Aaron Schendel

Empirical Analysis

Dr. Breutzmann – CS 320

Sort Wars

The program that I wrote first compares the quicksort, merge sort, and Shell’s sort sorting algorithms for arrays of element lengths 10,000 to 20,000 with increments of 2,000. As expected the results of using these three algorithms on lists of incrementing size is that quicksort is on average the fastest with merge sort next and Shell’s sort taking the longest. When the sort times are divided by n\*log(n) one can see that due to the results in that column staying close to the same it is empirically proven that these three sorts have a big-O efficiency of O(n\*log(n)).

The next thing that the program does is it runs through the selection sort algorithm on arrays of element lengths 1,000 to 2,000 with increments of 200. Immediately one can see that this algorithm is probably slower than the other three on larger lists simply due to the array sizes chosen to compare. This guess is then proven when the next column takes the sort time and divides it by n2 and the numbers stay nearly constant. Doing this empirically proves that selection sort is an O(n2) algorithm.