

Sorting Analysis

[Submit Assignment](#)

Due Friday by 11:59pm **Points** 50 **Submitting** a file upload

Implement three sorting routines of your choice. One must be not quadratic (e.g. quicksort, heapsort, mergesort, or shellsort). Create input files of four sizes: 10000, 20000, 50000 and 100000 integers. For each size file make 3 versions. One version will be randomly ordered, one version will be in descending order, and one version will be in ascending order. This means you have a set of 12 input files. Run each sort against all input files. Thus, you will run average case, worst case (descending) and best case (ascending). For each run, access the system clock to get time values. The call to the clock should be placed as closely as possible to the beginning and end of each sort. Time the sorts several times to get an average.

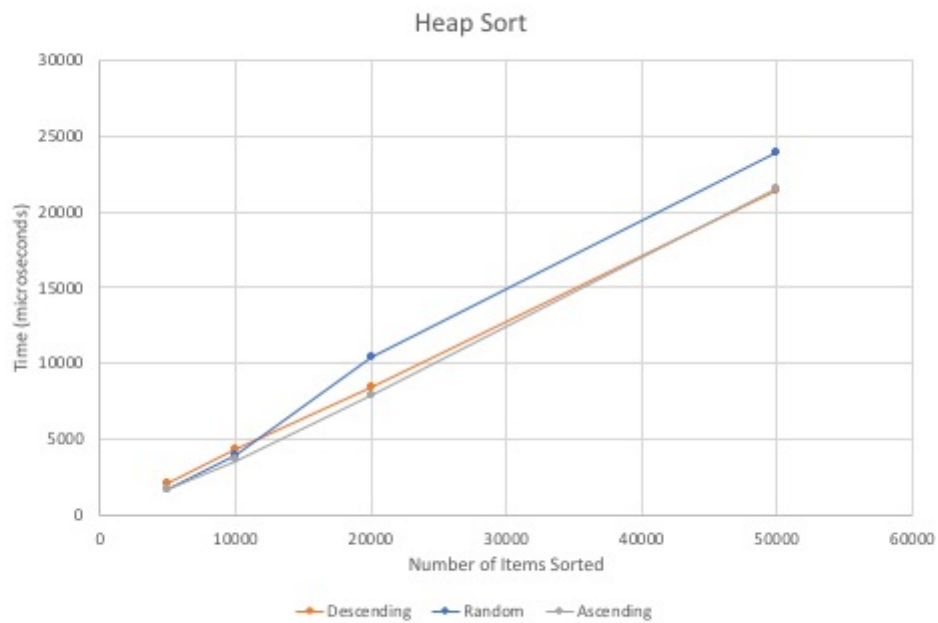
Turn in a commentary comparing the three sorts and their performance. It should include the following.

--An introduction indicating what sorts were tested, how many times they were tested

--x-y (scatter) plots of run time versus file size.

--An analysis. Comment on the relative run times between the different sorts, the effect of the order of the data (ascending, descending, or random). Include Comment on how the size of the file is related to the run time and the computational complexity indicated by the plots.

Here is a sample of what your plots should look like:



Your plots must be x-y. The trend for each line must be able to be seen. If (for example) your ascending case is much lower than the other cases, put it on another plot with a more appropriate time scale. You may make as many plots as necessary.

Your final submission should include:

1. Your implementation of the three sorting algorithms (your source code)
 2. Your full report as a PDF
- Note that the report is 50% of the project grade and should include everything mentioned in the assignment description.

You do not need to submit the files that you sorted or any code that you may have used to generate, test, or time them.

Sorting Rubric

Criteria	Ratings				Pts
Sorts Properly Implemented Sorts implemented, one is not quadratic.	10.0 pts Full Marks		0.0 pts No Marks		10.0 pts
Test Data Test data 10000 to 100000 and ascending, descending and random	10.0 pts Full Marks		0.0 pts No Marks		10.0 pts
Plots Correct plots of time versus file size	10.0 pts Full Marks	6.0 pts Plots not properly implemented Not x-y plots. X axis not points not proportional to value.	3.0 pts Sorts not properly implemented Not time versus number of elements.	0.0 pts No Marks	10.0 pts
Comment on differences in performance between the sort routines	5.0 pts Full Marks		0.0 pts No Marks		5.0 pts
Comment on on the indicated computational complexity Does the plot show quadratic? Log linear? Linear? Why or why not?	10.0 pts Full Marks	5.0 pts Not all computational complexities correct	5.0 pts Not all computational complexities indicated	0.0 pts No Marks	10.0 pts
Comment on on the affect of order What is the affect of ascending, descending and random order?	5.0 pts Full Marks		0.0 pts No Marks		5.0 pts
Total Points: 50.0					