Sean Mulholland ID No: 932652783 mulholls@onid.oregonstate.edu

Lab 7 Document-

It's clear that complexity greatly affects the run time of our programs. This was immediately illustrated to me when I used my bubble sort function on a file with 10000 values. This seems like a fairly small file size with all the talk of "big data" this and "big data" that. I let it run for close to 2 minutes and killed my program both times I tried. I then tested some intermediary values at 6000 entries and that took 25 seconds to sort, where as a 3000 value file took ~ 3 seconds. This math doesn't quite add up perfectly to the n^2 time predicted, but it's close enough to convince me to never bubble sort anything ever again!

Other results were as expected, my linear search takes more time through larger files in a pretty linear fashion:

Linear Search Data	0 at 1/4 * values (sec)	0 at 1/2 * values (sec)	0 at 3/4 * values (sec)
1000 values in data file	5.2 e -5	6.8 e -5	1.01 e -4
3000 values in data file	.0001	.000192	.00278
6000 values in data file	.000188	.000368	.000547

The bubble sort time grows rapidly as the data sets increase in size:

Bubble Sort Data	Time (sec)
1000 values in data file	.18
3000 values in data file	3.4
6000 values in data file	25

And I used the bubble sort algorithm to sort my values prior to searching with the binary search, and the search time was clearly overrun by the $O(n^2)$ time even in these small values.

Binary Search Data	0 at 1/4 * values (sec)	0 at 1/2 * values (sec)	0 at 3/4 * values (sec)
1000 values in data file	.18	.18	.18
3000 values in data file	3.4	3.4	3.4
6000 values in data file	25	25	25

I'm sure if I used a better sorting algorithm the Binary search would be totally different. But I think its best application would be when the data coming in was already sorted.