SORTING AND PROBLEM SIZE

Question 1

Create a program (C++) that will perform a few sorting algorithms on an array of size n. In this assignment n is the size of an array to be sorted.

> Running your code at the command line should look something like this: ./lab1. The output will initially include 6 measurements of time for (suitable) values of n chosen by you for each sorting algorithm. Subsequently it will include an additional 6 for a combined version of the sorting algorithms. In the end, 18 time values will be output to the console.

Question 2

Create an array of random elements between 1 and 4n, to populate the array of size n.

> For each value of n in your "main" function, an array is created. This should be filled with randomly chosen values that are between 1 and 4n. Not all values will appear.

(a) Implement Selection Sort to sort the array.

> It is up to you how you choose to implement this

(b) Implement Merge Sort to sort the array.

> It is up to you how you choose to implement this

ANALYSIS

Question 3

In your main function, run both merge sort and selection sort on 6 different choices for n. Output to the console the runtime comparing Selection Sort and Merge sort for each choice of n.

Question 3 continued on next page...

Ideally these should be formatted in a way that makes clear which sorting algorithm is running, the value of n, and how long it took to complete

CAN WE DO BETTER?

Question 4

Repeat question 3 with a modified Merge Sort, where Selection Sort is called within Merge Sort. Try to improve the runtimes from Question 3.

Here you must consider under what condition you might benefit from combining sorting strategies and why

You should submit your solution to the D2L Dropbox for Lab 1. Please name your CPP file according to the scheme lastname_firstname_lab1.cpp.

Here you will replace lastname and firstname with your own name.