Ian Myers
Assignment 5 Tables

Execution Time (Bubble Sort)

	4 processes (4x1)	8 processes (4x2)	16 processes (4x4)
40,000 Elements	0.364787	0.140794	0.147212
400,000 Elements	33.046611	8.407684	2.317448

Execution Time (Quick Sort)

	4 processes (4x1)	8 processes (4x2)	16 processes (4x4)
40,000 Elements	0.043585	0.072785	0.122430
400,000 Elements	0.092577	0.112099	0.150195

Analysis

The execution time for bubble sort falls in line with the concept that the more processes a program has the faster it will go. Except for 8 to 16 processes, which shows a slight speed increase for 8 processes. All other entries to the table show that as the number of processes increases, the time it takes to sort decreases. This makes sense with intuition that if we can split up more of the original array we can sort the array in less time. This is most notably highlighted in 400,000 elements from 4 to 8 processes. This shows a decrease in time from 33 to 8 seconds. That is a dramatic shift. The execution time for quicksort, however, does not follow this same intuition. All the entries show that as the number of processes increases, the execution time increases. This strongly combats the intuition taken from bubble sort. I believe this is due to the way quicksort goes about sorting things which takes significantly less time than bubble sorting. This leaves the most time consuming computation to be the odd even sorting which requires every process to speak to each other. Therefore as the number of processes increases, it takes more time to go through the odd even sorting portion of the computation, which results in 16 processes taking more time to sort than 4 processes.