

CS230 Assign7 Task1

Experiment Questions

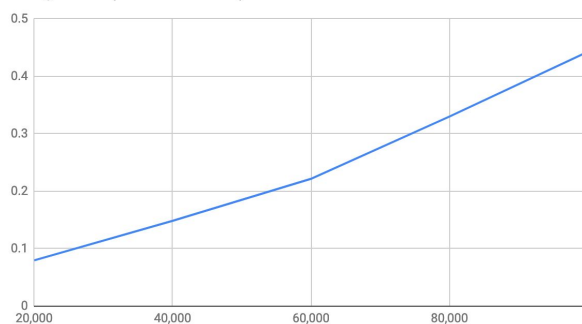
Shirley Lei, Emily Yin

1. Describe the n vs. runtime trend for each sorting algorithm. What happens to the derivative (slope) of each line as n increases? (For instance, does the slope get steeper, shallower or stay the same for larger values of n?)

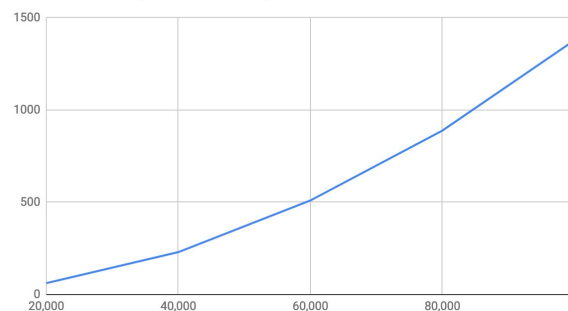
[See graph below]. Generally, the slope of n vs. runtime becomes steeper for larger values of n (see slope calculations as well as line charts). However, we do notice a slight decrease in slope at the middle dataset size (60,000).

n	MergeSort Runtimes (seconds)	Slope for MergeSort	SelectionSort Runtimes (seconds)	Slope for SelectionSort
20,000	0.079732366	0.0000039866183	62	0.0031
40,000	0.148564713	0.0000037141178 25	230	0.00575
60,000	0.221722063	0.0000036953677 17	510	0.0085
80,000	0.330083689	0.0000041260461 13	889	0.0111125
100,000	0.442849676	0.0000044284967 6	1378	0.01378

MergeSort (n vs runtimes)



SelectionSort (n vs runtimes)



2. Which algorithm was fastest on large n? Which was fastest on small n?

The MergeSort algorithm was significantly faster on both large and small values of n.

CS230 Assign7 Task1

Experiment Questions

Shirley Lei, Emily Yin

3. Reflect on the practical significance of these differences in runtime. Under what circumstances would the difference in runtimes you described above matter? Would the usefulness of a more efficient algorithm matter more or less if you were expecting to run an analysis once vs. if you ran it hundreds or thousands of times?

The differences in runtimes between the two algorithms matter more significantly for datasets of larger sizes. For example, for a dataset of $n=20,000$, the runtime difference between the two algorithms is only around a minute, whereas for a dataset of $n=100,000$, this runtime difference is significantly larger at around 23 minutes. This pattern is due to the fact that the runtime for both algorithms grow *increasingly* as n increases. The usefulness of a more efficient algorithm matters more if we were expecting to run an analysis hundreds or thousands of times, since the difference in runtime each time the analysis is run adds up, and so a more efficient algorithm would be significantly time-saving.