Colin Ngo

Lab 6

Selection Sort		
List Size	Comparisons	Time (seconds)
1,000 (observed)	499500	0.036368199998833006
2,000 (observed)	1999000	0.1385493999987375
4,000 (observed)	7998000	0.5597324999980628
8,000 (observed)	31996000	2.2225857000012184
16,000 (observed)	127992000	8.988649400002032
32,000 (observed)	511984000	38.83317380000153
100,000 (observed)	3.74877232*(10^9)	453.8553986000006
500,000 (estimated)	9.3719308*(10^10)	8745.44443
1,000,000 (estimated)	3.74877232*(10^11)	34981.77772
10,000,000 (estimated)	3.74877232*(10^13)	3498177.772

Insertion Sort		
List Size	Comparisons	Time (seconds)
1,000 (observed)	248360	0.02890320000005886
2,000 (observed)	970642	0.11222979999729432
4,000 (observed)	4045812	0.524608799998532
8,000 (observed)	16004402	2.2252671999995073
16,000 (observed)	63775667	8.680567300001712
32,000 (observed)	257021203	35.04484429999866
100,000 (estimated)	2.13439071*(10^9)	275.214924
500,000 (estimated)	5.33597678*(10^10)	6880.373099
1,000,000 (estimated)	2.13439071*(10^11)	27521.4924
10,000,000 (estimated)	2.13439071*(10^13)	2752149.24

1. Which sort do you think is better? Why?
    a. The insertion sort is better because the constant, comparisons, and time are smaller.
2. Which sort is better when sorting a list that is already sorted (or mostly sorted)? Why?

    a. The insertion sort is better at sorting because it only has to go through less comparisons than the selection sort.
3. You probably found that insertion sort had about half as many comparisons as selection sort. Why? Why are the times for insertion sort not half what they are for selection sort? (For part of the answer, think about what insertion sort has to do more of compared to selection sort.)
    a. The insertion sort doesn't have to go through the whole other part of the list while the selection sort has to find the greatest element on the other larger part of the list.