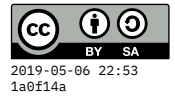


Assignment 4: sorting

26 March 2019



Task

For this assignment, we will collect some data that will help us understand the performance characteristics of different sorting algorithms. Visit this site called xSortLab¹, produced by David J. Eck of Hobart and William Smith Colleges.

You can play with the Visual Sort at the top if you want to, but for the assignment I want to focus on the “Timed Sort” box below that.



¹math.hws.edu
/eck/js/sorting/xSortLab.html

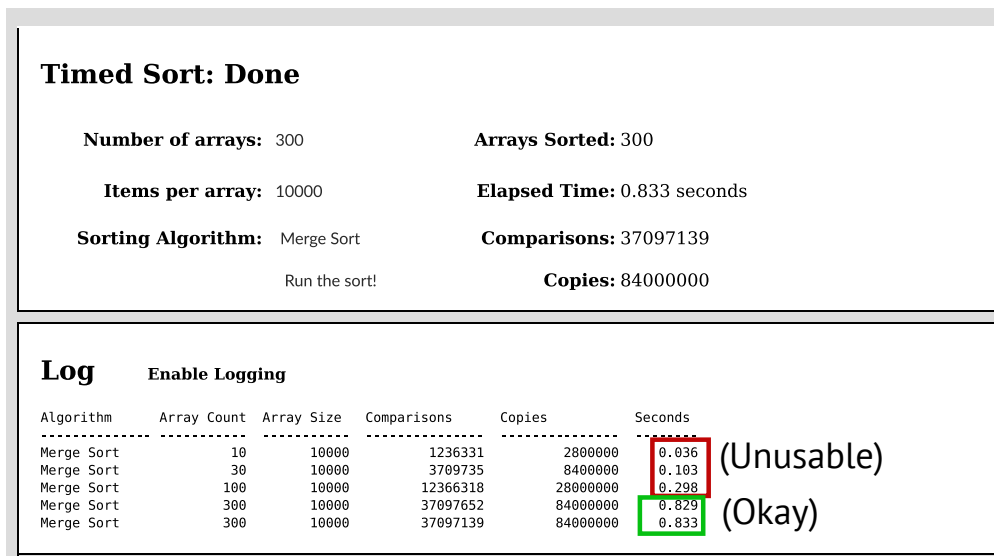


Figure 1: The “Timed Sort” feature of xSortLab

I want you to test and report the results on **four** different sorting algorithms, run on **three** different sizes of arrays (*aka* lists), which can vary from 10,000 up to 200,000. So you will generate **twelve** different pieces of data. The result we’ll focus on is just **Elapsed Time**.

For the sorting algorithms, **do not use Insertion Sort**. We’ll only use results for Bubble, Selection, Merge, and Quick Sort.

Timing something that finishes very quickly can be inaccurate, so we repeat the procedure on a **Number of arrays** to force it to take longer. Whenever your elapsed time is shorter than 0.5 seconds, you should bump up **Number of arrays** to make it take longer.



²forms.gle/bx
vgytsjRcn4nf
PZ6

As you try different settings, all the results will be recorded in the **Log** section. Submit them using this form², one trial run per submission. (So you should submit the form **twelve times**.)