolic-like curve, are they all straight lines, do they have the same types of curves in about the same places?
      - What do you think is accounting for the differences?

3. Using the same data from above, plot the integer array curve and the double array curve on the same plot for the merge sort.

   - How do the times for integer and double arrays compare?
   - Do the curves have approximately the same shape? That is, do they increase along a parabolic-like curve, are they all straight lines, do they have the same types of curves in about the same places?
   - What do you think is accounting for the differences?

4. Using the same data from above, plot the integer array curve and the double array curve on the same plot for the quick sort.

   - How do the times for integer and double arrays compare?
   - Do the curves have approximately the same shape? That is, do they increase along a parabolic-like curve, are they all straight lines, do they have the same types of curves in about the same places?
   - What do you think is accounting for the differences?

5. Using the same data from above, plot the integer array curve and the double array curve on the same plot for the algorithm library sort.

   - How do the times for integer and double arrays compare?

- Do the curves have approximately the same shape? That is, do they increase along a parabolic-like curve, are they all straight lines, do they have the same types of curves in about the same places?
- What do you think is accounting for the differences?

6. Does it look like any of the three algorithms perform better across the board of data types? That is, do any of them show smaller time increases when going from integers to doubles? What might cause these differences?

7. Put all the charts, graphs, and your discussion answers into a single document. Export the document to a PDF file and upload it to MyClasses.