Lab 6: Sorting

Introduction:

Welcome back! Hopefully you had a restful spring break, because this lab is likely to give you a headache. Today we're doing sorting algorithms. All of the driver code is written up for you, and quicksort is complete so that you can reference it, but you are going to need to write some pieces of some sorting algorithms.

Lab 6:

First, accept the lab from this link: <u>https://classroom.github.com/a/wiPYBBkq</u>

Mergesort: Most of mergesort and merge are already written for you. You'll need to complete the recursive part of mergesort. Trust me, you're going to want to use the merge function in this part. If you don't understand why, look back in the lectures and make sure you understand what's going on. (Hint: this type of material may appear on an exam.)

With mergesort, change that MAXSIZE variable at the top of the program to be 100 until you know you have it working. Then, change it back to 100,000 for timing purposes.

Heapify: Just like before, you'll need to write the recursive part of this function. This time you don't really need to worry about heapsort, but it may be helpful to read through the code and make sure you understand the relationship between heapsort and heapify.

Once you have your functions written, go ahead and run the program on your machine, then clone into a pi and run it there as well. Remember when running on the pi's that if you make MAXSIZE too big, they may not ever finish. Log those times in your Readme. If you skip this step, you'll lose points.

Change this week for my sanity:

Please don't include your executable files (i.e. a.out, Lab6Out, etc) in your repo. Since it's new, this week I won't take points if you include them. You'll be saving me some brain power.

Submission:

For this week, you'll need to update your README.md and submit your completed Lab6.cpp file. This lab will be due this Sunday, March 30th, at 11:59pm.