Benjamin Tate
3/11/16
CS 162 – Section 400
Lab 10 Analysis and Discussion

Results:

Fibonacci's Sequence											
n =	1	5	10	15	20	25	27	30	35	39	*average*
Iteration time (ns)	8102	9429	8032	9498	9429	9219	8101	9010	7683	9628	*8813.1*
Double recursion time (ns)	5099	5657	6984	14108	83531	852622	1379087	8945067	67467284	433559673	*51231911.2*
Factorials											
n =	1	2	3	4	5	6	7	8	10	12	*average*
Single recursion time (ns)	4889	5028	5238	5517	5378	4959	4889	5238	5727	5098	*5196.1*
Tail recursion time (ns)	3212	3212	3841	3282	3422	3282	3492	3213	3772	3142	*3387*

Analysis and discussion:

For the Fibonacci's sequence calculations, the iterative method was by far more efficient than the double recursive method at higher values of n, as expected. For my n values, I decided that there wasn't much need to go higher than around 40, because the difference between the two methods was already so apparent at that point, and any higher than that and the wait times started to actually get kind of annoyingly long. I decided to test 10 values of n, and spread them out relatively evenly, with a couple closer together at the beginning, middle and end (1 and 5, 25 and 27, 35 and 39). Like I said, I expected that the iterative method would be more efficient at higher values of n, but there were a couple things that surprised me. For one, I didn't expect the double recursion method to be significantly faster at any values of n, but it turned out that it was through at least n = 10. Also, I expected both methods to rise steadily with n, and for the iterative times to rise more slowly than the recursive, but there didn't seem to be any significant pattern of increase in the iterative method's execution time. In fact, the time at n = 35 was actually less than that of n = 1 by nearly as much as n = 1 was lower than the average. It's possible that this would have changed with far greater values of n, but like I said, at that point the recursive time was so high that it seemed unnecessary to wait.

The factorial methods were surprising in that neither method's execution time seemed to have a significant upward trend with rising values of n. That said, the tail recursion method was significantly faster than the single recursion method, as expected. In fact, the very slowest tail recursion time (at n = 3) was still faster than the single recursion's fastest time (at n = 7 and n = 1) by more than 1000ns.

Unfortunately, no significant comparisons can be drawn between the iterative or double recursive methods and the single or tail recursive methods, since their respective functions were totally different and therefore obviously took different amounts of computing power.