James Gardner

Professor Davis

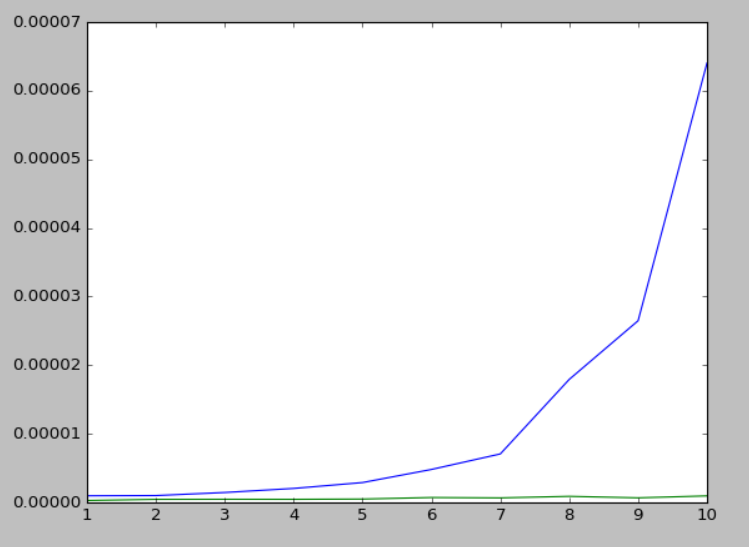
Intro to CS

26 October 2016

Algorithm Analysis

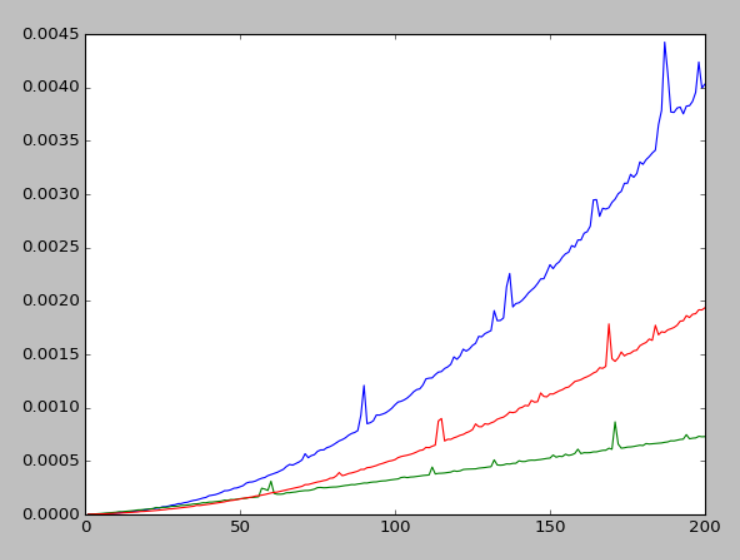
In both graphs, any sudden spikes are outliers in the random number generator.

GCD (Blue) and EUC(Green) Test



Since the worst-case scenario for the GCD algorithm is 2^(number of bits), the running time is proportional to about big ‘O’ of n^2. The worst-case scenario for the EUC algorithm is the number of bits, and its running time is proportional to n. The above graph shows that as the number of bits increase, the time required to calculate the gcd increased dramatically for the GCD algorithm, while the EUC algorithm increases at a much slower rate, so the EUC has a faster running time. Changing the number of tests and bits creates similar results, which collaborates with the class discussion of running times for these algorithms, where the EUC algorithm is faster than the GCD algorithm.

Sorting Algorithms – Bubble(BLUE), Merge(GREEN), and Selection(RED)



The worst-case scenario for both the Bubble Sort and Selection Sort is proportional to big ‘O’ of n^2, which also means the running time increases as the number of bits and trials increases. On the other hand, Merge Sort is proportional to big ‘O’ of n\*log (base 2) of n, which is much slower than n^2. Merge Sort’s running time is much faster than either Bubble or Selection Sort, as the time it takes to sort a list is much smaller than the other algorithms. The above graphic shows the time it took for each algorithm to sort each list over differing list lengths and number of bits. These findings collaborate with the class hypotheses since Merge Sort is much faster than Bubble or Selection Sort. (Interestingly enough, it appears that Selection Sort is faster than Bubble Sort, even though they both share the same proportionality to n in terms of running time;)